HPPN-based Prognosis for Hybrid Systems

Pauline Ribot[1], Elodie Chanthery[2], and Quentin Gaudel[3]

[1,2] *LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS, Toulouse, France*
pauline.ribot@laas.fr
elodie.chanthery@laas.fr

[3] *Easymile, 8 Chemin des Trente-Six Ponts 31400 Toulouse, France*
quentin.gaudel@easymile.com

Abstract

This paper presents a model-based prognosis method for hybrid systems i.e. that have both discrete and continuous behaviors. The current state of the hybrid system is estimated by a diagnosis process and the prognosis process uses this state estimation to predict the future states and to determine the end of life (EOL) or the remaining useful life (RUL) of the system. The Hybrid Particle Petri Nets (HPPN) formalism is used to model the hybrid system behavior and degradation. A HPPN-based diagnoser has already been defined to provide a current state estimation that takes uncertainty about the system model and observations into account. We propose to generate a prognoser from the HPPN model of the system. This prognoser is initialized and updated with the result of the HPPN-based diagnoser. It computes a distribution of beliefs over the future mode trajectories of the system and predicts the system RUL/EOL. The prognosis methodology is demonstrated on a three tanks example.

1. Introduction

Recent industrial systems have become so complex that explaining their behaviors is often intractable for humans, especially when they are exposed to failures. Prognostics and Health Management (PHM) aims at developing tools that can support maintenance or repair tasks by reducing the global costs due to unavailability and repair actions, but it can also optimize the mission reward by replanning or reconfiguring the system. An efficient health monitoring technique is required to detect, isolate (diagnosis) and predict faults (prognosis) leading to failures. Prognosis is the prediction of the system future states and of the times of the occurrences of the faults that lead to these states. Most of the time, it is related to the determination of the system's End Of Life (EOL), which is the time when the system is not operational anymore, and of the Remaining Useful Life (RUL) that is the remaining period before it reaches its end of life (Engel, Gilmartin, Bongort, & Hess, 2000).

A system is considered as hybrid if it exhibits both discrete and continuous dynamics. Hybrid systems are usually described as a system of multiple modes that represents its continuous evolutions (continuous dynamics) under different operational conditions (Bayoudh, Travé-Massuyes, & Olive, 2008). This mode representation is convenient to model systems that have specific continuous dynamics for each mode, but it cannot model dynamics depending not only on both the discrete state and the continuous state, but also on the set of events that occurred on the system. Such dynamics are used to model particular phenomena, like system degradation. In most industrial systems, for example, if the degradation is not observable, it is estimated as fault occurrence probabilities. The degradation thus depends on the stress level associated with the current mode but, in some cases, also relies on the analysis of the set of events that occurred on the system. The failure probability of a component could depend on the time spent in the different modes, particularly when the component is used for critical actions. Then the system degradation has to be considered as a characteristic that depends on both the discrete and continuous dynamics of the hybrid system. The evolution of this characteristic is defined by what we call the degradation dynamics. The system degradation state is the current value of this degradation dynamics. In order to clarify the model of the system, we extend the multi-mode representation by associating underlying degradation dynamics with each mode. The definition of a mode is thus enriched and is a combination of a discrete state of the Discrete Event System (DES) with continuous dynamics and degradation dynamics (Gaudel, Chanthery, & Ribot, 2014). The state of the hybrid system is defined as the combination of its discrete, continuous and degradation states. Finally, as we work on real complex systems, we consider that all the variables are subject to uncertainty.

Our previous works introduced a modeling framework called Hybrid Particle Petri Nets (HPPN). In (Gaudel et al., 2014), we proposed to use HPPN to model a hybrid system, and we tracked its current state under uncertainty by generating a diagnoser. A diagnoser is defined with the HPPN framework and uses the observations (system inputs and outputs) on the system to compute a diagnosis. The system diagnosis is a distribution of beliefs over its past mode trajectories that include its possible current states and the fault occurrences. It also includes the system degradation estimation, which is a significant advantage to perform prognosis. In (Gaudel, Chanthery, & Ribot, 2015), this diagnosis approach was applied on a simulated three-tank system.

The contribution of this paper compared to previous works is the introduction of a prognoser. The health monitoring approach is enriched with a prognoser that aims at computing the prognosis of the system under uncertainty, based on the current diagnosis and future inputs. Like the diagnoser, the prognoser is a HPPN generated from the system HPPN-based model. The system prognosis is a distribution of beliefs over its future mode trajectories and its RUL/EOL.

This paper is organized as follows. Section 2 gives some related work. Section 3 recalls the HPPN framework and the methodology to model the hybrid system behavior and degradation by using HPPN. Section 4 details the prognosis method. Section 5 provides simulation results obtained by testing the proposed method on the three-tank case study. Conclusions and future works are discussed in the final section.

2. RELATED WORK

Hybrid systems are the core interests of numerous researches in many areas, such as modeling, verification, control, and monitoring. In system modeling, different structures have been introduced to represent hybrid dynamics: hybrid automata (HA), hybrid bond graphs (HBG), hybrid Petri nets (HPN), Partially Observed Petri Nets (POPN), etc. Such models have largely been used or extended for hybrid system diagnosis. Some works particularly focus on hybrid system diagnosis with the intent to use it for prognosis purposes, and then generally consider degradation monitoring. However, most of these models do not take into account uncertainties about the model (errors in model parameter estimation for example) or observations (sensor errors for example). Table 1 presents some references to these works and sums up the related work for the prognosis task.

In (Vianna & Yoneyama, 2015), the diagnosis monitors the system behavior and its degradation in order to have a better estimate as a start for the prognosis process via the Interactive Multiple-Model (IMM) algorithm, but the approach is limited to continuous systems. With the same purpose, (Gaudel et al., 2014) uses HPPN to monitor the hybrid system degradation in addition to its behavior, considering many sources of uncertainty.

In (Chanthery & Ribot, 2013), hybrid automata are used to model the system and to generate a prognoser that determines the system RUL, whenever a new diagnosis is available. The method is demonstrated on a simulated study case in (Zabi et al., 2013). The behavior and the degradation are monitored, but future actions are not considered in the prognosis process, making the approach limited.

The authors of (Yu et al., 2014) develop a model-based sequential failure prognosis for hybrid systems where some faults are not detectable immediately. Dynamic fault isolation is performed with Hybrid Bond graph. A mode dependent fault signature matrix is proposed and a waiting time is used to allow all faults to exhibit their symptoms on residuals, especially faults that are only detected with continuous signals. No detail is given concerning the duration of the waiting time. The degradation behavior of each component is mode dependent and estimated by a hybrid differential evolution algorithm. The RUL of the component is computed by the estimation of the degradation and a threshold. A sequential prognosis algorithm, including a standard prognosis module and auxiliary module, is proposed. The standard prognosis module is activated once one inconsistency is detected and is based on the set of suspected faults through fault isolation, whereas the auxiliary module is triggered when a new mode change occurs during the standard prognosis and is based on the ”true” faults, predicted by the standard prognosis. These works do not take into account uncertainty about the observation and the model.

The work of (Daigle et al., 2015) extend the model-based prognostics paradigm to hybrid systems. It relies on previously established methods for hybrid state estimation, and provides an approach to predict RUL/EOL given a hybrid model, a state estimate, and a specification of future input

Table 1. Related Work on Prognosis of Hybrid Systems.

Modeling	References
HA	(Chanthery & Ribot, 2013; Zabi, Ribot, & Chanthery, 2013)
HBG and ARR	(Yu, Wang, & Luo, 2014)
Conflict and Monte-Carlo	(Daigle, Roychoudhury, & Bregon, 2015)
Generic caracterization	(Ribot, Pencolé, & Combacau, 2013)
POPN (no RUL)	(Basile, Chiacchio, & Tommasi, 2009)

uncertainty. It describes how the resulting probability distribution for RUL/EOL may be multi-modal due to mode-switching in the predicted future behavior. Petri nets-based approaches often deal with the prediction of event occurrences from a predictability perspective, i.e. the system monitoring indicates either a fault can still occur, or not.

In (Basile et al., 2009), a generalized marking is used to consider unobservable event occurrences while minimizing the state space explosion, whilst the problem is approached with Partially Observed Petri Nets (POPN) in (Lefebvre, 2014). These works do not however provide any quantitative information about RUL estimations.

In conclusion, the best way to evaluate the state of hybrid systems under uncertainties seems to use Petri Nets. The *Hybrid Particle Petri Nets* (HPPN) have the particularity to add the hybrid point of view and to deal with various types of uncertainty. The main contribution of the paper is to propose a complete methodology for prognosis of hybrid systems under uncertainty, illustrated by a case-study.

3. SYSTEM HEALTH MODELING WITH HPPN

The *Hybrid Particle Petri Nets* (HPPN) are an extension of Petri nets. They have been introduced in (Gaudel et al., 2014). We recall here the main concepts so as they can be understood without reading the earlier publications.

Hybrid systems are usually described with a mode representation. In this work, we define a mode by the combination of one continuous dynamic, one operational condition and one degradation dynamic, which are represented with three kinds of places in a HPPN. Symbolic places represent the different discrete health states of the system. Continuous (resp. degradation) dynamics are associated with numerical (resp. degradation) places. One place can thereby be part of several mode representations. Two different modes, however, cannot share the same two symbolic and numerical places.

A Hybrid Particle Petri Net is defined as a tuple $\langle P, T, A, \mathcal{A}, E, X, \mathtt{D}, \mathcal{C}, \mathcal{D}, \Omega, M_0 \rangle$ where:

- P is the set of places, partitioned into numerical places P^N, symbolic places P^S and degradation places P^D,
- T is the set of transitions,
- $A \subset P \times T \cup T \times P$ is the set of arcs,
- $\mathcal{A}$ is the set of arc annotations,
- E is the set of event labels,
- $X \subset \mathbb{R}^{n_N}$ is the state space of the continuous state vector, with $n_N \in \mathbb{N}_+$ the number of continuous state variables,
- $\mathtt{D} \subset \mathbb{R}^{n_D}$ is the state space of the degradation state vector, with $n_D \in \mathbb{N}_+$ the number of degradation state variables,

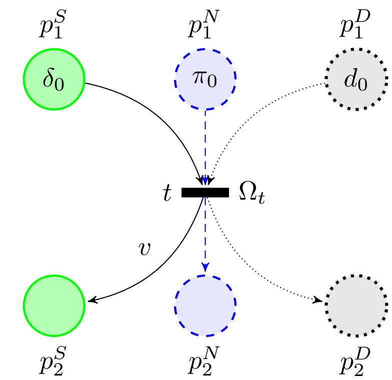

Figure 1. Example of a simple HPPN at time $k = 0$.

- $\mathcal{C}$ is the set of dynamic equation sets associated with numerical places, representing continuous dynamics,
- $\mathcal{D}$ is the set of dynamic equation sets associated with degradation places, representing degradation dynamics,
- Ω is the set of conditions associated with transitions,
- M_0 is the initial marking of the HPPN, it represents the system initial conditions.

An example of a simple HPPN is illustrated in Figure 1. Symbolic places are represented by plain circles, numerical and degradation places are represented by discontinuous and dotted circles. Transitions are represented by black lines. Arcs connecting transition and symbolic places (resp. numerical and degradation places) are represented by plain arrows (resp. discontinuous and dotted arrows).

Symbolic places model the discrete states of the system and are marked by configurations. A configuration $\delta_k \in M_k^S$ is a symbolic token at time k, whose value is the set b_k of events that occurred on the system until time k:

$$b^k = \{(v, \kappa) | \kappa \leq k\}. \tag{1}$$

The set of event labels $E = E_o \cup E_{uo}$ is partitioned into observable (E_o) and unobservable (E_{uo}) event labels. An event is defined as a couple $e = (v, k)$ where $v \in E$ is an event label and k the time of occurrence of e. An anticipated fault is represented by an unobservable event label $f \in E_{uo}$.

A numerical place $p^N \in P^N$ is associated with a set of equations $C \in \mathcal{C}$ modeling system continuous dynamics and its corresponding model noise and measurement noise:

$$C : \begin{cases} x_k &= \mathbf{f}(k, x_{k-1}, u_k) + \mathbf{v}(k, x_{k-1}, u_k) \\ y_k &= \mathbf{h}(k, x_k, u_k) + \mathbf{w}(k, x_k, u_k) \end{cases}, \tag{2}$$

where $x_k \in X$ is the continuous state vector, $u_k \in \mathbb{R}^{n_u}$ is a vector of n_u continuous input variables, $\mathbf{f}$ is the noiseless continuous state equation, $\mathbf{v}$ is the continuous process noise equation, $y_k \in \mathbb{R}^{n_y}$ is a vector of n_y continuous output variables, $\mathbf{h}$ is the noiseless continuous output equation, and $\mathbf{w}$ is the continuous output noise equation. Numerical places are marked with particles. A particle $\pi_k \in M_k^N$ is a numerical token at time k whose value is a possible continuous state $x_k \in X$ of the system at time k. The notion of particle is central in the HPPN structure as it represents an imprecise knowledge on the continuous states of the system. For example, for the diagnoser object, particles represents the results of a particle filter method.

A degradation place $p^D \in P^D$ is associated with a set of equations $D \in \mathcal{D}$ modeling system degradation dynamics:

$$D : \begin{cases} \mathtt{d_k} = \mathbf{g}(k, \mathtt{d}_{k-1}, b_{k-1}, x_{k-1}, u_k) + \\ \qquad \mathbf{u}(k, \mathtt{d}_{k-1}, b_{k-1}, x_{k-1}, u_k) \end{cases}, \qquad (3)$$

where $\mathtt{d}_k \in \mathtt{D}$ is the degradation state vector, $\mathbf{g}$ is the noiseless degradation state equation and $\mathbf{u}$ is the degradation process noise equation. Degradation places are marked with degradation tokens. A degradation token $d_k \in M_k^D$ links a configuration δ_k and a particle π_k, and its value is a possible degradation state $\mathtt{d}_k \in \mathtt{D}$ of the system at time k.

The marking M_k of a HPPN at the discrete time k is composed of tokens distributed in symbolic, numerical or degradation places: $M_k = M_k^S \cup M_k^N \cup M_k^D$.

A hypothesis on the system is a set of tokens linked together that represent the system state and mode trajectory. A hypothesis contains only one configuration but it can contain many particles and degradation tokens representing an imprecise knowledge on the continuous and degradation states, e.g. $\{\delta_k^1, \pi_k^1, ..., \pi_k^{n_k}, d_k^1, ..., d_k^{n_k}\}$, where $n_k \in \mathbb{N}_+$ represents the precision of the hypothesis at time k, and where the n_k degradation tokens link the n_k particles to δ_k^1. The links between tokens are used to represent the possible modes of the system; they are specific to HPPN. In case of ambiguity, the initial marking could contain several hypotheses: for example, $M_0 = \{\delta_0^1, \pi_0^1, d_0^1\} \cup \{\delta_0^2, \pi_0^2, d_0^2\}$, where d_0^1 links δ_0^1 and π_0^1, and d_0^2 links δ_0^2 and π_k^2.

Transitions model changes of modes and their conditions describe the circumstances of the changes. It means that any transition $t \in T$ must have three places (one of each type) in its sets of input places and three places in its set of output places. A set of three conditions $\Omega_t = \{\omega^S, \omega^N, \omega^D\}$ is associated to any transition t. A condition $\omega : M_k \to \mathbb{B}$, with $\mathbb{B} = \{\top, \bot\}$ (the set of the logical values TRUE and FALSE), can be either a test on a token value, always satisfied ($\top$), or never satisfied ($\bot$). A symbolic condition ω^S can thus either be $\top$ or $\bot$, or it can test the occurrence of an event labeled with $v \in E$ (observable or unobservable event; fault, mission or command event). In that last case, it takes the form $\omega^S(\delta_k) = occ(b_k, v)$, to test if the event set b_k of δ_k contains the event (v, k). A numerical condition ω^N (resp. degradation condition ω^D) can be $\top$ or $\bot$ or be a constraint on the continuous state (degradation state). In the last case, $\omega^N(\pi_k) = c(x_k)$ is a test on the continuous state vector x_k of π_k.

If ω^S and ω^N are conditions that should be satisfied at the same time, ω^D is an alternative condition of the mode change. In a general way, the system changes of mode if $(\omega^S \wedge \omega^N) \vee \omega^D$ is satisfied. Degradation dynamics are used to model the system degradation and degradation conditions are used to let the degradation state affect the system behavior. For example, if the degradation is modeled by a fault occurrence probability, a degradation condition could be a boolean function satisfied if the probability is higher than a predefined threshold selected by the user.

A transition $t \in T$ may be fired if there is at least one token in any of its input places and with respect to its associated conditions $\Omega_t \in \Omega$. If a transition t is fired, the tokens satisfying conditions Ω_t are not consumed like in classical Petri nets, but moved and so their links are preserved.

An arc $a \in A$ that connects a transition t to a symbolic place p^S, may be annotated with an event label $v \in E$. In such a case, a configuration δ that is moved to p^S after the firing of t at time k, sees its event set b updated with the event (v, k). Consequently, annotations $\mathcal{A}$ change the configuration values during transition firing.

4. HYBRID SYSTEM PROGNOSIS

4.1. Overview of the Health Monitoring Method

Prognosis aims at predicting the system future states and its RUL/EOL, by using the current diagnosis and future inputs available from a mission scenario for example. Particularly, the goal is to determine if and when the system will enter a failure mode and will not be operational during an arbitrary prediction horizon τ_p.

The HPPN framework is used to define three different objects: a model of the hybrid system, a diagnoser and a prognoser. An overview of the health monitoring method is given in Algorithm 1 and illustrated in Figure 2.

Algorithm 1 HPPN-based monitoring methodology

```
1: HPPN_Φ ← CreateHPPNModel()
2: HPPN_Δ ← GenerateHPPNDiagnoser(HPPN_Φ)
3: HPPN_Π ← GenerateHPPNPrognoser(HPPN_Φ)
4: for all k do
5:     O_k ← (U_k^S, u_k^N, Y_k^S, y_k^N)
6:     Δ_k ← Update(HPPN_Δ, k, O_k)
7:     Π_k ← Prognose(HPPN_Π, Δ_k, U_k^+)
8: end for
```

The first offline step is the modeling of the hybrid system in the HPPN framework. The system model $HPPN_\Phi$ can be directly built from a multimode description or created from expert knowledge. The second offline step (line 2) is

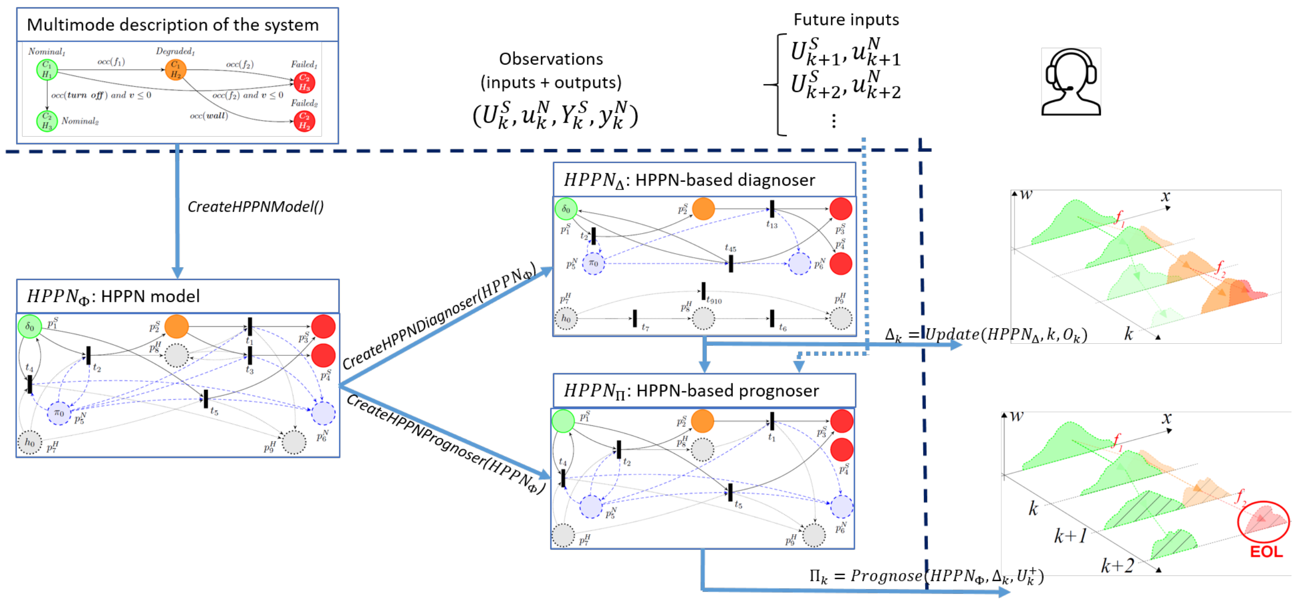

Figure 2. Overview of the health monitoring method.

the generation of a HPPN-based diagnoser $HPPN_{\Delta}$ from the system model. These two previous steps were developed in (Gaudel et al., 2014) to perform diagnosis of hybrid systems under uncertainty. The HPPN-based diagnoser uses the discrete and continuous observations (system inputs and outputs) to compute a diagnosis Δ_k. The diagnosis Δ_k is given by the marking of the HPPN-based diagnoser $HPPN_{\Delta}$ that represents a distribution of beliefs obtained by particle filtering. This marking contains all diagnosis hypotheses. A diagnosis hypothesis is a set of tokens linked together through degradation tokens, which represent possible current states of the system and past mode trajectory at time k, e.g. $\{\delta_k^1, \pi_k^1, ..., \pi_k^{n_k}, d_k^1, ..., d_k^{n_k}\}$, where $n_k \in \mathbb{N}_+$ represents the precision of the diagnosis hypothesis at time k, and where the n_k degradation tokens link the n_k particles to δ_k^1.

The last offline step (line 3) of the health monitoring method is the generation of a HPPN-based prognoser $HPPN_{\Pi}$ from the system model. Then the online process (line 4-8) uses the system consecutive observations O_k (discrete and continuous inputs and outputs) first to update the diagnoser marking and compute the diagnosis Δ_k. Finally, the system prognosis Π_k at time k is computed (line 7) from diagnosis Δ_k and discrete and/or continuous future inputs U_k^+.

All these steps of the health monitoring method and the created objects (model, diagnoser and prognoser) are illustrated in Figure 2.

4.2. Prognoser Generation

Let us consider a system model $HPPN_{\Phi} =$

$$\langle P_{\Phi}, T_{\Phi}, A_{\Phi}, \mathcal{A}_{\Phi}, E_{\Phi}, X_{\Phi}, \mathtt{D}_{\Phi}, \mathcal{C}_{\Phi}, \mathcal{D}_{\Phi}, \Omega_{\Phi}, M_{0\Phi}\rangle. \quad (4)$$

The prognoser of $HPPN_{\Phi}$ is defined by $HPPN_{\Pi} =$

$$\langle P_{\Pi}, T_{\Pi}, A_{\Pi}, \mathcal{A}_{\Pi}, E_{\Pi}, X_{\Pi}, D_{\Pi}, \mathcal{C}_{\Pi}, \mathcal{H}_{\Pi}, \Omega_{\Pi}, M_{0\Pi}\rangle \quad (5)$$

and is generated with the following steps.

1. The prognoser has to simulate the health evolution of the system in the future. Its places, event label set, and state spaces are the same as those of the HPPN model: $P_{\Pi} = P_{\Phi}$, $E_{\Pi} = E_{\Phi}$, $X_{\Pi} = X_{\Phi}$, $\mathtt{D}_{\Pi} = \mathtt{D}_{\Phi}$.

2. Model varying parameters (unknown parameters) are considered as state variables and are estimated during the diagnoser process. Even if the model parameters do not constitute the only source of uncertainty of a model-based prognosis, we consider that state process noise during the prognosis process can be neglected. C_{Φ} and D_{Φ} modeling continuous and degradation dynamics are thus defined as follows:

$$C_{\Pi} : \begin{cases} x_k &= \mathbf{f}_{\Phi}(k, x_{k-1}, u_k) \\ y_k &= \mathbf{h}_{\Phi}(k, x_k, u_k) \end{cases}, \quad (6)$$

where $\mathbf{f}_{\Phi}$ and $\mathbf{h}_{\Phi}$ are the continuous state and output equations of C_{Φ}.

$$D_{\Pi} : \{\ \mathtt{d}_k = \mathbf{g}_{\Phi}(k, \mathtt{d}_{k-1}, b_{k-1}, x_{k-1}, u_k)\ , \quad (7)$$

where $\mathbf{g}_{\Phi}$ is the degradation state equation of D_{Φ}.

3. Finally, the prognosis process uses conditioned firing, but conditions on the occurrences of unobservable or measured events cannot be satisfiable by simulation in the future. The prognoser works with only input variables or events. Consequently, if a symbolic condition ω^S tests the occurrence of an event that is not a discrete input, it becomes $\top$, in this way the prediction process is not blocked. If both ω^S and ω^N are $\top$, they become $\perp$, so $\omega^S \wedge \omega^N$ can never be satisfied. The conditions on the degradation state can be used, however, to simulate the future occurrences of events, such as fault events, so they are not changed. This means it is possible to compute the system prognosis even if the future actions are not known, as long as time affects the system state. At this step, the arc annotations are conserved to record the set of simulated events in configurations event sets: $\mathcal{A}_\Pi = \mathcal{A}_\Phi$.
4. If all conditions in Ω_t are $\perp$, i.e. it is no significant information for the prediction, the transition t is useless and removed of the net. The set of transitions T_Π is defined as:
$$T_\Pi = T_\Phi \setminus \{t/\Omega_t = \{\perp, \perp, \perp\}\} \quad (8)$$
The sets of conditions Ω_Π, arc annotations $\mathcal{A}_\Pi$, and the set of arcs A_Π are thus reduced accordingly.

4.3. Prognosis Process

The prognosis process steps are given in Algorithm 2 and detailed in the section.

Algorithm 2 Prognose

Input: $HPPN_\Pi, \Delta_k, U_k^+$
Output: $\hat{M}_{k_{\mathrm{EOP}}|k}$

```
1: Initialize(HPPN_Π, Δ_k)
2: for all U_κ ∈ U_k^+ do
3:     if OneHypothesisInNonFailureMode(HPPN_Π) then
4:         M̂_{κ|k} ← Update(HPPN_Π, κ, U_κ)
5:     else
6:         break for loop
7:     end if
8: end for
9: k_EOP ← κ
```

The function $Prognose$ takes as input the prognoser $HPPN_\Pi$ generated from the system model $HPPN_\Phi$, the current diagnosis Δ_k and a set of available future discrete and continuous inputs $U_k^+ = \{U_\kappa | \kappa \in \{k+1, ..., k+\tau_p\}\}$, where τ_p is the prediction horizon. The output of the $Prognose$ function is the marking of the prognoser $\hat{M}_{k_{\mathrm{EOP}}|k}$ at the End of Prediction (EOP), where $k_{\mathrm{EOP}} \leq k + \tau_p$.

To keep diagnosis uncertainty, the initial marking of the prognoser $HPPN_\Pi$ is based on the current diagnosis Δ_k given by the marking of the HPPN-diagnoser $HPPN_\Delta$ at time k (line 1). Any diagnosis hypothesis $\{\delta_k, \pi_k^1, ..., \pi_k^n, d_k^1, ..., d_k^n\} \subset \Delta_k$, where the n degradation tokens link the n particles to the configuration δ_k, is reproduced in the prognoser initial marking and takes the form of an equivalent distribution $\{\delta_k^1, \pi_k^1, d_k^1, ..., \delta_k^m, \pi_k^m, d_k^m\}$, in which, for any $i \in \{1, ..., m\}$, d_k^i links δ_k^i and π_k^i, and any δ_k^i has the same value b_k as δ_k.

For prognosis purpose, we redefine the notion of hypothesis on the system future states. A prognosis hypothesis in the HPPN-prognoser $HPPN_\Pi$ at time $\kappa \geq k$ is represented by a set of triplets (a configuration, a particle and a degradation token) $\{\delta_\kappa^1, \pi_\kappa^1, d_\kappa^1, ..., \delta_\kappa^m, \pi_\kappa^m, d_\kappa^m\}$ in which, for any $i \in \{1, ..., m\}$, d_κ^i links δ_κ^i and π_κ^i. The number of triplets $m \in \mathbb{N}_+$ for a prognosis hypothesis is determined with a stochastic scaling algorithm that is a solution to balance precision and computational performance during the algorithm. The stochastic scaling algorithm determines the precisions to give to all the prognosis hypotheses, based on their belief degrees computed by the diagnosis process using particle filtering, and by using three scale parameters ρ_Π^{min}, ρ_Π^{max} and ρ_Π^{tot}. Parameters ρ_Π^{min} and ρ_Π^{max} are respectively the minimum number and the maximum number of tokens of each type to represent a prognosis hypothesis. Parameter ρ_Π^{tot} is the total number of tokens of each type, available to represent all the prognosis hypotheses. Once initialized with Δ_k, the HPPN-based prognoser $HPPN_\Pi$ has as many configurations as particles and degradation tokens: $|M_k^S| = |M_k^N| = |M_k^D|$.

Without performance constraints, a diagnosis hypothesis is obviously completely reproduced, i.e. $m = n$: any particle π_k or degradation token h_k in Δ_k is duplicated. To improve computational performance of the prognosis process, however, the diagnosis hypotheses in Δ_k are partially reproduced.

Once the HPPN-based prognoser $HPNN_\Pi$ is initialized, its marking evolves according to the future inputs U_k^+ (lines 2-8). For each future input pair $U_\kappa = (U_\kappa^S, u_\kappa^N)$ and if the system is not predicted to be in a failure mode (line 3), the marking $\hat{M}_{\kappa-1|k}$ is updated to $\hat{M}_{\kappa|k}$ (line 4) in three steps:

1. *Configuration value update.* If the discrete input set U_κ^S contains an input label u^S that is observable, all the configuration event sets are updated with the event (u^S, κ):
$$\forall \delta \in M^S, b \leftarrow b \cup \{(u^S, \kappa)\}. \quad (9)$$
2. *Transition firing.* All enabled transitions are fired. When a transition t is fired, all the tokens of the prognosis hypotheses in its input places which satisfy conditions Ω_t are moved to the output places of t. Configuration values are updated with the arc annotations $\mathcal{A}_\Pi$.
3. *Particle and degradation token value update.* After the transition firing, a particle value x is updated with the continuous state equation $\mathbf{f}$ associated to the numerical

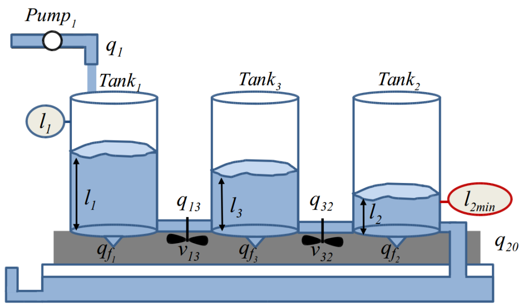

Figure 3. Water tank system.

place the particle belongs, using the continuous input vector u_κ^N. In the same way, a degradation token value d is updated with the degradation state equation $\mathbf{g}$ associated to the degradation place the token belongs, using the continuous input vector u_κ^N and b and x, the values of the configuration and the particle it links.

At the end of the function $Prognose$ (line 9), κ represents the last time where the system may enter a failure mode. It then represents the End Of Pediction (EOP).

The prognosis Π_k takes the form of the marking of the HPPN-prognoser $HPPN_\Pi$ at the end of the process:

$$\Pi_k = \hat{M}_{k_{\mathrm{EOP}}|k}. \tag{10}$$

This marking gives a distribution of beliefs over the system future mode trajectories until time k_{EOP}. The times when the system will possibly enter a failure mode are determined from these future trajectories and are represented as a belief distribution over the system EOL.

5. CASE STUDY

The studied system, illustrated in Figure 3 and described in detail in (Gaudel et al., 2015), is composed of three tanks connected in series. $Pump_1$ delivers a constant water flow q_1 in $Tank_1$. $Tank_2$ empties with an output flow q_{20}. The main function of the system is to maintain a water level l_2 in $Tank_2$ superior to l_{2min}. The 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. This last fault is supposed to occur only after another fault occurrence. Leak occurrence is strongly related to the tank wear or degradation. This degradation is not observable but it is caused by the water pressure. Three qualitative stress levels have been identified: low stress for an operation with a possible water outflow, medium stress for an operation with an impossible water inflow and outflow and high stress for an operation with a possible water inflow but an impossible water outflow. Valve degradation leading to faults f_4 and f_5 is related to the number of valve openings and closings. From this degradation knowledge, it is possible to determine some degradation laws (or aging laws) for the system components that can be represented by probability laws of fault occurrences.The evolution of fault probabilities depends on both the continuous dynamic and the discrete events that occurred on the system. For the leak faults f_1, f_2 and f_3, this evolution depends on the tank stress level. For the valve faults f_4 and f_5, it depends on the number of switches (openings and closings).

In order to simplify the study and the result analysis, only the events associated to the valve v_{13} and the faults f_1, f_4 and f_0 are considered in the following. A multimode description of the health evolution of the water tank system is presented in Figure 4. Ten behavioral modes are identified. For example, in the initial mode Nom_1, both valves are open and let the water transfer from one tank to another (continuous dynamic C_1). In this mode, the stress level for tanks is low (degradation dynamic D_1). When the discrete control event $close_{v_{13}}$ occurs, the system goes into mode Nom_2, where the valve v_{13} is closed (continuous dynamic C_2). In this mode, the stress level for $Tank_2$ and $Tank_3$ is low but the stress level for $Tank_1$ is high (degradation dynamic D_2). From any degraded mode, the system goes into a failure mode when l_2 is inferior to l_{2min} (indicator of fault occurrence f_0).

5.1. HPPN model

The HPPN model of the system is available on the web[1]. The 10 modes have been decomposed into 19 places and 14 transitions represent the 14 mode changes. As explained in section 3, one place can be part of several mode representations. Four noise-free equations of continuous dynamics C_1, C_2, C_3 and C_4 are defined. The noise is the same for each water level and is represented by a gaussian noise with a constant standard deviation of 0.002. The sampling time is 15 sec.

The degradation state is composed of 5 continuous variables which represent the probabilities of fault occurrences. The probability distribution of fault occurrence $f_i, \forall i \in \{1, ..., 5\}$, is a 2-parameter Weibull model. By fixing the shape parameter β_i to 1, the Weibull model is similar to an exponential law by describing random fault process. The scale parameter η_i is representative of the remaining useful life before the fault occurrence f_i. The evolution of the occurrence probability $\mathrm{p}_i = \mathrm{p}_o(f_i)$ of a fault f_i at time k is given by:

$$\mathrm{p}_{i_k} = W_b(k, \eta_i) = \int_0^k \frac{1}{\eta} \mathrm{e}^{\left(-\frac{k}{\eta}\right)} \mathrm{d}k, \tag{11}$$

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

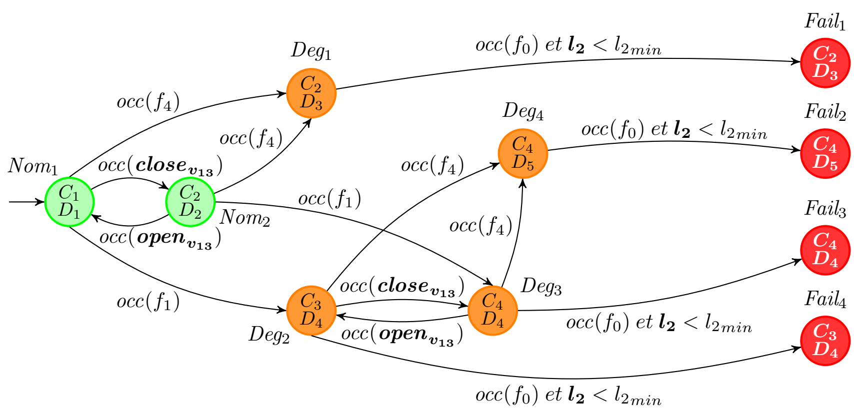

Figure 4. Multimode description of the health evolution of the water tank system.

where W_b is the Weibull cumulative function with $\beta = 1$.

From Equation 11 is derived the generic degradation evolution function for each fault f_i, $i \in \{1, ..., 5\}$:

$$\mathrm{p}_{i_{k+1}} = \mathrm{p}_{i_k} - \mathrm{e}^{\dfrac{-1}{\eta_{i_k}}(-\eta_{i_k} \ln(1 - \mathrm{p}_{i_k}) + k)} + \mathrm{e}^{\dfrac{-1}{\eta_{i_k}} - \eta_{i_k} \ln(1 - \mathrm{p}_{i_k})}. \quad (12)$$

The fault occurrence probabilities of valves p_4 and p_5 increase rapidly with the number of openings and closings. At each time k, the scale parameter η_4 of the Weibull model which represents the degradation of the valve v_{13} (resp. η_5 for the valve v_{32}) depends on the coefficient α_{13} (resp. α_{32}) that stands for the number of event occurrences $open_{v_{13}}$ and $close_{v_{13}}$ (resp. $open_{v_{32}}$ and $close_{v_{32}}$) until time k:

$$\eta_{4_k} = \frac{\eta^0_{13}}{\alpha_{13_k}}, \quad \eta_{5_k} = \frac{\eta^0_{32}}{\alpha_{32_k}}, \quad (13)$$

where η^0_{13} (resp. η^0_{32}) is representative of the remaining life before the valve v_{13} (resp. v_{32}) gets stuck in the closed position. By considering $\eta^0_{13} = \eta^0_{32} = \eta^0$ and $\eta^0 = 3.10^7$, this remaining useful life is about four years.

The parameters η_1, η_2 and η_3 of leak occurrence probabilities p_1, p_2 and p_3, are constant in time but different depending on the stress level of tanks in each mode. Three values of the scale parameter η have been determined to represent the stress levels of tanks: $\eta_a = 10^5$ for low stress, $\eta_b = 8.10^4$ for medium stress and $\eta_c = 6.10^4$ for high stress. For example, in mode Nom_1, the parameters η_1, η_2 and η_3 of leaks are instantiated with η_a for low stress. The predefined threshold selected by the user for deciding that a fault occurred knowing its degradation modeled by a fault occurrence probability is 0.9.

5.2. HPPN-based prognoser

The HPPN-based prognoser, available on the web[2], is generated from the HPPN model. It contains the same number of places and transitions as the HPPN model. All mode changes are predictable and no transition is removed during the prognoser generation.

The scale parameters of the HPPN-based prognoser are $\rho^{min}_{\Pi} = 1, \rho^{max}_{\Pi} = 3, \rho^{tot}_{\Pi} = 50$ so that the evolution of 16 to 50 prognosis hypotheses will be simulated with a precision m between 1 and 3.

Three scenarios are tested. The initial system mode is always Nom_1. The valves v_{13} and v_{32} are open and have never been used ($b_0 = \emptyset$), the continuous state $x_0 = [l_{10}, l_{20}, l_{30}] = [0.60, 0.55, 0.58]$, the degradation state $d_0 = [\mathrm{p}_{10}, \mathrm{p}_{20}, \mathrm{p}_{30}, \mathrm{p}_{40}, \mathrm{p}_{50}] = [0, 0, 0.001, 0, 0]$. The water flow delivered in $Tank_1$ 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 occurs at 201840s and f_0 occurs at 206040s. The HPPN-prognoser computes a prognostic result every 30 minutes with a prediction horizon $\tau_p = 6.10^5$s.

In Scenario 1, the future continuous and discrete inputs are supposed to be accurately known: q_1 is constant and v_{13} is closed every hour during 20 minutes after 310 minutes of operation. The EOL estimations computed by the prognoser are expected to be close to the EOL of the simulated system that is 206040s. Figure 5 shows the RUL estimations of the system. The RUL of the prognosis hypotheses are drawned in different shades of grey: those with higher belief degrees are black, while those with low belief degrees are in light gray. The EOL of the simulated system is represented by a discontinuous line. The EOL with the higher belief degrees is about 10800 seconds before the EOL of the simulated system. These results are the expected ones. The diagnoser correctly

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

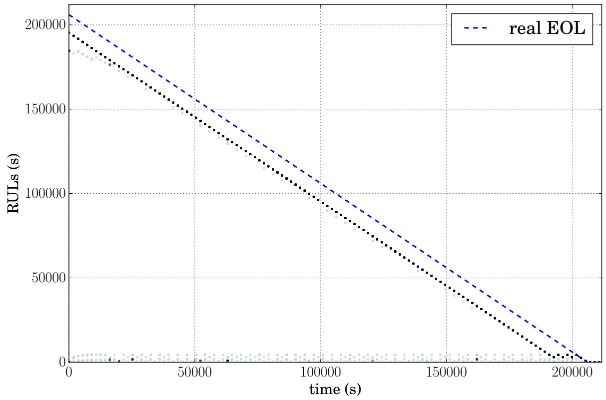

Figure 5. Scenario 1: RUL estimations.

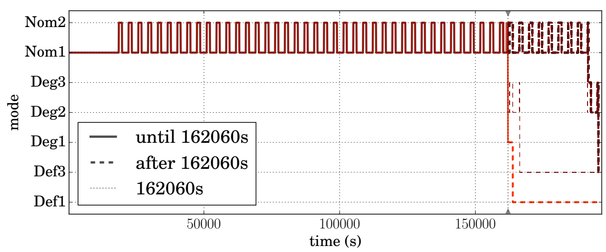

Figure 6. Scenario 1: Belief distribution on past and future trajectories of the system.

estimates the system mode and the prognoser uses this information to compute the RUL with a fault probability threshold fixed at 0.9. It explains the constant gap between the predicted RUL of hypotheses with the higher belief degrees and the real RUL. Because of the ambiguity in the diagnosis result, the prognoser estimates a shorter RUL between 460s and 4300s for some hypotheses with low belief degrees that consider the system is in a degraded mode. Figure 6 shows the belief distribution on the past and future trajectories of the system at 162060s by combining diagnosis and prognosis hypotheses. This result can be more informative for health management purpose.

In Scenario 2, the future inputs are now unknown. The prognoser takes as input current flow q_1 but it cannot simulate the system mode changes associated to the occurrence of discrete events. Then it cannot simulate that the system enters mode Nom_2 with a high stress level. The EOL estimations are then overestimated (Figure 7). Until the valve v_{13} is closed, the EOL of the hypothesis with the higher belief degree is approximately 235000s, that is to says 28960 seconds after the EOL of the simulated system. After 18600s, the EOL with the higher belief degree is sooner because the diagnoser estimates the system stays more longer in the mode Nom_2 where it is more stressed than in the mode Nom_1. These results were

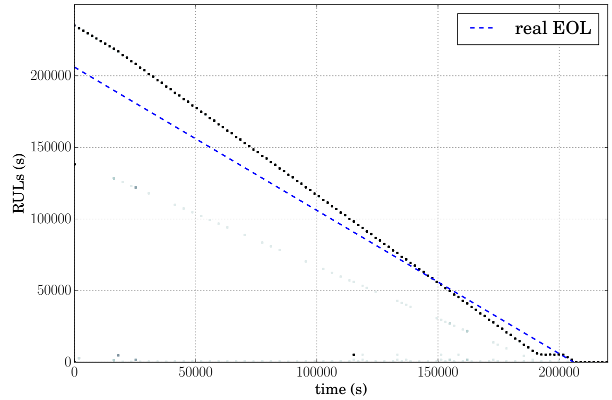

Figure 7. Scenario 2: RUL estimations.

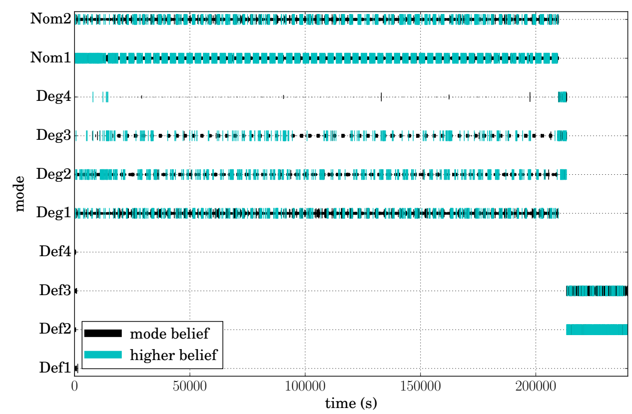

Figure 8. Scenario 3: Belief distribution on the system mode.

expected, they illustrate that the diagnoser correctly estimates the degradation state and the prognoser uses this estimation to propagate this state in the future. Figure 7 shows the convergence of the predicted RUL towards the real RUL.

In scenario 3, the future inputs are supposed to be known. The objective is to illustrate the impact of diagnostic results on prognostic results. The fault f_1 is injected at 209700s when the system enters in the mode Nom_2 and the fault f_0 occurs at 213360s. The EOL of the simulated system is 213360s. Figure 8 shows the belief distributions on the system mode. Continuous observations were simulated with four times greater noise than the specifications. The diagnoser cannot identify the continuous dynamics from observations, so no mode predominates the others in the results.

Figure 9 shows the RUL estimations of the system which are similar to results obtained for Scenario 1 (see Figure 5). However some hypotheses that consider a shorter RUL have a more higher belief degree than in Scenario 1. These results

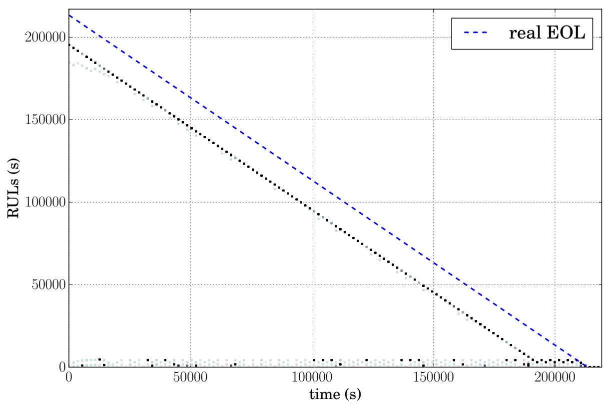

Figure 9. Scenario 3: RUL estimations.

Table 2. Computational performance of the HPPN-based health monitoring method applied on the water tank system.

Scale parameters		Execution time Δ_k (s)	Execution time Π_k (s)	RAM max. (MB)
$(40, 80, 1500)_\Delta$ $(1, 5, 100)_\Pi$	min.	0.12	0.08	
	max.	3.96	201.73	
	av.	1.22	44.29	102.56

were predictible because the diagnostic results are more uncertain than for Scenario 1.

The three scenarios illustrate how the prognoser uses information contained in the current diagnosis and the knowledge on future inputs of the system and how the prognoser propagates the related uncertainty in the future. Table 2 exposes computational performance of diagnosis and prognosis processes for the chosen scale parameters . The complexity of a HPPM model is difficult to evaluate, the number of places and transitions, the computational complexity of continuous and degradation dynamics, the number of state variables are parameters involved in general performance of the monitoring method. Computational performance can still be controlled by choosing suited scale parameters. By reducing scale parameters, the execution times and maximum RAM are reduced but it also affect the monitoring efficiency because the number of diagnosis and prognosis hypotheses is reduced.

6. CONCLUSION

This work proposes an approach of prognosis of hybrid systems based on Hybrid Particle Petri Nets, and its application on a three-tank case study.

Future works will focus on the HPPN model, diagnoser and prognoser verifications. We also aim at investigating how the HPPN-based prognosis results may influence the HPPN-based diagnoser monitoring (and vice versa). The prognoser could, for example, increase the precision for the monitoring of a diagnosis hypothesis on the system, whose predicted future trajectory is particularly critical. The main application perspective is the application of the methodology on a real rover. The stochastic scaling algorithm will provide a compromise between performance and available computational resources, through the setting of scale parameters and so manage the complexity of real complex systems.

REFERENCES

Basile, F., Chiacchio, P., & Tommasi, G. D. (2009). Fault diagnosis and prognosis in Petri Nets by using a single generalized marking estimation. In 7^{th} *IFAC SAFE-PROCESS.*

Bayoudh, M., Travé-Massuyes, L., & Olive, X. (2008). Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *IFAC World Congress* (pp. 7265–7270).

Chanthery, E., & Ribot, P. (2013). An Integrated Framework for Diagnosis and Prognosis of Hybrid Systems. In 3^{rd} *Workshop on Hybrid Autonomous System.* Italy.

Daigle, M., Roychoudhury, I., & Bregon, A. (2015). Model-based Prognostics of Hybrid Systems. In *Annual Conf. of the PHM Society.* USA.

Engel, S., Gilmartin, B., Bongort, K., & Hess, A. (2000). Prognostics, the real issues involved with predicting life remaining. In *Aerospace Conf., IEEE* (Vol. 6, p. 457-469).

Gaudel, Q., Chanthery, E., & Ribot, P. (2014). Health Monitoring of Hybrid Systems Using Hybrid Particle Petri Nets. In *Annual Conf. of the PHM Society.* USA.

Gaudel, Q., Chanthery, E., & Ribot, P. (2015). Hybrid Particle Petri Nets for Systems Health Monitoring under Uncertainty. *Int. Journal of PHM*, *6*(022).

Lefebvre, D. (2014). Fault Diagnosis and Prognosis With Partially Observed Petri Nets. *IEEE Trans. on Systems, Man, and Cybernetics*, *44*(10), 1413–1424.

Ribot, P., Pencolé, Y., & Combacau, M. (2013). Generic characterization of diagnosis and prognosis for complex heterogeneous systems. *Int. Journal of Prognostics and Health Management*, *4*(023).

Vianna, W. O. L., & Yoneyama, T. (2015). Interactive Multiple-Model Application for Hydraulic Servovalve Health Monitoring. In *Annual Conf. of the PHM Society.* USA.

Yu, M., Wang, D., & Luo, M. (2014). Model-Based Prognosis for Hybrid Systems With Mode-Dependent Degradation Behaviors. *IEEE Trans. on Industrial Electronics*, *61*(1), 546–554.

Zabi, S., Ribot, P., & Chanthery, E. (2013). Health Monitoring and Prognosis of Hybrid Systems. In *Annual Conf. of the PHM Society.*