A Holistic Advanced Diagnosis Approach For Systems Under Uncertainty

Amaury Vignolles, Elodie Chanthery, Pauline Ribot

A Holistic Advanced Diagnosis Approach For Systems Under Uncertainty

Amaury Vignolles[1,2], **Elodie Chanthery**[1,2] and **Pauline Ribot** [1,3]
[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
e-mail: first.last@laas.fr
[2]Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
[3]Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France

Abstract

This paper presents a holistic method for health monitoring of systems under uncertainty. The ambition is to propose a method that can be applied indifferently to systems usually monitored by methods specific to the dynamics of these systems. Here, we want to be able to monitor continuous dynamic systems, discrete event systems or hybrid systems using the same formalism and the same methodology.To do so, the formalism of Heterogeneous Petri Nets (HtPN) is introduced. Then, the article focuses on an advanced diagnostic method. This method tracks the system health state taking into account uncertainties related to modeling and observations through the construction of an HtPN-based diagnoser. Two application cases illustrate the efficiency of the proposed diagnostic method: one is purely discrete whilst the other is hybrid.

1 Introduction

Depending on the chosen level of abstraction, the behavior of complex industrial systems can be represented by discrete and/or continuous dynamics. Literature on system health monitoring usually classifies dynamic systems in one of the following categories: Discrete Event Systems (DES), Continuous Systems (CS) or Hybrid Systems (HS), presenting both discrete and continuous aspects. In this paper, we propose a new formalism for modeling and monitoring the health evolution of systems that can be either purely discrete, purely continuous, or hybrid. In this way, we say that the proposed health monitoring method is holistic, as it can be applied to any type of system.

Moreover, uncertainties are intrinsically linked to complex systems. Monitoring these systems therefore requires taking into account uncertainties related to modeling and continuous and/or discrete observations of the system. The formalism and the health monitoring method that we propose must take these uncertainties into account.Hence, the formalism we propose should allow to represent parallelism, synchronization or temporal dynamics. Thus, uncertainties and degradation dynamics can be represented and evaluated.

The proposed formalism is called Heterogeneous Petri Nets (HtPN).Based on this formalism, we propose an advanced diagnostic method that can be applied to any type of systems (DES, CS or HS). Advanced diagnosis aims at tracking the system current health state. The concept of health state is related to both the degradation of the system and the occurrence of faults. A diagnoser based on the HtPN formalism is defined to monitor the system health state under uncertainty (models, discrete observations and sensors). The health state outputted by the diagnoser includes discrete, continuous and degradation information of the system.

The paper is organized as follow. Section 2 gives some related work. Section 3 presents the HtPN formalism. Section 4 proposes the advanced diagnostic methodology insisting on the management of uncertainties. Then, Section 5 illustrates the diagnosis methodology on two different study cases: a discrete event system and a hybrid system, showing that in both cases, the proposed diagnosis method succeeds in monitoring the system health state taking into account uncertainties. Section 6 concludes the article and gives some leads to future works.

2 Related Work

This section aims to give a vision of the main formalisms that can be used for representing the behavior of complex systems. The objective is to show the limitations of existing formalisms and to identify useful characteristics for systems subject to degradation and uncertainty such as parallelism, synchronization or temporal dynamics.

In hybrid automata, defined in [1], each discrete state is associated with a continuous dynamics. Only one state of this model can be active at a time: the notion of parallelism does not exist. The transitions are defined by 5-tuples $(q, Guard, \sigma, Jump, q')$ with q the state before the transition, q' the state after the transition, $Guard$ the condition to fulfill in order to fire the transition, σ the event received or emitted during the transition firing and $Jump$ the changes on the continuous variables taking place during the firing.The concepts such as $Guard$ and $Jump$ are very interesting for our study. Even if hybrid automata composition is possible, they cannot share a common state. Therefore it is impossible to encompass uncertainty as we would like to do.

Hybrid Petri nets, defined in [2], have continuous and discrete places. Tokens situated in continuous places are real numbers, whereas tokens in discrete places are natural numbers. Two types of transitions can be distinguished: continuous transitions and discrete transitions. In the case of continuous transitions, a crossing quantity is defined and acts like a weight on the arcs. It is possible for transitions to have both types of places as inputs. This formalism allows

to represent parallelism as different tokens can evolve simultaneously in the model. However, the way continuous places are defined does not exactly fit our need because they are not associated with any continuous dynamics. This formalism does not make it possible to model the continuous behavior of the system or its degradation.

In mixed Petri nets, proposed by [3], a place can be continuous and associated to one or more differential equations. A place can also represent a discrete phenomenon. The continuous variables evolution is modeled through the set of equations activated when a place is marked or when a marking is true. The ideas of $Jump$ and $Guard$ from hybrid automaton are also present in this formalism. Parallelism is possible as long as the marked places do not change the value of the same variable through equations. The continuous variables are shared with the whole system and evolve following the active equations. This model is a bit restrictive because it does not allow two active places in parallel to modify the same global variables.

A Petri net based model to represent heterogeneous embedded system (PRES) is introduced in [4]. In this formalism, a token is a pair $k = (v_k, r_k)$ where v_k is the token value, which can be of any type, and where r_k is the token time. Places are associated with one type of token (integer for example). A transition is associated with an output function, which makes the token value evolve, a function delay, which makes the token time evolve and a guard, which is the condition to fire the transition. These notions are interesting but some aspects remain limiting: the combination of discrete and continuous behaviors is hard and the token value does not evolve in the places but solely using the transitions and their output functions.

A generic modeling framework for diagnosing heterogeneous systems is proposed in [5], but uncertainty related to modeling and observations is not considered and no specific diagnostic method is implemented in this work.

A model-based diagnostic method for hybrid systems is presented in [6]. It performs fault detection and isolation by using residuals and a fault signature. This method focuses on faults and not on the system degradation.Moreover, a single fault assumption is done in this work, explained by the fact that the system behavior after the fault occurrence is unknown. In our study, multiple faults can occur on the system. It means the system behavior after the occurrence of faults must be known beforehand for all faults. A non-specified fault cannot be detected with the proposed method.

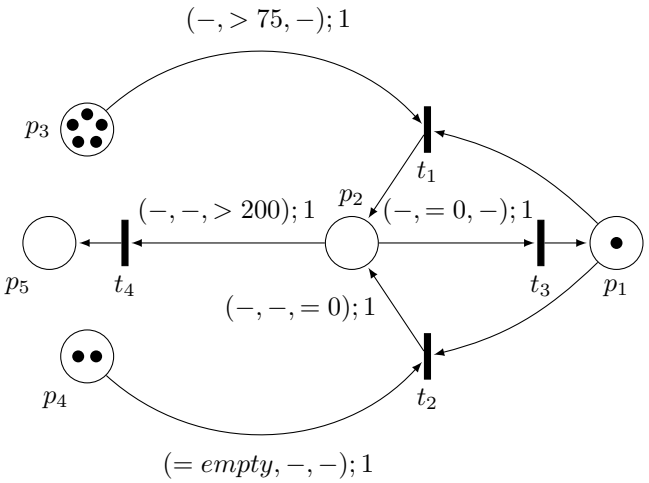
A formalism named Hybrid Particle Petri Nets (HPPN) is proposed in [7]. A diagnostic method has been developed based on this formalism. However, it can only be applied to hybrid systems. As the HPPN formalism was limited to hybrid systems and required the knowledge of a large number of parameters, we have worked on its simplification and on the extension of its application to a large class of systems by taking into account the most interesting concepts of all the formalisms of our knowledge.

3 Heterogeneous Petri Nets Formalism

An Heterogeneous Petri Net (HtPN) is formally defined as $HtPN =< P, T, A, Guard, Jump, E, X, \Gamma, C, D, \mathbb{M}_0 >$ where:

- P is the set of places;
- T is the set of transitions;
- $A \subset (P \times T \cup T \times P)$ is the set of arcs;
- $Guard$ is the set of conditions associated with the incoming arcs;
- $Jump$ is the set of assignments associated with the outgoing arcs;
- E is the set of event labels: $E = E_o \cup E_{uo}$, where E_o is the set of the labels of observable events and E_{uo} is the set of the labels of unobservable events;
- $X \subset \mathbb{R}^{n_N}$ is the state space of the continuous state vector, where $n_N \in \mathbb{N}_+$ is the finite number of continuous state variables;
- $\Gamma \subset \mathbb{R}^{n_D}$ is the state space of the degradation state vector, where $n_D \in \mathbb{N}_+$ is the finite number of degradation state variables;
- C is the set of continuous system dynamics;
- D is the set of degradation system dynamics;
- $\mathbb{M}_0$ is the initial marking of the network.

An example of an HtPN is given in Figure 1 and is detailed in the next sections.



(a)

Places P	Continuous dynamics C	Degradation dynamics D
p_1	-	-
p_2	$x_{k+1} = x_k - 1$	$\gamma_{k+1} = \gamma_k + 3$
p_3	$x_{k+1} = x_k + 3$	-
p_4	$x_{k+1} = x_k + 2$	-
p_5	-	-

(b)

Figure 1: Example of an HtPN (a) and dynamics associated to its places (b)

3.1 Places

In HtPN, a place $p \in P$ is associated with a set of equations $C_p \in C$ modeling one of the system continuous dynamics and the associated noise (uncertainties on the evolution and on the measurements) as well as a set of equations $D_p \in D$ modeling one of the system degradation dynamics:

$$p = \begin{Bmatrix} C_p \\ D_p \end{Bmatrix} \quad (1)$$

The continuous dynamics C_p represents the evolution of the continuous state vector x when the system is in the place p. The degradation dynamics D_p represents the evolution of the degradation state vector γ when the system is in the place p. A degradation dynamic represents the knowledge on fault occurrence or evolution. These continuous and degradation dynamics can be simple IO functions or more complex models learned through machine learning, for example. In the case no continuous or degradation dynamics is specified for the place p, the symbol $-$ is used to define C_p and D_p.

Equations associated to the places of the HtPN in Figure 1(a) are presented in Table 1(b). Places p_1 and p_5 have neither continuous nor degradation dynamics. Places p_3 and p_4 only have continuous dynamics. Place p_2, on the other hand, has both continuous and degradation dynamics.

3.2 Tokens

A place p contains tokens that each have three attributes $< \delta_k^h, \pi_k^h, \phi_k^h >$, representing respectively discrete, continuous and degradation information for a token h at time k. These attributes evolve according to discrete events and dynamics associated to the place the token h belongs to. The exponent h will be omitted in the following section for better readability.

The discrete information carried by h is called a *configuration*. The configuration δ_k of a token is the set of events that have occurred in the system up to the time k:

$$\delta_k = \{(v, \kappa) | \kappa \leq k\} \tag{2}$$

where (v, κ) represents an event $v \in E$ that occur at time κ.

The continuous information carried by h is called the *continuous state* of the token. The continuous state π_k represents the continuous state vector $x_k \in X$ of the token at time k. The continuous state of the token evolves according to the continuous dynamics C_p of the place it belongs to. If no continuous dynamics is specified, the continuous state remains constant.

The degradation information carried by h is called the *degradation state* of the token. The degradation state ϕ_k represents the degradation state vector $\gamma_k \in \Gamma$ at time k. The degradation state of the token evolves according to the degradation dynamics D_p of the place it belongs to. If no degradation dynamics is specified, the degradation state remains constant.

Definition 1 (Marking of the HtPN)**.** *The marking $\mathbb{M}_k$ of a HtPN at time k is the distribution of tokens in the different places. The marking of a place p at time k is denoted $M_k(p)$.*

Initial marking $\mathbb{M}_0$ represents the initial conditions of the system. Each token carries its initial configuration (the set of events that have occurred until time 0), its initial continuous state and its initial degradation state.

3.3 Arcs

The arcs are divided into two sets: $A_{\bullet t}$ contains the incoming arcs and $A_{t\bullet}$ contains the outgoing arcs of transitions:

$$A = A_{\bullet t} \cup A_{t\bullet}$$

Incoming arcs

An arc $a_{p,t} \in A_{\bullet t}$ going from place $p \in P$ to transition $t \in T$ wears a set $\Omega_{p,t} \in Guard$:

$$\Omega_{p,t} = \left\{(\Omega_{p,t}^S, \Omega_{p,t}^C, \Omega_{p,t}^D); \rho_{p,t}\right\} \tag{3}$$

where $(\Omega_{p,t}^S, \Omega_{p,t}^C, \Omega_{p,t}^D)$ represents respectively a discrete, a continuous and a degradation condition, and $\rho_{p,t} \in \mathbb{N}^+$ is a weight.

The *discrete condition* $\Omega_{p,t}^S$ is related to the configuration of the tokens located in the upstream places of a transition t. It may test the occurrence of one (or more, in the case of a logical expression) event $v \in E$: $\Omega_{p,t}^S(\delta_k) = occ(b_k, v)$ which is true if $v \in b_k$.

The *continuous condition* $\Omega_{p,t}^C$ is related to the continuous state of the tokens in the input places of transition t. It may represent a constraint or a test on the continuous state vector x_k: $\Omega_{p,t}^C(\pi_k) = c(x_k)$, where $c(x_k)$ outputs TRUE iff $x_k = \Omega_{p,t}^N(\pi_k)$.

The *degradation condition* $\Omega_{p,t}^D$ is related to the degradation state of the tokens in the input places of t. It may represent a constraint on the degradation state vector γ_k: $\Omega_{p,t}^D(\phi_k) = c(\gamma_k)$, where $c(\gamma_k)$ outputs TRUE iff $\gamma_k = \Omega_{p,t}^D(\phi_k)$.

If one of these conditions is not specified, it is represented by the symbol $-$ in the condition set. By default, $\Omega_{p,t} = \{(\top, \top, \bot); 1\}$.

Pre is the matrix containing the values of the weights of the incoming arcs, of dimensions $P \times T$. $Pre(p,t)$ denotes the value of the weight of the arc $a_{p,t}$ connecting the place p to the transition t.

For example, in Figure 1(a), the arc a_{p_3,t_1} is associated to the set $\Omega_{p_3,t_1} = \left\{(\Omega_{p_3,t_1}^S, \Omega_{p_3,t_1}^C, \Omega_{p_3,t_1}^D); \rho_{p_3,t_1}\right\}$, with:

- $\Omega_{p_3,t_1}^S = -$, which means that no discrete condition is defined,
- $\Omega_{p_3,t_1}^C => 75$, which means that the continuous state of the token in p_3 has to be greater than 75,
- $\Omega_{p_3,t_1}^D = -$, which means that no degradation condition is defined,
- $\rho_{p_3,t_1} = 1$, which means that only 1 token will be consumed during the transition firing.

Definition 2 (Accepted token)**.** *A token h is said to be accepted by an incoming arc if it satisfies either (i) the discrete and continuous conditions of the arc or (ii) the degradation condition of the arc.*

We note $Ha(a_{p,t}, p)$ the set of tokens in the place p that are accepted by the arc $a_{p,t}$.

Definition 3 (Validated arc)**.** *Let $a_{p,t}$ be an arc, $Card(Ha(a_{p,t}, p))$ the number of tokens* accepted *by $a_{p,t}$ in the input place and $\rho_{p,t}$ the weight of the arc. The arc $a_{p,t}$ is said to be validated if $Card(Ha(a_{p,t}, p)) \geq \rho_{p,t}$.*

Outgoing arcs

An arc $a_{t,p} \in A_{t\bullet}$ going from a transition $t \in T$ to a place $p \in P$ carries a set $\Omega_{t,p} \in Jump$:

$$\Omega_{t,p} = \left\{(\Omega_{t,p}^S, \Omega_{t,p}^C, \Omega_{t,p}^D); \rho_{t,p}\right\} \tag{4}$$

where $(\Omega_{t,p}^S, \Omega_{t,p}^N, \Omega_{t,p}^D)$ represents respectively a discrete, a continuous and a degradation assignment and $\rho_{t,p} \in \mathbb{N}^+$ is a weight.

When the $-$ symbol replaces an assignment it means that no change is made to the concerned attribute. By default, $\Omega_{t,p} = \{(-,-,-); 1\}$.

The *discrete assignment* $\Omega_{t,p}^S$ concerns the configurations of the tokens passing through the arc $a_{t,p}$. Let δ_k be the configuration of a token h passing through this arc at time k wearing the event set b_k.

- if $\Omega^S_{t,p} = v$, the event $v \in E$ is concatenated with the current configuration of the token passing through the arc: $b_{k+1} \longleftarrow b_k \cup (v, k+1)$;
- if $\Omega^S_{t,p} = b_{new}$, where b_{new} is a set of timed events, the configuration is completely reset and only contains b_{new}: $b_{k+1} \longleftarrow b_{new}$;
- else if $\Omega^S_{t,p} = -$, the configuration is unchanged: $b_{k+1} \longleftarrow b_k$.

The *continuous assignment* $\Omega^C_{t,p}$ concerns the continuous state of the tokens passing through the arc $a_{t,p}$. Let π_k be the continuous state of a token h crossing the arc at time k. Suppose that π_k carries the value x_k,

- if $\Omega^C_{t,p} = x_{new}$, where x_{new} represents a new numerical value for the token continuous state π_k then: $x_{k+1} = x_{new}$,
- else if $\Omega^C_{t,p} = -$, the continuous state is unchanged: $x_{k+1} = x_k$.

The continuous assignment $\Omega^C_{t,p}$ provides an initial condition for the continuous state of the token passing through the arc, then the set of equations $C_p \in C$ determines the evolution of the continuous state of the token in the output place p.

The *degradation assignment* $\Omega^D_{t,p}$ concerns the degradation state of tokens passing through the arc $a_{t,p}$. Let ϕ_k be the degradation state of a token h crossing the arc at time k and γ_k be the value of ϕ_k,

- if $\Omega^D_{t,p} = \gamma_{new}$, where γ_{new} is a numerical value: $\gamma_{k+1} = \gamma_{new}$,
- else if $\Omega^D_{t,p} = -$, the degradation state is unchanged: $\gamma_{k+1} = \gamma_k$.

The degradation assignment $\Omega^D_{t,p}$ provides an initial condition for the degradation state of the token passing through the arc, then the set of equations $D_p \in D$ determines the evolution of the degradation state of the token in the output place p.

$Post$ is the matrix containing the values of the weights of the outgoing arcs, of dimensions $P \times T$. $Post(t,p)$ corresponds to the weight of the arc connecting the transition t to the place p.

The *weight* $\rho_{t,p}$ defines the number of tokens to be put in the output place of the arc $a_{t,p}$.

For example, in Figure 1(a), the arc a_{t_2,p_2} is associated to the set $\Omega_{t_2,p_2} = \left\{(\Omega^S_{t_2,p_2}, \Omega^C_{t_2,p_2}, \Omega^D_{t_2,p_2}); \rho_{t_2,p_2}\right\}$, with:

- $\Omega^S_{t_2,p_2} = -$, which means that the configuration of the token will not be updated,
- $\Omega^C_{t_2,p_2} = -$, which means that the continuous state of the token will not be updated,
- $\Omega^D_{t_2,p_2} = 0$, which means the degradation of the token going through a_{t_2,p_2} will be set to 0,
- $\rho_{t_2,p_2} = 1$, which means that only one token will be placed in p_2 after the transition firing.

3.4 Firing Rules

Definition 4 (Enabled transition)**.** *A transition $t \in T$ is said to be enabled at time k if all incoming arcs of t are validated (cf. Definition 3):*

$$enabled(t) \equiv (\forall p \; s.t \; a_{p,t} \in A_{\bullet t}, \; Card(Ha(a_{p,t}, p)) \geq \rho_{p,t}) \tag{5}$$

where $Card(Ha(a_{p,t}, p))$ is the number of tokens accepted by the arc $a_{p,t}$, and $\rho_{p,t}$ is the weight defined in the condition set $\Omega_{p,t}$ for $a_{p,t}$.

We recall that $Ha(a_{p,t}, p)$ represents the set of accepted tokens by an arc $a_{p,t}$ in a place p. When $Card(Ha(a_{p,t}, p)) > \rho_{p,t}$, a choice function has to be defined. The function ${}^\bullet\zeta : \mathbb{N}^+ \times Ha \rightarrow H_a$ selects $\rho_{p,t}$ tokens in the place p to be fired among the set $Ha(a_{p,t}, p)$.

The set of accepted and selected tokens in place p to fire a transition t is defined by $\Psi(p,t)$. Let $p \subset P \wedge Pre(p,t) \neq 0$

$$\Psi(p,t) = \begin{cases} {}^\bullet\zeta(\rho_{p,t}, Ha(a_{p,t}, p)) \text{ if } Card(Ha(a_{p,t}, p)) > \rho_{p,t} \\ Ha(a_{p,t}, p) \text{ otherwise} \end{cases} \tag{6}$$

Definition 5 (Set of fired tokens)**.** *From $\Psi(p,t)$, we define $\Psi(\bullet t)$ which is the set of tokens fired by transition t. Let $p_1, p_2, ..., p_i$ be the set of input places of t:*

$$\Psi(\bullet t) = \Psi(p_1, t) \cup \Psi(p_2, t) \cup ... \cup \Psi(p_i, t) \tag{7}$$

A choice function $\zeta^\bullet : \mathbb{N}^+ \times \Psi(\bullet t) \rightarrow \Psi(\bullet t)$ is defined. It determines the tokens that will be kept, duplicated or deleted among the set of fired tokens, and thus put in the output place p of the transition.

Firing of a transition

During a transition firing, the tokens fired by the transition t are moved in the output place of t. The attributes of these tokens are either updated or are unchanged.

The firing of a transition t at time k is formally defined as follows: $\forall \; p \in P$ s.t. $Pre(p,t) \neq 0$ and $\forall \; p' \in P$ s.t. $Post(t,p') \neq 0$,

$$\begin{aligned} M_{k+1}(p) &= M_k(p) - \rho_{p,t} \\ M_{k+1}(p') &= M_k(p') + \rho_{t,p'} \end{aligned} \tag{8}$$

where $\rho_{p,t}$ is the weight carried by the arc connecting the place p to the transition t, $\rho_{t,p'}$ is the weight carried by the arc connecting t to the place p', and $M_k(p)$ is the number of tokens in the place p at time k.

The marking of the HtPN $\mathbb{M}_k(p)$ represents all the tokens in the place p at time k.

$$\begin{aligned} \mathbb{M}_{k+1}(p) &= \mathbb{M}_k(p) \setminus \Psi(p,t) \\ \mathbb{M}_{k+1}(p') &= \mathbb{M}_k(p') \cup \zeta^\bullet(\rho_{t,p'}, \Psi(\bullet t)) \end{aligned} \tag{9}$$

4 Advanced Diagnosis Method

4.1 HtPN-Based Diagnoser

Advanced diagnosis aims at tracking the system current health state.

Definition 6 (Health state)**.** *The health state of a system is the state in which the system is actually functioning, represented through a token and the information it carries: its configuration, its continuous state and its degradation state.*

To track the system health state, we propose to build a diagnoser from a HtPN model. It is important to note that, in the current state of our work, whilst the diagnoser is built from an HtPN model, the use of some notions differs between the HtPN-based diagnoser and the HtPN model. For instance, the parallelism is used in an HtPN to represent different components evolving simultaneously. In the diagnoser, the parallelism is used to represent different diagnosis hypotheses about the system health state. The diagnoser process takes as inputs the sets of discrete and continuous observations on the system. It is based on a two-steps process. The first one is the prediction which aims at

determining the future marking. The second one is the correction, which updates the predicted marking according to new observations. The output of the diagnoser process is an estimation, at any time k, of the system health state that takes the form of the marking of the HtPN-based diagnoser $\Delta_k = \hat{\mathbb{M}}_k$.

Definition 7 (Diagnosis hypothesis)**.** *Each token in the HtPN diagnoser represents a diagnosis hypothesis, i.e. an hypothesis on the health state of the system. It contains available knowledge about the continuous and degradation states of the system at time k and events that have occurred on the system up to time k.*

4.2 Uncertainty Management

Several types of uncertainty are taken into account by using the HtPN formalism.

Knowledge-based uncertainty must be taken into account because the model does not reflect perfectly reality, for the discrete and continuous parts of the model. Due to the inherent imprecision of sensors, we also consider uncertainty about observations. Two types of uncertainty are then considered: the discrete uncertainty dealing with the discrete model and observations; and the continuous uncertainty dealing with the imprecision on the continuous model and continuous values.

Regarding the *discrete aspects*, the discrete model of the system may include discrete uncertainty as impossible or incomplete event sequences. Concerning the discrete observations, an event may occur without being observed: this is a missing observation. Dually, an event may be observed whereas it has not really occurred: this is a false observation.

Discrete uncertainty is managed at two levels in the HtPN-based diagnoser:

- Every discrete condition $\Omega^S_{p,t}$ is replaced by a TRUE ($\top$) condition during the diagnoser generation.
- During the prediction step of the diagnoser process, the diagnoser uses what is called pseudo-firing of transitions like in [8] or [9] to consider the occurrences of each event consistent with the discrete dynamic. Pseudo-firing creates new diagnosis hypotheses.

Transition pseudo-firing duplicates tokens: tokens in the input places of the transition are not moved but *duplicated* and their duplicates are moved in the output places of the transition.

Definition 8 (Transition pseudo-firing)**.** *Let $t \in T$ be an enabled transition, the pseudo-firing of the transition $t \in T$ at time k is formally defined by: $\forall\, p \in P$ s.t. $Pre(p,t) \neq 0$ and $\forall\, p' \in P$ s.t. $Post(t,p') \neq 0$*

$$\begin{aligned} M_{k+1}(p) &= M_k(p), \\ M_{k+1}(p') &= M_k(p') + \rho_{t,p'} \end{aligned} \tag{10}$$

The *continuous uncertainty* is managed thanks to continuous assignments and weights on the outgoing arcs. Thanks to weights, tokens are duplicated and follow continuous dynamics (which can be noisy) associated with the output place they belong to. Following the idea of the SSA algorithm described in [10], the weights of the outgoing arcs of the HtPN are multiplied by n^N_{min} which usually represents the minimum number of tokens to monitor a possible heath state of the system.

4.3 Diagnoser Generation

Let $HtPN_\Phi = \langle P_\Phi, T_\Phi, A_\Phi, Guard_\Phi, Jump_\Phi, E_\Phi, X_\Phi, \Gamma_\Phi, C_\Phi, D_\Phi, \mathbb{M}_{0\Phi}\rangle$ be the system model. The diagnoser is a tuple $HtPN_\Delta = \langle P_\Delta, T_\Delta, A_\Delta, Guard_\Delta, Jump_\Delta, E_\Delta, X_\Delta, \Gamma_\Delta, C_\Delta, D_\Delta, \mathbb{M}_{0\Delta}\rangle$, which is generated from the system model in four steps.

Step 1 consists in copying the HtPN system model, except for $Guard$ and $Jump$ sets. Discrete, continuous and degradation state spaces, as well as continuous and degradation dynamics are the same as those of the model:

$$\begin{aligned} P_\Delta = P_\Phi, T_\Delta = T_\Phi, A_\Delta = A_\Phi, E_\Delta = E_\Phi, X_\Delta = X_\Phi, \\ \Gamma_\Delta = \Gamma_\Phi, C_\Delta = C_\Phi, D_\Delta = D_\Phi, \mathbb{M}_{0\Delta} = \mathbb{M}_{0\Phi}. \end{aligned} \tag{11}$$

Step 2 consists in setting values to the discrete assignation in $Jump_\Delta$ for outgoing arcs of the transitions for whose a discrete condition is defined in $Guard_\Phi$:

$$\Omega^S_{t,p} \leftarrow \Omega^S_{p,t}. \tag{12}$$

This convey the idea that the event associated to a discrete condition should have been observed when the transition is fired. Adding the event to the discrete assignation adds the event to the token configuration, which allow for the comparison with the observed discrete events later on.

Step 3 consists in multiplying the weights of all the outgoing arcs by n^C_{min} to manage continuous uncertainty as explained before: for all $p \in P_\Delta$ and for all $t \in T_\Delta$ such that $a_{t,p} \in A_\Delta$,

$$\rho_{t,p} \leftarrow \rho_{t,p}.n^C_{min} \tag{13}$$

Step 4 consists in assigning values to the conditions in $Guard_\Delta$, corresponding to the conditions associated with the incoming arcs of the transitions: for all $p \in P_\Delta$ and for all $t \in T_\Delta$ such that $a_{p,t} \in A_\Delta$,

$$\Omega^S_{p,t} \leftarrow \top;\ \ \Omega^D_{p,t} \leftarrow \bot \tag{14}$$

All the discrete conditions are then set to TRUE, in agreement with the uncertainty management concerning the occurrences of events. It means that all configurations of tokens in the HtPN-based diagnoser satisfy the discrete conditions. The diagnoser considers at any time the occurrence of each event that can occur from the estimated health mode. All the degradation conditions are set to FALSE in order to disconnect the diagnoser marking evolution from the degradation state. At the moment, the diagnoser follows the degradation but does not use it to estimate fault occurrences. The degradation will be used in future work to perform prognosis. Continuous conditions are the same as those of $HtPN_\Phi$.

The continuous and degradation assignments are not changed during the process.

4.4 Diagnoser Process

The initial marking $\mathbb{M}_0$ of the HtPN-based diagnoser represents the initial health state of the system and is the initial distribution of tokens in the HtPN. Each token h in the initial distribution has three attributes: a configuration with value b^h_0, a continuous state π^h_0 and a degradation state γ^h_0. N_{H0} denotes the initial number of tokens in the HtPN.

From the initial marking and the initial set of discrete and continuous inputs, the diagnoser marking $\mathbb{M}_k$ evolves at time k according to the observations $O_k = O^S_k \cup O^C_k$, where O^S and O^C respectively represent the set of discrete observations and the set of continuous observations. The

estimated marking at time k denoted $\hat{\mathbb{M}}_k$ represents all the possible diagnosis hypotheses on the system health state at time k.

The marking evolution in the HtPN-based diagnoser is based on two steps, prediction and correction, which combine the transition pseudo-firing, particle filters and an algorithm called the Stochastic Scaling Algorithm (SSA), already presented in [10]. In particle filtering, the number of particles defines the precision of the filter. Here the particles are represented by tokens: each token is one particle. The goal of the SSA is to avoid the combinatory explosion and to limit the number of tokens at each step of the algorithm.

The prediction step of the online diagnoser process aims at determining the future marking of the diagnoser $\hat{\mathbb{M}}_{k+1|k}$. It is based on the firing of the enabled transitions and on the update of the token values. All the enabled transitions are fired according to the rules described in Section 3.4. This implies the assumption that a single event can occur at time k. The event set b_k of a configuration δ_k evolves through the discrete assignment described in Section 3.3. The value of the continuous state of the token evolves according to the dynamics of the place the token belongs to after the transition firing if the continuous dynamics exists, (as well as the value of its degradation state if the degradation equation exists). It is possible for equations to contain noise. Noise in equations is considered to take into account uncertainty about model continuous dynamics: the duplicated tokens have different state values because of the random noise function in continuous dynamics.

The correction step of the diagnoser process updates the predicted marking $\hat{\mathbb{M}}_{k+1|k}$ to the estimated marking $\hat{\mathbb{M}}_{k+1|k+1}$ according to new observations O_{k+1}.

It is based on the computation of scores of all hypotheses contained in the predicted marking and on the resampling of the tokens.

Scores of hypotheses are calculated with two quantities Pr^S and Pr^C representing the gaussian probability distributions over the discrete and the continuous states, respectively. Pr^S gives the *configuration weight* and is computed as the inverse of exponential of the distance between the configuration event set and $O^-_{k+1} = \{O_\kappa | \kappa \leq k+1\}$, the set of discrete observations until $k+1$. Pr^C gives the normalized state weight and is calculated according to the distance between the token continuous state value and continuous observations O^C_{k+1}.

Then, the score of one hypothesis $h_k =< \delta_k, \pi_k, \gamma_k >$ at time k is computed using a weighted function of the sum of its configuration and its continuous state weights:

$$Score(h_k) = \alpha \times Pr^S(\delta_k) + (1-\alpha) \times Pr^C(\pi_k), \quad (15)$$

where $\alpha \in [0,1]$ is the coefficient indicating the global confidence of the discrete part relatively to the continuous part. The score of a hypothesis is always between 0 and 1.

The set of particles/tokens is then resampled according to their scores. Like in classical particle filtering, some hypotheses are deleted, others are duplicated if they have an high score. The parameter n^N_{max} is the maximum number of particles available to monitor all hypotheses. The total number of particles after the resampling is always less than or equal to n^N_{max}. In particle filtering, the number of particles defines the precision of the filter but is also a computational performance factor, then n^N_{max} can be set up to fulfill performance constraints.

The diagnosis Δ_k is deduced from the marking of the HtPN-based diagnoser $HPPN_\Delta$ at time k:

$$\Delta_k = \hat{\mathbb{M}}_k \quad (16)$$

It represents all diagnosis hypotheses as a distribution of beliefs over the current health state and how this health state has been reached. In other words, the marking $\hat{\mathbb{M}}_k$ indicates the belief over the fault occurrences and the continuous and degradation states. The HtPN-based diagnoser results include the results of a classical diagnoser in terms of fault occurrences. In a classical diagnoser, however, every diagnosis hypotheses has the same belief degree. A HtPN-based diagnoser handles more uncertainty and evaluates the ambiguity according to the tokens places and values.

5 Applications

To illustrate how the proposed diagnostic method can deal with different kinds of system we propose to apply it on two systems of different types: a discrete event system and a hybrid system. A new software, named HeMU (Heterogeneous systems Monitoring under Uncertainty) has been developed in Python. Computations are done on Linux, with an Intel(R) Core(TM) i5-9400H CPU 2.50GHz. Both systems were modeled in the proposed HtPN framework. Results of simulations and diagnosis are explained in this section.

5.1 Discrete Event System Application

The studied discrete event system (DES) is illustrated in Figure 2. This DES is represented by an automaton $A = \langle Q, \Sigma, \delta, x_0 \rangle$, where:

- $Q = \{Nom1, Nom2, Deg1, Deg2\}$ is the set of discrete states;
- $\Sigma = \{o_1, o_2, o_3, f\}$ is the set of events;
- δ represents the transition function ;
- $q_0 = Nom1$ is the initial state.

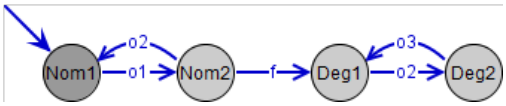


Figure 2: Example of a Discrete Event System

Some observable and fault events are defined for diagnosis purpose. The set of observable/measured events is $\Sigma_o = \{o_1, o_2, o_3\} \subset \Sigma$ and the unobservable fault event to be diagnosed is $f \in \Sigma_{uo} \subset \Sigma$.

Model and simulation

The HtPN representation for this automaton was created with the HeMU software. For this application, no continuous dynamics nor degradation have been filled in the model. Only the places and discrete conditions were specified.

The results of the HtPN model simulation is presented in Figure 3. From initial state $Nom1$, the following event sequence occurs: $o_1.f.(o_2.o_3)^*$. It means that fault f was injected after the observation o_1, and after its occurrence, observations o_2 and o_3 are repeated until the end of the simulation.

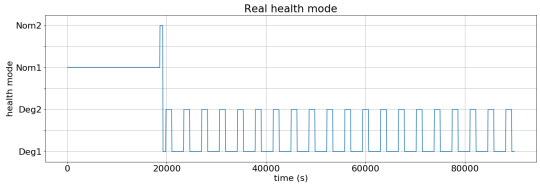


Figure 3: Health mode simulation for the DES

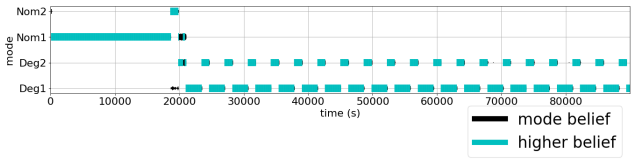


Figure 4: Diagnosis for the DES

Diagnosis

Results obtained by applying the proposed HtPN-based diagnosis method are illustrated by Figure 4.

It gives all the diagnosis hypotheses on the possible system states. Blue lines indicate the hypotheses with the highest belief. State $Nom2$ is identified after the observation o_1. As it cannot be known whether the fault has yet occurred, a small belief is present on state $Deg1$ (as shown by the small black line). Then two diagnosis hypotheses with almost the same belief (big black line with appearance of blue for Nom1 and a majority of blue with appearance of black for Deg2) are computed after observing the event o_2: $Nom1$ and $Deg2$. There is an ambiguity on the system's state because f is not diagnosable. f is diagnosed without ambiguity after observing o_3, the diagnoser is in the state $Deg1$.

5.2 Hybrid System Application

The studied hybrid system is composed of three tanks connected in series by two valves. The system behavior and the multimode description are described in detail in [11]. To sum up, a pump delivers a constant water flow q_1 in the first tank and a second tank empties with an output flow q_{20}. The system has to maintain a water level l_2 in the second tank greater than l_{2min}. A multimode description of the health evolution of the water tank system is presented in Figure 5.

Ten behavioral modes (i.e. discrete states) are identified. To each mode are associated continuous dynamics C_1, C_2, C_3, C_4 and degradation dynamics D_1, D_2, D_3, D_4, D_5. The continuous variables are the water levels in the tanks and the degradation variables give the probability of anticipated faults in the system. Equations for continuous and degradation dynamics can be found on the web[1].

Valves are controlled by discrete and observable input signals: $open_{v_{13}}$, $close_{v_{13}}$, $open_{v_{32}}$ and $close_{v_{32}}$. Six faults, considered as non observable discrete events, may occur on the system: f_1, f_2 and f_3 represent leaks in each tank, f_4 and f_5 represent v_{13} and v_{32} stuck in closed position, and f_0 corresponds to a water level l_2 below l_{2min} leading to the system failure. In order to simplify the illustration, only the events associated to the valve v_{13} and the faults f_1, f_4 and f_0 are considered in this representation.

[1] https://homepages.laas.fr/echanthe/hymu/water_tanks

Model and simulation

The HtPN model was created from this multimode representation. To the ten behavioral modes correspond 10 places containing equations for continuous and degradation dynamics. Model transitions are associated either to discrete event occurrences or tests on continuous or degradation variables. Simulation of the HtPN model of the water tank system is presented in Figure 6.

In the initial mode Nom_1, the valves v_{13} and v_{32} are open, the continuous state representing the water levels in the tanks is $x_0 = [l_{10}, l_{20}, l_{30}] = [0.60, 0.55, 0.58]$ and the degradation state is $d_0 = [\mathrm{p}_{10}, \mathrm{p}_{20}, \mathrm{p}_{30}, \mathrm{p}_{40}, \mathrm{p}_{50}] = [0, 0, 0, 0, 0]$. The water flow q_1 delivered by the pump is constant. After 310 minutes of operation, v_{13} is closed every hour during 20 min in order to perform a water treatment in $Tank_1$. Fault f_1 is injected at 201840s and f_0 occurs at 206040s.

Diagnosis

Diagnosis result for the water tank is given by Figure 7. It gives all hypotheses on the possible system health mode. Blue lines indicates the diagnosis hypotheses with the highest belief. Although not represented here, the degradation value of each hypothesis is available. These diagnosis results are consistent with the actual health state of the simulated system.

If the maximum number of tokens in the HtPN-based diagnoser at any time is set to 200, the execution time of the proposed diagnosis method is 4 minutes 22 seconds. If the maximum number of tokens is 400, the execution time is 7 minutes 47 seconds.

Although this section focuses on only two examples of discrete and hybrid systems, a purely continuous system could also have been simulated. It would have been sufficient to define a single place containing continuous dynamics. But for diagnosis purpose, at least one discrete event corresponding to the fault to be diagnosed would have had to be defined.

6 Conclusion and Future Work

This article presents a holistic advanced diagnosis approach for systems under uncertainty. This approach is based on a formalism based on Petri Nets named HtPN. HtPN can be use for specifying various systems (DES, CS, HS) and diagnosing them for health monitoring purposes. This formalism makes possible to take into account the uncertainties on the modeling and on the observations of the system. In terms of health monitoring, a diagnostic method based on HtPN has been proposed. It allows not only to manage uncertainties on the observations, but also the monitoring of the current state of the system as well as its degradation state. Applications on a pure Discrete Event System and on a Hybrid System prove that the modeling framework and the monitoring method are relevant and efficient for different types of systems taking into account uncertainties.

Future work will focus on the development of a prognostic method based on HtPN and on computational complexity management.

References

[1] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio

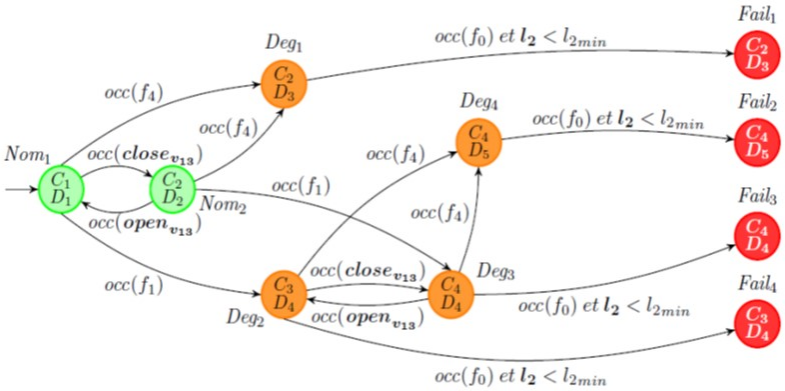


Figure 5: Multimode description of the hybrid water tank system.

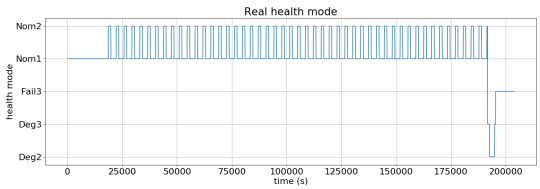


Figure 6: Health mode simulation for the Hybrid System

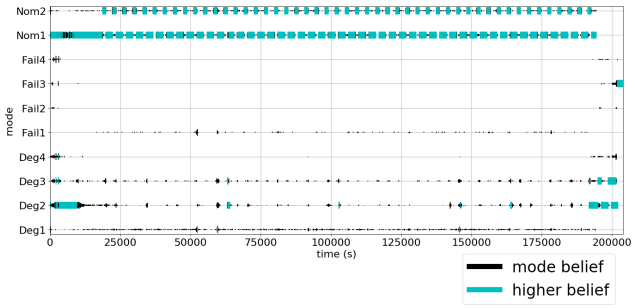


Figure 7: Diagnosis for the Hybrid System

Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 138(1):3–34, 1995.

[2] Hassane Alla and René David. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, 8(01):159–188, 1998.

[3] C Valentin-Roubinet. Hybrid systems modelling: mixed Petri nets. In *IEEE Conference CSCC*, volume 99, 1999.

[4] Luis Alejandro Cortés, Petru Eles, and Zebo Peng. A Petri net based model for heterogeneous embedded systems. In *Proc. NORCHIP Conference*, pages 248–255, 1999.

[5] Pauline Ribot, Yannick Pencolé, and Michel Combacau. Generic characterization of diagnosis and prognosis for complex heterogeneous systems. *International Journal of Prognostics and Health Management*, 4, 2013.

[6] Sriram Narasimhan and Gautam Biswas. Model-based diagnosis of hybrid systems. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans*, 37(3):348–361, 2007.

[7] Quentin Gaudel, Elodie Chanthery, Pauline Ribot, and Matthew J. Daigle. Diagnosis of hybrid systems using Hybrid Particle Petri nets: theory and application on a planetary rover. In Moamar Sayed-Mouchaweh, editor, *Fault Diagnosis of Hybrid Dynamic and Complex Systems*, pages 209–241. Springer Verlag, 2018.

[8] Charles Lesire and Catherine Tessier. Particle Petri nets for aircraft procedure monitoring under uncertainty. In *Applications and Theory of Petri Nets*, pages 329–348. Springer, 2005.

[9] L Zouaghi, A Alexopoulos, A Wagner, and E Badreddin. Modified particle Petri nets for hybrid dynamical systems monitoring under environmental uncertainties. In *IEEE/SICE Int. Symposium on System Integration*, pages 497–502, 2011.

[10] Pauline Ribot, Elodie Chanthery, Quentin Gaudel, and Matthew J Daigle. Hybrid particle Petri net based prognosis of a planetary rover. *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[11] Pauline Ribot, Elodie Chanthery, and Quentin Gaudel. HPPN-based Prognosis for Hybrid Systems. In *Annual Conf. of the PHM Society*, St. Petersburg, United States, 2017.