HYDIAG : extended diagnosis and prognosis for hybrid systems

Elodie Chanthery, Yannick Pencolé, Pauline Ribot, Louise Travé-Massuyès

HYDIAG: extended diagnosis and prognosis for hybrid systems

Elodie Chanthery[1,2], **Yannick Pencolé**[1], **Pauline Ribot**[1,3], **Louise Travé-Massuyès**[1]
[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
e-mail: [firstname.name]@laas.fr
[2]Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
[3]Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France

Abstract

HYDIAG is a software developed in Matlab by the DISCO team at LAAS-CNRS. It is currently a software designed to simulate, diagnose and prognose hybrid systems using model-based techniques. An extension to active diagnosis is also provided. This paper aims at presenting the native HYDIAG tool, and its different extensions to prognosis and active diagnosis. Some results on an academic example are given.

1 Introduction

HYDIAG is a software developed in Matlab, with Simulink. The development of this software was initiated in the DISCO team with contributions about diagnosis on hybrid systems [1]. It has undergone many changes and is currently a software designed to simulate, diagnose and prognose hybrid systems using model-based techniques [2; 3; 4]. An extension to active diagnosis has been also realized [5; 6]. This article aims at presenting the native HyDiag tool and its different extensions to prognosis and active diagnosis.

Section 2 recalls the hybrid formalism used by HYDIAG. Section 3 presents the native HYDIAG tool that simulates and diagnoses hybrid systems. Section 4 explains how HYDIAG has been extended in HYDIAGPRO to prognose and diagnose hybrid systems. Section 5 presents the extension to active diagnosis. Experimental results of HYDIAG and its extension HYDIAGPRO are finally presented in Section 6.

2 Hybrid Model for Diagnosis

HYDIAG deals with hybrid systems. Such a system must be modeled by a hybrid automaton [7]. Formally, a hybrid automaton is defined as a tuple $S = (\zeta, Q, \Sigma, T, C, (q_0, \zeta_0))$ where:

- ζ is a finite set of continuous variables that comprises input variables $u(t) \in \mathbb{R}^{n_u}$, state variables $x(t) \in \mathbb{R}^{n_x}$, and output variables $y(t) \in \mathbb{R}^{n_y}$.
- Q is a finite set of discrete system states.
- Σ is a finite set of events.
- $T \subseteq Q \times \Sigma \rightarrow Q$ is the partial transition function between states.
- $C = \bigcup_{q \in Q} C_q$ is the set of system constraints linking continuous variables.
- $(\zeta_0, q_0) \in \zeta \times Q$, is the initial condition.

Each state $q \in Q$ represents a behavioural mode that is characterized by a set of constraints C_q that model the linear continuous dynamics (defined by their representations in the state space as a set of differential and algebraic equations). A behavioural mode can be nominal or faulty (anticipated faults). The unknown mode can be added to model all the non anticipated faulty situations. The discrete part of the hybrid automaton is given by $M = (Q, \Sigma, T, q_0)$, which is called the *underlying discrete event system (DES)*. Σ is the set of events that correspond to discrete control inputs, autonomous mode changes and fault occurrences. The occurrence of an anticipated fault is modelled by a discrete event $f_i \in \Sigma_f \subseteq \Sigma_{uo}$, where $\Sigma_{uo} \subseteq \Sigma$ is the set of unobservable events. $\Sigma_o \subseteq \Sigma$ is the set of observable events. Transitions of T model the instantaneous changes of behavioural modes. The continuous behaviour of the hybrid system is modelled by the so called *underlying multimode system* $\Xi = (\zeta, Q, C, \zeta_0)$. The set of directly measured variables is denoted by $\zeta_{OBS} \subseteq \zeta$.

An example of a hybrid system modeled by a hybrid automaton is shown in Figure 1. Each mode q_i is characterized by state matrices A_i, B_i, C_i and D_i.

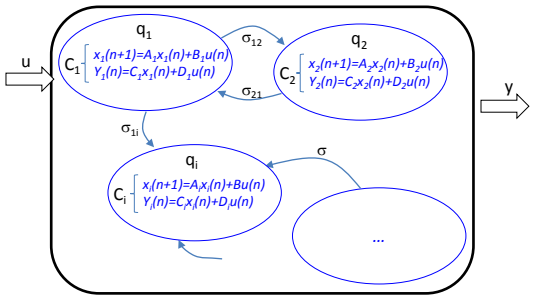


Figure 1: Example of an hybrid system

3 Overview of the native HYDIAG diagnoser

The method developed in [1] for diagnosing faults on-line in hybrid systems can be seen as interlinking a standard diagnosis method for continuous systems, namely the parity space method, and a standard diagnosis method for DES, namely the diagnoser method [8].

3.1 How to use HYDIAG ?

Step 1: hybrid model edition

HYDIAG allows the user to edit the modes of a hybrid automaton S as illustrated in Figure 1. To model the system, the user must first provide the following information to the HYDIAG software: the number of modes, the number of discrete events that can be observable or unobservable, and the sampling period used for the underlying multimode system (defined by the set of state matrices of the state space representation of each mode).

There are optional parameters that are helpful to initialize the mode matrices automatically before editing them: the number of entries for the continuous dynamics, the number of outputs for continuous dynamics, the dimensions of each matrix A. The number of entries (resp. outputs) must be the same for all the modes.

The simulator of the edited model has no restrictions on the number of modes or the order of the continuous dynamics, it is generically designed. Online computations are performed using Matlab / Simulink. Results provided by Matlab can be reused if a special need arises. Figure 2 shows an overview of the software interface.

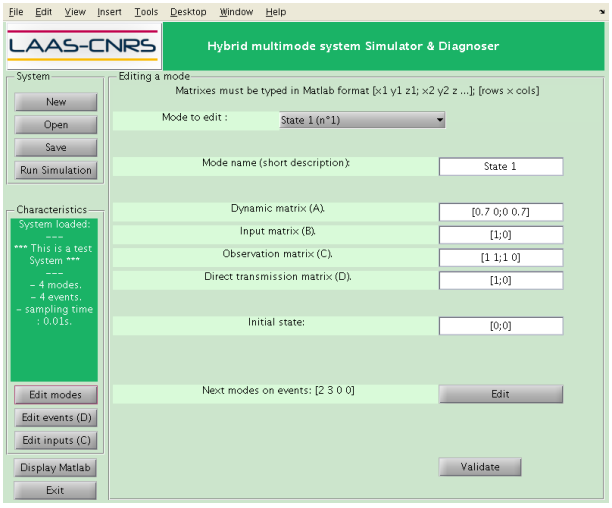


Figure 2: HYDIAG Graphical User Interface

Step 2: building the diagnoser

HYDIAG automatically computes the analytical redundancy relations (ARRs) by using the parity space approach [9]. Details of this computation can be found in [10].

The idea of HYDIAG is to capture both the continuous dynamics and the discrete dynamics within the same mathematical object. To do so, the discrete part of the hybrid system $M = (Q, \Sigma, T, q_0)$ is enriched with specific observable events that are generated from continuous information. The resulting automaton is called the *Behaviour Automaton* (BA) of the hybrid system. HYDIAG then builds the diagnoser of the Behaviour Automaton (see [8]) by using the DIADES[1] software also developed within the DISCO team at LAAS-CNRS (see an example of diagnoser in Figure 7).

Step 3: system simulation and diagnosis

Given the built hybrid diagnoser, HYDIAG then loads a set of timed observations produced by the system and it provides at each observation time an update of the diagnosis of the system by triggering the current transition of the hybrid diagnoser that matches the current observation. It is possible to define in HYDIAG a simulation scenario for the modeled system with a duration and a time sample defined by the user.

[1] http://homepages.laas.fr/ypencole/DiaDes/

3.2 Software architecture with extensions

The general architecture of HYDIAG and its two extensions (see the next sections for their description) is presented on Figure 3. Ellipses represent the objects handled by the software, rectangles with rounded edges depict HYDIAG functions and rectangles with straight edges correspond to external DIADES packages. The behaviour automaton is at the heart of the architecture as HYDIAG and both its extensions rely on it to perform diagnosis, active diagnosis and prognosis.

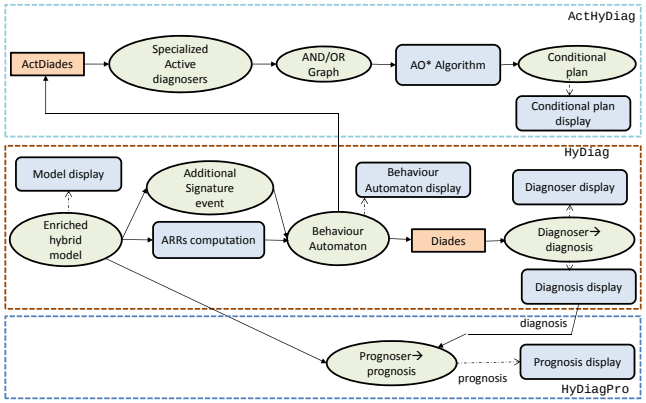


Figure 3: HYDIAG architecture with its extensions HYDIAGPRO and ACTHYDIAG.

4 HYDIAGPRO : an extension for Prognosis

HYDIAG has been extended in order to provide a prognosis functionality to the software [4]. The prognosis function computes (1) the fault probability of the system in each behavioural mode, (2) the future fault sequence that will lead to the system failure, (3) the Remaining Useful Life (RUL) of the system.

In HYDIAGPRO, the initial hybrid model is enriched by adding for each behavioural mode a set of aging laws: $S^+ = (\zeta, Q, \Sigma, T, C, \mathcal{F}, (q_0, \zeta_0))$ where $\mathcal{F} = \{F^q, q \in Q\}$ and F^q is a set of aging laws one for each anticipated fault $f \in \Sigma_f$ in mode q. The aging modeling framework that is adopted in HYDIAGPRO is based on the Weibull probabilistic model [11] (see more details in [4]). The Weibull fault probability density function $W(t, \beta_j^q, \eta_j^q, \gamma_j^q)$ gives at any time the probability that the fault f_j occurs in the system mode q. Weibull parameters β_j^q and η_j^q are fixed by the system mode q and characterise the degradation in mode q that leads to the fault f_j. Parameter γ_j^q is set at runtime to memorize the overall degradation evolution of the system accumulated in the past modes [11].

The prognoser uses the aging laws in S^+ to predict fault occurrences (see Figure 3). The prognoser uses the current diagnosis result to update on-line these aging laws (the parameters γ_j^q) according to the operation time in each behavioural mode. For each new result of diagnosis, the prognosis function computes the most likely sequence of dated

faults that leads to the system failure. From this sequence is estimated the system RUL [4].

5 ACTHYDIAG: Active Diagnosis

The second extension of HYDIAG provides an active diagnosis functionality to the software (see Figure 3). The inputs are the same as for HYDIAG but an additional file indicates the events of S that are actions, as well as their respective cost. Based on the behaviour automaton, we compute a set of specialised active diagnosers (one per fault): such a diagnoser is able to predict, based on the behaviour automaton, whether a fault can be diagnosed with certainty by applying an action plan from a given ambiguous situation [6]. From these diagnosers, we also extract a planning domain as a AND/OR graph.

At runtime, when HYDIAG is diagnosing, the diagnosis might be ambiguous. An active diagnosis session can be launched as soon as a specialised active diagnoser can analyse that the current faulty situation is discriminable by applying some actions. If the active diagnosis session is launched, an AO* algorithm starts and computes a conditional plan from the AND-OR graph that optimises an action cost criterion. It is important to note that in the case of a system with continuous dynamics, only discrete actions are contained in the active diagnosis plan issued by ACTHYDIAG. In particular, it is assumed that if it is necessary to guide the system towards a value on continuous variables, the synthesis of control laws must be performed elsewhere.

6 HyDiag/HyDiagPro Demonstration

Water tank system model

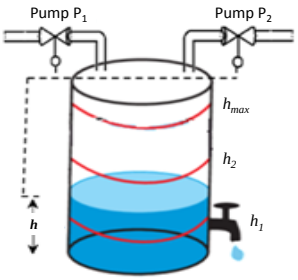
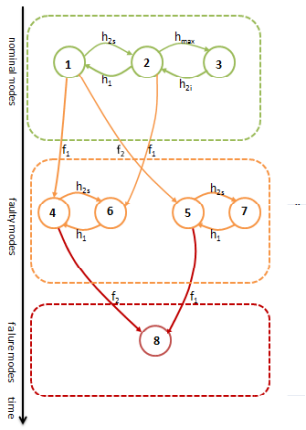


Figure 4: Water tank system

HYDIAGPRO has been tested on a water tank system (Figure 4) composed of one tank with two hydraulic pumps (P_1, P_2). Water flows through a valve at the bottom of the tank depending on the system control. Three sensors (h_1, h_2, h_{max}) detect the water level and allow to set the control of the pumps (on/off). It is assumed that the pumps may fail only if they are on. The discrete model of water tank and the controls of pumps are given in Figure 5. Discrete events in $\Sigma = \{h_1, h_{2s}, h_{2i}, h_{max}, f_1, f_2\}$ allow the system to switch into different modes. Observable events are $\Sigma_o = \{h_1, h_{2s}, h_{2i}, h_{max}\}$. Two faults that correspond to the pump failures are anticipated $\Sigma_f = \{f_1, f_2\}$ and are not observable.The Weibull parameter values of aging models $\mathcal{F} = \{F^{q_i}\}$ are reported in Table 1.

The underlying continuous behaviour of every discrete mode q_i for $i \in \{1..8\}$ is represented by the same state



mode \ pump	Pump1	Pump 2
1	ON	ON
2	ON	OFF
3	OFF	OFF
4	Fail	ON
5	ON	Fail
6	Fail	OFF
7	OFF	Fail
8	Fail	Fail

Figure 5: Water tank DES model

Table 1: Weibull parameters of aging models

Aging laws		β	η	**Aging laws**		β	η
F^{q_1}	$f_1^{q_1}$	1.5	3000	F^{q_2}	$f_1^{q_2}$	2	3000
	$f_2^{q_1}$	1.5	4000		$f_2^{q_2}$	1	7000
F^{q_3}	$f_1^{q_3}$	1	8000	F^{q_4}	$f_1^{q_4}$	NaN	NaN
	$f_2^{q_3}$	1	7000		$f_2^{q_4}$	2	4000
F^{q_5}	$f_1^{q_5}$	2	3000	F^{q_6}	$f_1^{q_6}$	NaN	NaN
	$f_2^{q_5}$	NaN	NaN		$f_2^{q_6}$	1	7000
F^{q_7}	$f_1^{q_7}$	1	8000	F^{q_8}	$f_1^{q_8}$	NaN	NaN
	$f_2^{q_7}$	NaN	NaN		$f_2^{q_8}$	NaN	NaN

space:

$$\begin{cases} X(k+1) &= AX(k) + BU(k) \\ Y(k) &= CX(k) + DU(k) \end{cases} \quad (1)$$

where the state variable X is the water level in the tank, continuous inputs U are the flows delivered by the pumps P_1, P_2 and the flow going through the valve, $A = (1)$, $B = \begin{pmatrix} eTe/S \\ eTe/S \\ eTe/S \end{pmatrix}$ with Te the sample time, S the tank base area and $e_i = 1$ (resp. 0) if the pump is turned on (resp. turned off), $C = (1)$ and $D = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

HYDIAG results

Figure 6 presents the set of results obtained by HYDIAG and HYDIAGPRO on the folllowing scenario. The time horizon is fixed at $T_{sim} = 4000$h, the sampling period is $T_s = 36$s and the filter sensitivity for the diagnosis is set as $T_{filter} = 3$min. The residual threshold is 10^{-12}. The scenario involves a variant use of water (max flow rate = 1200L/h) depending on user needs during 4000h. Pumps are automatically controlled to satisfy the specifications indicated above. Flow rate of P_1 and P_2 are respectively 750L/h and 500L/h.

The diagnoser computed by HYDIAG is given in Figure 7. Each state of the diagnoser indicates the belief state in the model enriched by the abstraction of the continuous part of the system, labelled with faults that have occurred on the system. This label is empty in case of nominal mode. In the scenario, fault f_1 was injected after 3500h and fault f_2 was not injected.

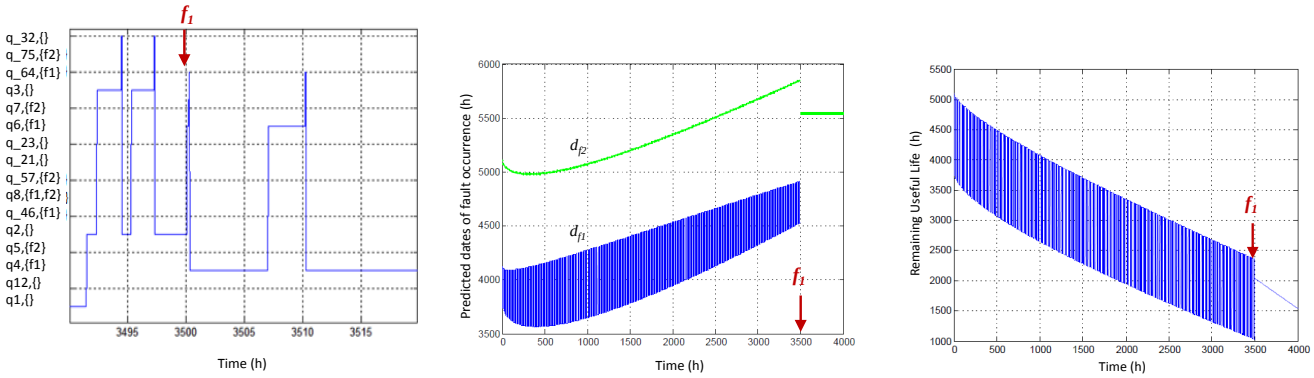


Figure 6: Scenario: Diagnoser belief state (left), Prognosis results of degradations d_{f_1} and d_{f_2} (middle), System RUL (right).

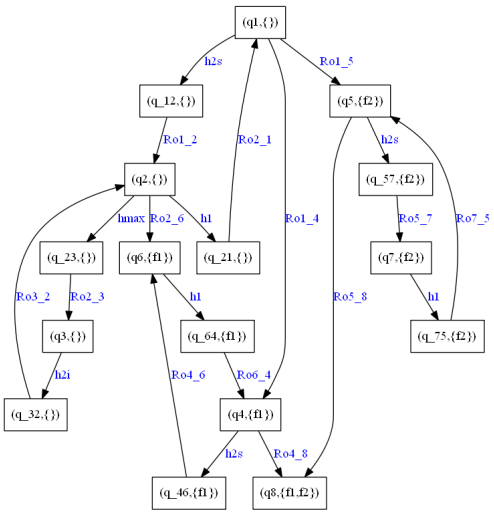


Figure 7: Diagnoser state tracker

Left hand side of Figure 6 shows the diagnoser belief state just before and after the fault f_1 occurrence. Results are consistent with the scenario: before 3500h, the belief states of the diagnoser are always tagged with a nominal diagnosis. After 3500h, all the states are tagged with f_1.

Middle of Figure 6 illustrates the predicted date of fault occurrence (d_{f_1} and d_{f_2}). At the beginning of the process, the prognosis result is: $\Pi_0 = (\{f_1, 4120\}, \{f_2, 5105\})$. It can be noted that the predicted dates d_{f_1} and d_{f_2} of f_1 and f_2 globally increase. Indeed, the system oscillates between stressful modes and less stressful modes. To make it simple, we can consider that in some modes, the system does not degrade, so the predicted dates of f_1 and f_2 are postponed. Before 3500h, the predicted date of f_1 is lower than the one of f_2. After 3500h, the predicted date of f_2 is updated, knowing that the system is in a degraded mode. Finally, the prognosis result is $\Pi_{3501} = (\{f_2, 5541\})$. Figure 6 shows the evolution of the RUL of the system. At $t = 3501$, as the fault f_2 is estimated to occur at $t = 5541$, the system RUL at $t = 3501$ is $5541 - 3501 = 2040$h.

7 Conclusion

HYDIAG is a software developed in Matlab, with Simulink, by the DISCO team, at LAAS-CNRS. This tool has been extended into HYDIAGPRO to simulate, diagnose and prognose hybrid systems using model-based techniques. Some results on an academic example are exposed in the paper. An extension to active diagnosis is also presented. The active diagnosis algorithm is currently tested on a concrete industrial case.

References

[1] M. Bayoudh, L. Travé-Massuyès, and X. Olive. Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *IFAC World Congress*, pages 7265–7270, 2008.

[2] P. Ribot, Y. Pencolé, and M. Combacau. Diagnosis and prognosis for the maintenance of complex systems. In *IEEE International Conference on Systems, Man, and Cybernetics*, 2009.

[3] Elodie Chanthery and Pauline Ribot. An integrated framework for diagnosis and prognosis of hybrid systems. In *the 3rd Workshop on Hybrid Autonomous System (HAS)*, Roma, Italy, March 2013.

[4] S. Zabi, P. Ribot, and E. Chanthery. Health Monitoring and Prognosis of Hybrid Systems. In *Annual Conference of the Prognostics and Health Management Society* , 2013.

[5] M. Bayoudh, L. Travé-Massuyès, and Xavier Olive. Active diagnosis of hybrid systems guided by diagnosability properties. In *Proceeding of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09*, pages 1498–1503, Barcelona, Spain, 2009.

[6] E. Chanthery, Y. Pencolé, and N. Bussac. An ao*-like algorithm implementation for active diagnosis. In *10th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS,*, 2010.

[7] T. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 278–292, 1996.

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

[9] M Staroswiecki and G Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5):687–699, 2001.

[10] M. Maiga, E. Chanthery, and L. Travé Massuyès. Hybrid system diagnosis : Test of the diagnoser hydiag on a benchmark of the international diagnostic competition DXC 2011. In *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2012.

[11] P. Ribot and E. Bensana. A generic adaptative prognostic function for heterogeneous multi-component systems: application to helicopters. In *European Safety & Reliability Conference*, Troyes, France, September 18-22 2011.