



Hybrid Particle Petri Nets for Systems Health Monitoring under Uncertainty

Quentin Gaudel, Elodie Chanthery, Pauline Ribot

► To cite this version:

Quentin Gaudel, Elodie Chanthery, Pauline Ribot. Hybrid Particle Petri Nets for Systems Health Monitoring under Uncertainty. International Journal of Prognostics and Health Management, 2015, Special Issue Uncertainty in PHM, 6. hal-01229083

HAL Id: hal-01229083

<https://hal.science/hal-01229083>

Submitted on 16 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hybrid Particle Petri Nets for Systems Health Monitoring under Uncertainty

Quentin Gaudel¹, Elodie Chanthery², and Pauline Ribot³

^{1,2,3} CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
quentin.gaudel@laas.fr, elodie.chanthery@laas.fr, pauline.ribot@laas.fr

^{1,2} Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

³ Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France

ABSTRACT

This paper focuses on how to treat uncertainty in health monitoring of hybrid systems by using a model-based method. The Hybrid Particle Petri Nets (HPPN) formalism is defined in the context of health monitoring to model hybrid systems and to generate diagnosers of such systems. The main advantage of this formalism is that it takes into account knowledge-based uncertainty and uncertainty in diagnosis process. The HPPN-based diagnoser deals with occurrences of unobservable discrete events (such as faults) and is robust to false observations. It also estimates the continuous state of the system by using particle filtering. Finally, HPPN can represent the system degradation that is often dealt with using probabilistic tools. A hybrid technique is thus used to group all this knowledge and to deduce the diagnosis results. The approach is demonstrated on a three-tank system. Experimental results are given, illustrating how different kinds of uncertainty are taken into account when using HPPN.

1. INTRODUCTION

Real systems have become so complex that it is often impossible for humans to capture and explain their behaviors as a whole, especially when they are exposed to failures. It is therefore necessary to develop tools that can support operator tasks but that also reduce the global costs due to unavailability and repair actions. An efficient health monitoring technique has to be adopted to detect, isolate (diagnosis) and predict faults (prognosis) leading to failures.

Recent industrial systems exhibit an increasing complexity of dynamics that are both continuous and discrete. It has become difficult to ignore the fact that most systems are hy-

brid (Henzinger, 1996). In (Chanthery & Ribot, 2013), we extended the diagnosis approach proposed in (Bayoudd, Travé-Massuyes, & Olive, 2008) in order to integrate diagnosis and prognosis for hybrid systems. The approach uses hybrid automata and stochastic models for the system degradation. Diagnosis is performed using a Discrete Event System (DES) approach.

The main drawback of this approach is that the DES-oriented diagnosis framework explodes in number of states and it does not seem to be the most suited for the incorporation of the prognosis task. Prognosis is indeed a probabilistic prediction process and is highly subject to uncertainty. The health monitoring task usually has to take into account the different sources of uncertainty, such as model approximation, partial observability of the system and measurement noise. Diagnosis should help the decision making process. In case of ambiguity in diagnosis results, the traditional diagnoser fails at providing relevant information. By taking all uncertainty sources into account, the method we propose succeeds in quantifying each diagnosis result.

Our previous works introduced a two-level framework called *Hybrid Particle Petri Nets* (HPPN). (Gaudel, Chanthery, Ribot, & Le Corronc, 2014) proposed to use the *behavioral level* of the HPPN (HPPN-BL) to both specify the system behavior, which is hybrid but also uncertain, and track the system current health state with a diagnoser representation. (Gaudel, Chanthery, & Ribot, 2014) extended this work by introducing the *hybrid level* of the HPPN (HPPN-HL). The purpose of the hybrid level is to represent system hybrid characteristics, such as degradation, which can depend on both the discrete and continuous parts of the system. Getting some information about the degradation of the system is a significant advantage for elaborating a more precise diagnosis and to perform prognosis. This paper recalls these two contributions and illustrates all the concepts step by step with a running example

Quentin Gaudel et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

consisting in a three-tank system. The main contribution of the paper is to expose simulation results of the implemented HPPN-based health monitoring method. Experimental results are given, illustrating how different kinds of uncertainty are taken into account when using HPPN.

The paper is organized as follows. Section 2 presents related work on health monitoring based on Petri net formalism in the case of hybrid systems under uncertainty. Section 3 recalls the HPPN-BL formalism. Section 4 presents how HPPN-BL is applied to health monitoring and explains the generation of a behavioral diagnoser using the HPPN formalism. Section 5 presents the HPPN-HL formalism. Section 6 explains how HPPN-HL is applied to health monitoring and enrich the diagnosis results. Section 7 provides the results obtained by testing the HPPN-based health monitoring method on a three-tank system, focusing on the uncertainty management. Some conclusions and future work are discussed in the final section.

2. RELATED WORK

The diagnoser approach was introduced by (Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995). The diagnoser is basically a monitor that is able to process any possible observable event on the system. It consists in recording these observations and providing the set of possible faults whose occurrence is consistent with the observations. However, this approach is restricted to DES and does not manage uncertainty.

Some approaches, such as (Boubour, Jard, Aghasaryan, Fabre, & Benveniste, 1997)(Genc & Lafortune, 2007)(Soldani, Combacau, Subias, & Thomas, 2007)(Cabasino, Giua, & Seatzu, 2014), extend the diagnoser to DES modelled by Petri nets. (Genc & Lafortune, 2007) proposes a distributed version of the diagnoser in “Petri net diagnosers”. In (Cabasino et al., 2014), the authors study the diagnosability of a system, inspired by the diagnosability approach for finite state automata proposed by (Sampath et al., 1995). They used the Modified Basis Reachability Graph in order to build the reachability diagnoser that is represented as a graph. However, none of these approaches take into account continuous aspects, nor do they consider uncertainties in the system. (Soldani et al., 2007) proposes an approach for the localization of intermittent faults. This article proposes some clues for dealing with partial observability in the discrete event framework. It proposes a method based on Petri net modeling the normal functioning of the system observable behavior. The detection step consists in comparing the flow of observable events emitted by the monitored system to the flow predicted by the model. A localization mechanism, based on the diagnoser approach, points out the set of events potentially responsible for the faults.

Some works (Koutsoukos, Kurien, & Zhao, 2002), (Ru & Hadjicostis, 2009), (Basile, Chiacchio, & Tommasi, 2009), (Jianxiong et al., 2013) try to take into account uncertainty.

(Koutsoukos et al., 2002) uses a particle filtering technique to estimate the state of a hybrid system modeled as a hybrid automaton. Uncertainty related to discrete events is not taken into account and the system degradation is not considered. (Ru & Hadjicostis, 2009) uses partially observed Petri nets. Partially observed Petri nets are transformed into an equivalent labelled Petri net and an online monitor is built to diagnose faults and provide beliefs (degrees of confidence) regarding the occurrences of faults. However, this approach is limited because it only takes into account the uncertainties in the diagnosis results, not about the model or the event observations. (Basile et al., 2009) proposes to reduce the explosion of the state space by introducing generalized markings (negative tokens) to take into account uncertainty about the firing of transitions. (Jianxiong et al., 2013) uses the stochastic Petri nets to build a formal model of each component of integrated modular avionics architecture. However, for all these approaches, no continuous aspect in the model is taken into account.

To get a more compact representation and to capture all uncertainties related to the system, the observations and the diagnosis results, we propose to consider the Modified Particle Petri Nets (MPPN) formalism defined in (Zouaghi, Alexopoulos, Wagner, & Badreddin, 2011a).

MPPN are an extension of particle Petri nets (Lesire & Tessier, 2005) that combine a discrete event model (Petri net) with a continuous model (differential equations). The main advantage of MPPN is that uncertainties and hybrid dynamics are taken into account. A particle filter is used to integrate probabilities in the continuous state estimation process. In (Zouaghi et al., 2011a), the application is based on the mission monitoring. It does not consider different health states for the system. There is no matching with the diagnoser approach and the problem of ambiguous state tracking is not considered. In (Zouaghi, Alexopoulos, Wagner, & Badreddin, 2011b), the authors propose a design approach for the specification and the realization of dynamic system monitoring. They present the integration of system design and monitoring into a unified framework for the reuse of component descriptions and the automatic monitoring component generation. They automatically convert MPPN into an XML description and the monitoring parameters are added to the file. However, this observer is not formally defined, and no equivalence with a diagnoser is given. In addition, there is no mention of the health state notion for the system. This paper proposes to use HPPN for health monitoring. The HPPN-BL is based on the MPPN formalism and takes into account the hybrid dynamics and the inherent uncertainty of real systems. Then the work formally specifies a diagnoser object for computing online diagnosis. This paper also introduces the HPPN-HL that captures the system hybrid characteristics.

3. BEHAVIORAL LEVEL OF HYBRID PARTICLE PETRI NETS

This section recalls the behavioral level of the Hybrid Particle Petri Nets formalism (HPPN-BL). This formalism is directly inspired from (Zouaghi et al., 2011a) and adapted to fit diagnosis requirements. First the model structure is detailed, then its online use is presented.

Throughout the paper, a running example of the three-tank system, described in Figure 1, is used to illustrate the definitions and concepts, and the proposed extensions to diagnosis.

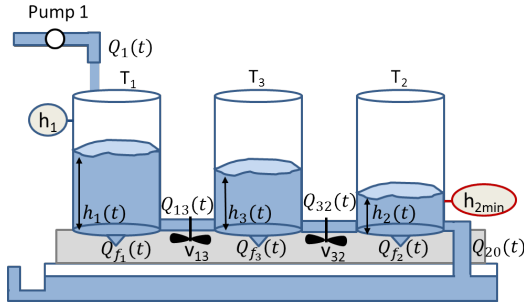


Figure 1. Three-tank system description.

The tanks are configured in a series circuit. The water comes from Pump 1 into tank T_1 . Flow $Q_1(t)$ delivered by the pump is supposed to be constant. The tank T_2 empties with flow $Q_{20}(t)$. The available measurements at time t are the water level in tank T_1 denoted $h_1(t)$ and the total mass of water in the three-tank system denoted $W(t)$. Valves v_{13} and v_{32} allow the flow to transfer between tanks. They are controlled by discrete control inputs $open_{v_{13}}$, $close_{v_{13}}$, $open_{v_{32}}$ and $close_{v_{32}}$. Fault events f_1 , f_2 and f_3 model leaks that may occur in tanks T_1 , T_2 and T_3 . The leak f_1 is three times larger than f_2 and f_3 . f_4 and f_5 are fault events that may occur on the valves. The valve v_{13} (respectively v_{32}) may become stuck closed f_4 (respectively f_5). The goal of the system is to maintain the water level in tank T_2 greater than a minimum value h_{2min} . The leak in tank T_1 is considered too large and therefore leads to system failure when f_1 occurs because the system is not able to achieve the goal anymore. Leaks in tanks T_2 and T_3 are separately not large enough to prevent the goal achievement, but the occurrences of the two leaks lead to system failure. Both f_4 and f_5 prevent the water delivered by the pump to reach the tank T_2 and then lead to system failures. It is then supposed that the system is in a failure mode either if f_1 or f_4 or f_5 occurs, or if f_2 and f_3 occur.

3.1. Definition

The HPPN formalism is an extension of Petri nets. A Petri net is a directed bipartite graph, in which a node can be either

a *place* or a *transition*. The directed *arcs* cannot link two places or two transitions. Places may contain *tokens*. The distribution of tokens over the places is called the *marking* of the net. A transition may be *fired* if it is *enabled* (ie. there is enough tokens in its input places). The firing of a transition consumes the required tokens in its input places and creates tokens in its output places.

The HPPN-BL is defined as a tuple $\langle P, T, Pre, Post, X, C, \gamma, \Omega, M_0 \rangle$ where:

- P is the set of places, partitioned into numerical places P^N and symbolic places P^S .
- T is the set of transitions (numerical T^N , symbolic T^S and mixed T^M).
- Pre and $Post$ are the incidence matrices of the net, of dimension $|P| \times |T|$.
- $X \subset \mathbb{R}^n$ is the state space of the numerical state vector.
- C is the set of dynamic equation sets of the system associated with numerical places, representing continuous state evolution.
- $\gamma(p^S)$ is the application that associates tokens with each symbolic place $p^S \in P^S$.
- Ω is the set of conditions (numerical Ω^N and symbolic Ω^S) associated with transitions.
- M_0 is the initial marking of the net.

The marking of the net is composed of tokens, that can be numerical tokens (particles) or symbolic tokens (configurations).

A numerical place $p^N \in P^N$ is associated with a set of dynamic equations representing the continuous behavior of the system. Numerical places thus model system continuous dynamics. Numerical places are marked by a set of particles $\pi_k^i = [x_k^i, w_k^i]$ with $i \in \{1, \dots, |M_k^N|\}$ where M_k^N is the set of all the particles in the net at time k . Particles are defined by their corresponding numerical state vector $x_k^i \in X$ and their weight $w_k^i \in [0, 1]$ at time k . The set of particles represents a distribution over the value of the numerical state vector.

Symbolic places model the discrete states of the system. A symbolic place $p^S \in P^S$ is marked by configurations δ_k^j with $j \in \{1, \dots, |M_k^S|\}$ where M_k^S is the set of configurations in the net at time k . The set of configurations represents all the possible current modes of the system.

The marking M_k of the HPPN-BL at time k consists of both kinds of tokens:

$$M_k = \{M_k^S, M_k^N\}. \quad (1)$$

Example 1 For the three-tank system, there exist 4 nominal discrete states, denoted $nom_1, nom_2, nom_3, nom_4$, representing the conditions on the valves defined in Table 1. A

fault event f_j , $1 \leq j \leq 5$ may occur in any nominal mode and leads to anticipated fault mode $1f_j, 2f_j, 3f_j, 4f_j$, respectively, so that 16 one-fault modes exist. For example, if f_2 occurs in nominal mode nom_3 , the system evolves in mode $3f_2$. Multiple faults are considered and faults are assumed to be independent, that is to say that $x_{f_i f_j}$ is the same mode as $x_{f_j f_i}$. For the three-tank system, 20 two-fault modes are determined. Each discrete state is represented by a symbolic place. For example the set $P_{nom}^S = \{nom_1, nom_2, nom_3, nom_4\}$ represents the nominal discrete states of the system. Each nominal discrete state is associated with a set of dynamic equations representing the different continuous behaviors of the system. For some cases, two or more discrete states have the same dynamic equations, so it is not necessary to define different continuous behaviors for all these discrete states. Each identified continuous dynamic is represented by a numerical place, for example the set of nominal behaviors $P_{nom}^N = \{D_1, D_2, D_3, D_4\}$ where D_1, D_2, D_3, D_4 are the system dynamics defined in Table 1, with S_1, S_2, S_3 the tank sections and K_{ij} real coefficients. All continuous dynamics of the three-tank system are provided in the appendix.

3.2. Firing Rules

A transition models a change in the continuous dynamics and/or a change of system mode. A symbolic transition is conditioned by an observable discrete event. A numerical transition is conditioned by a set of constraints on continuous observable variables. Finally, a mixed transition is conditioned by an observable discrete event and a set of constraints on continuous observable variables.

Let $Pre(t_j)$ be the set of input places of a transition $t_j \in T$:

$$Pre(t_j) = \{p_i | Pre(i, j) \neq 0, i \in \{1, \dots, |P|\}\}. \quad (2)$$

$Post(t_j)$ is the set of its output places:

$$Post(t_j) = \{p_i | Post(i, j) \neq 0, i \in \{1, \dots, |P|\}\}. \quad (3)$$

$\forall p_i \in P$, $M_k(p_i)$ is the set of tokens in p_i at time k and $m_k(p_i) = |M_k(p_i)|$ is the number of tokens in p_i at time k .

Definition 1 A numerical or a symbolic transition t_j is fire-enabled at time k if:

$$\forall p_i \in Pre(t_j), m_k(p_i) \geq Pre(i, j). \quad (4)$$

A numerical transition $t_j^N \in T^N$ is associated with numerical conditions $\Omega^N(t_j^N)$, where $\Omega^N(t_j^N)(\pi) = 1$ if the particle satisfies the conditions. For example, if $\pi = [x, w]$ follows the constraint equation c and b is a trigger value, numerical conditions can be defined as $\Omega^N(t_j^N)(\pi) = (c(x) > b)$. $\Omega^S(t_j^S) = (occ(e))$ are the symbolic conditions assigned to a symbolic transition $t_j^S \in T^S$. $occ(e)$ is a boolean indicator of the occurrence of the discrete event e : $occ(e) = 1$ if e occurs. Then, a configuration δ satisfies symbolic conditions $\Omega^S(t_j^S)$

when $\Omega^S(t_j^S)(\delta) = 1$, i.e. when the event e occurs.

Firing rules of a transition $t_j \in T$ use the following set of variables over their domains of definition. It is assumed that: $p^N \in (Pre(t_j) \cap P^N)$, $p'^N \in (Post(t_j) \cap P^N)$, $p^S \in (Pre(t_j) \cap P^S)$, $p'^S \in (Post(t_j) \cap P^S)$.

If $t_j \in (T^N \cup T^M)$, let $\mathcal{S}_k^N(p^N)$ be the set of particles in p^N that satisfy the conditions $\Omega^N(t_j)$ at time k : $\mathcal{S}_k^N(p^N) \subseteq M_k^N(p^N)$ with $\pi \in \mathcal{S}_k^N(p^N)$ if $\Omega^N(t_j)(\pi) = 1$ at time k .

The numerical firing uses the concept of classical firing with the particles satisfying the numerical conditions and the concept of pseudo-firing (ie. duplication) for the configurations. The duplication of configurations represents uncertainty about the occurrence of an unobservable discrete event.

Definition 2 The firing of a fire-enabled numerical transition $t_j^N \in T^N$ at time k is defined by:

$$\begin{cases} M_{k+1}^N(p^N) = M_k^N(p^N) \setminus \mathcal{S}_k^N(p^N) \\ M_{k+1}^N(p'^N) = M_k^N(p'^N) \cup \mathcal{S}_k^N(p^N), \end{cases} \quad (5)$$

$$\begin{cases} M_{k+1}^S(p^S) = M_k^S(p^S) \\ M_{k+1}^S(p'^S) = M_k^S(p'^S) \cup M_k^S(p^S). \end{cases} \quad (6)$$

An example of a numerical firing at time k is illustrated in Figure 2(a) adapted from (Zouaghi et al., 2011a).

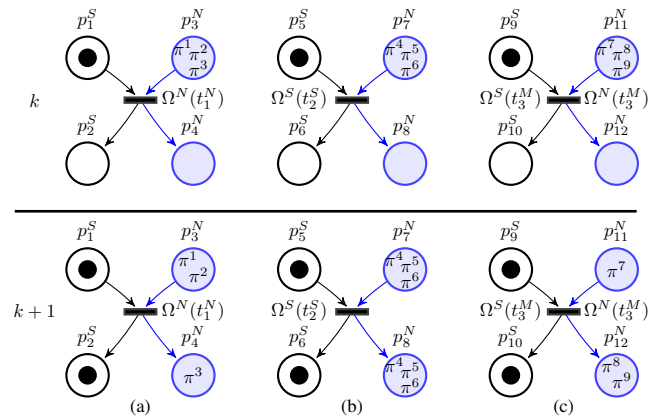


Figure 2. Illustration of firing rules of numerical (a), symbolic (b) and mixed (c) fire-enabled transitions.

In this example, t_1^N has only numerical conditions because it is a numerical transition. Particle π^3 satisfies the numerical conditions $\Omega^N(t_1^N)$ and thus is moved to p_4^N . The configuration in place p_1^S is duplicated in p_2^S .

The symbolic firing uses the concept of pseudo-firing for particles and configurations. The pseudo-firing of all the tokens models uncertainty about the non-occurrence of a discrete

Table 1. Three-tank system example: state equations for nominal discrete states.

	Valve state	State equations
nom_1	$v_{13} = open$ $v_{32} = open$	$D_1 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{ h_1(t) - h_3(t) } - K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$
nom_2	$v_{13} = closed$ $v_{32} = open$	$D_2 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{-K_{32}}{S_3}\sqrt{ h_3(t) - h_2(t) } \end{cases}$
nom_3	$v_{13} = closed$ $v_{32} = closed$	$D_3 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{-Q_{20}(t)}{S_2} \\ \dot{h}_3(t) = \frac{K_{13}}{S_3}\sqrt{ h_1(t) - h_3(t) } \end{cases}$
nom_4	$v_{13} = open$ $v_{32} = closed$	$D_4 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{Q_{20}(t)}{S_2} \\ \dot{h}_3(t) = \frac{K_{13}}{S_3}\sqrt{ h_1(t) - h_3(t) } \end{cases}$

event.

Definition 3 The firing of a fire-enabled symbolic transition $t_j^S \in T^S$ at time k is defined by:

$$\begin{cases} M_{k+1}^N(p^N) = M_k^N(p^N) \\ M_{k+1}^N(p'^N) = M_k^N(p'^N) \cup M_k^N(p^N), \end{cases} \quad (7)$$

$$\begin{cases} M_{k+1}^S(p^S) = M_k^S(p^S) \\ M_{k+1}^S(p'^S) = M_k^S(p'^S) \cup M_k^S(p^S). \end{cases} \quad (8)$$

Figure 2(b) illustrates an example of a symbolic firing. The symbolic transition t_2^S has only symbolic conditions and is fire-enabled. No configuration satisfies the symbolic conditions $\Omega^S(t_2^S)$, however, all the tokens are duplicated.

Mixed transitions (also called hybrid transitions in (Zouaghi et al., 2011a)) model the interactions between discrete events and system continuous dynamics. A mixed transition merges a symbolic transition with a numerical transition to consider the correlation between discrete observations and continuous observations. A mixed transition $t_j^M \in T^M$ is then associated with both numerical conditions $\Omega^N(t_j^M)$ and symbolic conditions $\Omega^S(t_j^M)$.

Definition 4 A mixed transition $t_j^M \in T^M$ is fire-enabled at time k if $\forall p_i \in Pre(t_j^M)$:

$$\begin{cases} m_k(p_i^S) \geq Pre(i, j), & \text{if } p_i^S \in Pre(t_j^M) \cap P^S \\ m_k(p_i^N) \geq 0, & \text{if } p_i^N \in Pre(t_j^M) \cap P^N. \end{cases} \quad (9)$$

The mixed firing uses the concept of classical firing with the particles satisfying the numerical conditions and the concept of pseudo-firing with the configurations. The pseudo-firing of configurations models uncertainty about the occurrence of

an observable discrete event that is supported by a change of continuous dynamics.

Definition 5 The firing of a mixed transition $t_j^M \in T^M$ at time k is defined by:

$$\begin{cases} M_{k+1}^N(p^N) = M_k^N(p^N) \setminus \mathcal{S}_k^N(p^N) \\ M_{k+1}^N(p'^N) = M_k^N(p'^N) \cup \mathcal{S}_k^N(p^N), \end{cases} \quad (10)$$

$$\begin{cases} M_{k+1}^S(p^S) = M_k^S(p^S) \\ M_{k+1}^S(p'^S) = M_k^S(p'^S) \cup M_k^S(p^S). \end{cases} \quad (11)$$

An example of a mixed firing is illustrated in Figure 2(c). t_3^M is a mixed transition therefore it has symbolic conditions and numerical conditions. The configuration in place p_9^S is duplicated. Regarding the numerical part, particles π^8 and π^9 satisfy $\Omega^N(t_3^M)$ so they are moved to p_{12}^N . Furthermore, π^7 stays in place p_{11}^N because it does not satisfy $\Omega^N(t_3^M)$.

3.3. State Estimation

HPPN-BL hybrid estimation is based on the marking evolution and on the particle filtering: a prediction step and a correction step are performed on the tokens.

For the sake of clarity, in this paper, it is assumed that a hybrid state is represented by a couple (p_i^S, p_j^N) . The initial marking of the HPPN-BL is $M_0 = \{M_0^S, M_0^N\}$ and the estimated marking at time k is $\hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N\}$ where $\hat{M}_k = \hat{M}_{k|k}$. The observations start at time $k = 1$, $O_1 = (O_1^S, O_1^N)$ where O^S and O^N respectively represent the observations corresponding to the symbolic part and the numerical part.

- (1) The prediction step is based on the evolution of the marking of the HPPN-BL and on the estimation of the parti-

cle values. It aims at determining all possible next states of the system $\hat{M}_{k+1|k} = \{\hat{M}_{k+1|k}^S, \hat{M}_{k+1|k}^N\}$. Noise is added during the particle values update to take into account uncertainty about the dynamic equations and thus about the continuous system model.

- (2) The correction step is based on the update of the prediction according to new observations on the system.
 - (a) A numerical correction, based on the particle filter algorithm, produces a probability distribution Pr_{DN} of the particles $\hat{M}_{k+1|k+1}^N$ over the numerical places. At this step, particle weights are updated using a probability distribution function depending on random noise that models uncertainty about continuous observations O_{k+1}^N .
 - (b) A symbolic correction then computes a probability distribution Pr_{DS} over the symbolic states of the system, depending on discrete observations O_{k+1}^S and on Pr_{DN} making the process hybrid.

Finally, in order to update the complete predicted marking $\hat{M}_{k+1|k}$, a decision making method is required. The result of the whole state estimation process is the estimated marking at time $k+1$, $\hat{M}_{k+1|k+1} = \{\hat{M}_{k+1|k+1}^S, \hat{M}_{k+1|k+1}^N\}$.

The main advantage provided by HPPN is the way they manage uncertainty. This article focuses on a way to use them in a context of health monitoring.

4. APPLICATION TO BEHAVIORAL MONITORING

Diagnosis is used to identify the possible causes of failures by reasoning on system observations. Diagnosis reasoning consists in detecting and isolating faults that may cause a system failure. Results of the diagnosis function lead to the current health state of the system. We are interested in representing changes in the system dynamics when one or several anticipated faults occur. We defined a *health mode* as a discrete health state coupled to a continuous dynamics (Gaudel, Chanthery, Ribot, & Le Corronc, 2014). As long as the system does not encounter any fault, it is in a *nominal mode*. Tracked faults are assumed to be permanent, i.e. once a fault happens, the system moves from a nominal mode to a *degraded mode* or faulty mode. Without repair, the system evolution is unidirectional as illustrated in Figure 3 and the system ends in a *failure mode* in which it is not operational anymore.

Example 2 The evolution of the three-tank system is illustrated in Figure 4. There are 8 degraded modes defined by the occurrence of fault events f_2 or f_3 and 28 failure modes defined either by the occurrence of f_1 or f_4 or f_5 or by the occurrence of f_2 and f_3 . 16 one-fault modes and the 20 two-fault modes are identified.

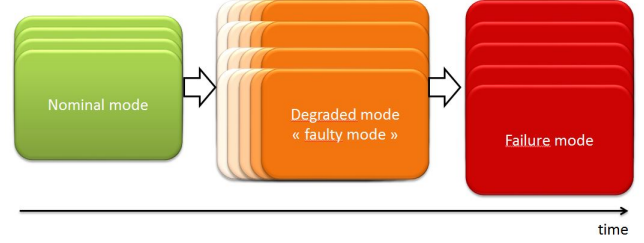


Figure 3. Unidirectional system evolution without maintenance or repair action.

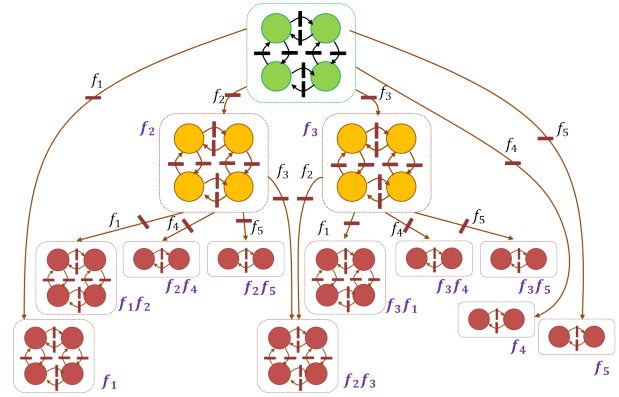


Figure 4. Unidirectional evolution of the three-tank system.

4.1. Behavior Modeling

With the definition of the HPPN-BL, it is possible to model a hybrid system behavior in a health monitoring context. Numerical places can represent the system continuous dynamics, and symbolic places can represent the different discrete health states of the system. Thus, a hybrid state (p_i^S, p_j^N) will represent a health mode of the system. Let $Q = \{q_m\}$ denote the set of health modes of the system:

$$q_m = (p_i^S, p_j^N) \in Q \text{ if } \exists t_l \in T, (p_i^S, p_j^N) \in (\text{Post}(t_l))^2. \quad (12)$$

Symbolic conditions are used to model the occurrence of observable discrete events belonging to Σ_o and unobservable discrete events belonging to Σ_{uo} (faults, mission events, interaction with the environment, ...). $\Sigma = \Sigma_o \cup \Sigma_{uo}$ is defined as the set of discrete events of the system. Numerical conditions model an observable change of continuous dynamics. System continuous dynamics are represented by differential equations.

A generic example of a system behavioral model is described in Figure 5. In this example, the system has three continuous behaviors represented by p_5^N, p_6^N, p_7^N and four health states p_1^S, p_2^S, p_3^S and p_4^S . Using Equation (12), five health modes are distinguishable.

Health modes $q_1 = (p_1^S, p_5^N)$ and $q_2 = (p_1^S, p_6^N)$ are two nominal modes switching from one to the other when sym-

bolic conditions $\Omega^S(t_1^S) = (occ(e_1))$ or symbolic conditions $\Omega^S(t_2^S) = (occ(e_2))$ are satisfied. Symbolic transitions t_1^S and t_2^S represent the occurrence of observable events $e_1 \in \Sigma_o$ and $e_2 \in \Sigma_o$ supporting a change of continuous dynamics between p_5^N and p_6^N , respectively. Health modes $q_3 = (p_2^S, p_6^N)$ and $q_4 = (p_3^S, p_6^N)$ are two degraded modes reachable from health mode q_1 by satisfying the symbolic conditions $\Omega^S(t_3^S) = (occ(f_1))$ and $\Omega^S(t_4^S) = (occ(f_2))$, respectively. Symbolic transitions t_3^S and t_4^S represent the occurrence of two unobservable fault events $f_1 \in \Sigma_{uo}$ and $f_2 \in \Sigma_{uo}$, respectively. Finally, $q_5 = (p_4^S, p_7^N)$ is a failure mode in which both events f_1 and f_2 occurred and that is reachable from the two degraded modes. $\Omega^N(t_5^S) = (occ(f_1))$ represent the occurrence of f_1 and $\Omega^S(t_6^S) = (occ(f_2))$ represent the occurrence of f_2 .

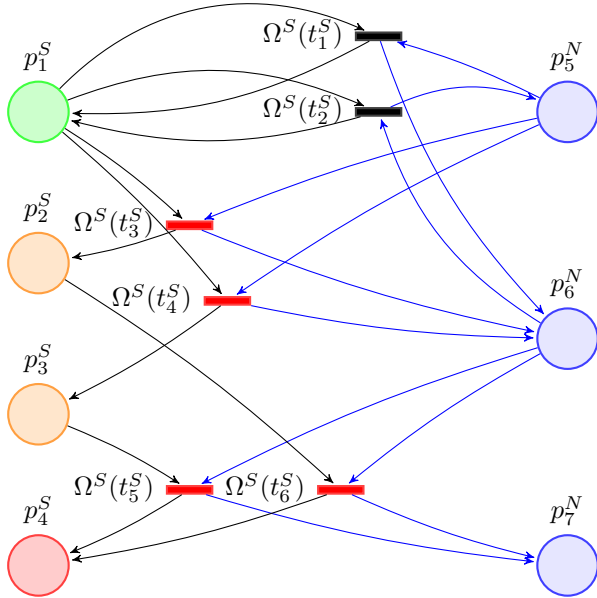


Figure 5. Example of system behavioral model using the HPPN-BL.

Example 3 The behavioral model of the three-tank system is limited to only two nominal modes and three failure modes for illustration purposes in Figure 6. Four dynamic equation sets are associated with D_1 , D_2 , D_{1f_1} and D_{2f_1} and five discrete health states are represented by nom_1 , nom_2 , $12f_4$, $1f_1$ and $2f_1$. Five health modes are defined. Health modes $q_1 = (nom_1, D_1)$ and $q_2 = (nom_2, D_2)$ are nominal modes in which the system switches from one to the other when the condition $occ(close_{v_{13}})$ or $occ(open_{v_{13}})$ is satisfied. Notations are simplified in the illustration with only the discrete event shown for the sake of readability. Three failure modes $q_3 = (12f_4, D_2)$, $q_4 = (1f_1, D_{1f_1})$, $q_5 = (2f_1, D_{2f_1})$ can be reached from nominal modes q_1 or q_2 by satisfying conditions $occ(f_1)$ or $occ(f_4)$ (simplified in f_1 and f_4 for illustration

purposes).

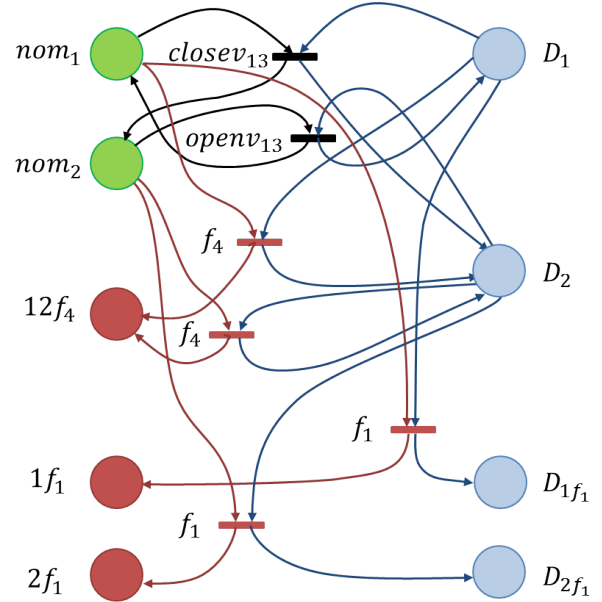


Figure 6. Three-tank partial behavioral model using the HPPN-BL.

The next section presents the methodology to build a state tracker object called a *diagnoser* from the behavioral system model.

4.2. Behavioral Diagnosis

In health monitoring context, diagnosis aims at tracking the system current health state. A common way in DES framework is to generate a diagnoser of the system from its model (Sampath et al., 1995). A diagnoser is basically a monitor that is able to process any possible observable event on the system. It consists in recording these observations and providing the set of possible faults whose occurrence is consistent with these observations.

Concerning hybrid systems, one approach is to build a hybrid diagnoser (Bayouh et al., 2008) from a hybrid automaton. The idea is to abstract the continuous part of the system to work with a discrete view of the system. This abstraction is done by using consistency tests, which take the form of a set of residuals. The diagnoser method is then directly applied to the resulting discrete event system. In previous work (Chanthery & Ribot, 2013), we extended this approach in order to integrate diagnosis and prognosis for hybrid systems. The main drawback of this approach is that the DES oriented diagnosis framework is not the most suited to incorporate the highly probabilistic prognosis task. Using the HPPN-BL, we succeeded in capturing knowledge-based and observation-based uncertainty. Consequently, we proposed a new diagnoser built from a HPPN-BL (Gaudel, Chanthery, Ribot, & Le Corrionc, 2014). The classical diagnoser is a fi-

nite state machine. Although this theoretical object is very interesting for studying properties of a system, such as diagnosability or controllability, it is absolutely not suited to embedded systems, because the diagnoser state number explodes for large models. Consequently, we choose to build a diagnoser based on a HPPN-BL model for the following reasons:

- it is not necessary to transform the specification of the system behavior another formalism,
- the HPPN model captures uncertainty, so the integration with prognosis becomes more natural,
- this representation is more compact than a hybrid automaton description, so the problem of embeddability of the diagnoser is reduced.

The diagnoser takes as inputs the HPPN-BL specifying the system behavior and the set of online observations on the system. The output of the diagnoser is an estimation of the system health state that takes the form of a marking of the diagnoser $\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N\}$.

4.2.1. Diagnoser Generation Based on HPPN-BL

Let us suppose that the system behavioral model is a HPPN-BL given by a tuple $\langle P, T, Pre, Post, X, F, \gamma, \Omega, M_0 \rangle$ as defined in Section 3.1.

The set of places of the diagnoser remains the same as the one of the system behavioral model. Concerning the transitions, two aspects have to be taken into account.

First, it is necessary to follow the system continuous dynamics with information resulting from the observed variables. Two sources of information can be taken into account:

- It is possible to generate a set of residuals from the set of differential equations C of the model. The residual generation has been widely studied in diagnosis studies. Fundamental results can be found in (Maquin et al., 1997). In the case where the system equations are linear, residuals can be generated by computing a set of analytical redundancy relations (ARR) with the parity space approach (Staroswiecki & Comtet-Varga, 2001). The parity space approach has been extended to multi-mode systems in (Cocquempot, El Meznyani, & Staroswiecki, 2004). In our case, we consider that it is possible to generate a residual r_i (or to simply have one by expert knowledge) for some numerical places p_i^N . Then, numerical conditions $\Omega^N(t_l)$ are associated with a transition t_l linking two numerical places p_j^N and p_i^N and carry a satisfaction test r_i , with $(i, j) \in \{1, \dots, |P^N|\}^2$ and $l \in \{1, \dots, |T|\}$. However, for some other places, only discrete information is available. The HPPN framework encompasses these characteristics and the diagnoser generation is adapted. If a residual r_i is available, this means that $\Omega^N(t_l)(\pi)$ is satisfied when r_i is satisfied for π .

Residuals allow for checking the consistency of observations with the model constraints. Since residuals are computed from observable continuous variables, they can be evaluated online with the incoming observations given by the sensors.

- The numerical correction process is used to update the prediction, by comparing the predicted particle values to observations of some continuous variables. This second way to check the consistency of the observed system behavior with the predicted one is implicit and does not appear in the diagnoser.

Secondly, because the diagnoser only captures the observable behavior of the system, a condition representing the occurrence of an unobservable discrete events would never be satisfied. The management of unobservable discrete event is done following these rules:

- R1. For any mixed transition $t_j^M \in T^M$ with symbolic conditions $\Omega^S(t_j^M)$ covering the occurrence of some unobservable events and some observable events, remove the symbolic part covering the occurrence of unobservable events, because these conditions would never be satisfied. The mixed transition t_j^M remains a mixed transition.
- R2. For any mixed transition $t_j^M \in T^M$ with symbolic conditions $\Omega^S(t_j^M)$ covering only the occurrence of unobservable events, remove the symbolic part of the transition. The mixed transition t_j^M is transformed in a numerical transitions $t_j^N \in T^N$.
- R3. For any symbolic transition $t_j^S \in T^S$ with symbolic conditions $\Omega^S(t_j^S)$ covering the occurrence of some unobservable events and some observable events, remove the symbolic part covering the occurrence of unobservable events. The transition t_j^S remains a symbolic transition.
- R4. For any symbolic transition $t_j^S \in T^S$ with symbolic conditions $\Omega^S(t_j^S)$ covering only the occurrence of some unobservable events, replace symbolic conditions with a test function always satisfied (*true()* function). The transition t_j^S remains a symbolic transition.

Concerning the observable discrete part of the system, occurrences of observable discrete events will be used as symbolic conditions.

Once the system behavioral model is defined and all numerical conditions are computed from the residual generation, the corresponding diagnoser can be generated with the following steps:

Step 1: Add the available numerical conditions $\Omega^N(t_j^S)$ to symbolic transition $t_j^S \in T^S$, with $j \in \{1, \dots, |T|\}$, if they exist. As a result, the symbolic transition t_j^S will be upgraded into a mixed transition $t_j^M \in T^M$.

Step 2: Manage the unobservable events with rules R1 to R4.

Ambiguity: The challenge of diagnosis is to have the ability to diagnose anticipated but unobservable faults in the system. In this context, modeling unobservable events can lead to ambiguity in the diagnoser. The occurrence of faults that cannot be distinguishable from the system observations will lead to ambiguous health states for the diagnoser. Therefore, a third step is needed during the diagnoser generation to track ambiguity. To do so, it is necessary to define a rule to merge two transitions of the same type. Two numerical transitions are mergeable if they are conditioned by the same change of dynamics and if they share the same symbolic places in their sets of inputs places.

Definition 6 (Mergeable numerical transitions)

Two numerical transitions $(t_i^N, t_j^N) \in (T^N)^2$, with $(i, j) \in \{1, \dots, |T^N|\}^2$ and $i \neq j$ are mergeable if:

$$(Pre(t_i^N) = Pre(t_j^N)) \wedge (Post(t_i^N) \cap P^N \cap Post(t_j^N) \neq \emptyset). \quad (13)$$

Note that condition (13) implies that the two transitions share the same numerical condition: $\Omega^N(t_i^N) = \Omega^N(t_j^N)$.

Definition 7 (Mergeable symbolic transitions)

Two symbolic transitions $(t_i^S, t_j^S) \in (T^S)^2$, with $(i, j) \in \{1, \dots, |T^S|\}^2$ and $i \neq j$ are mergeable if:

$$(Pre(t_i^S) = Pre(t_j^S)) \wedge (Post(t_i^S) \cap P^S \cap Post(t_j^S) \neq \emptyset). \quad (14)$$

Step 3: Merge all mergeable transitions while there are at least two mergeable transitions using the following definitions.

Definition 8 (Merging of two numerical transitions)

The merging of two mergeable numerical transitions $(t_i^N, t_j^N) \in (T^N)^2$, with $(i, j) \in \{1, \dots, |T^N|\}^2$ and $i \neq j$ is defined by two steps as follows:

(1) Creation of a new transition t_{ij}^N characterized by:

$$\begin{cases} Pre(t_{ij}^N) &= Pre(t_i^N) \\ Post(t_{ij}^N) &= Post(t_i^N) \cup Post(t_j^N) \\ \Omega^N(t_{ij}^N) &= \Omega^N(t_i^N), \end{cases} \quad (15)$$

(2) Addition of t_{ij}^N and deletion of t_i^N and t_j^N in T :

$$T = (T \setminus \{t_i^N, t_j^N\}) \cup \{t_{ij}^N\}. \quad (16)$$

Definition 9 (Merging of two symbolic transitions)

The merging of two mergeable symbolic transitions $(t_i^S, t_j^S) \in$

$(T^S)^2$, with $(i, j) \in \{1, \dots, |T^S|\}^2$ and $i \neq j$ is defined by two steps as follows:

(1) Creation of a new transition t_{ij}^S characterized by:

$$\begin{cases} Pre(t_{ij}^S) &= Pre(t_i^S) \\ Post(t_{ij}^S) &= Post(t_i^S) \cup Post(t_j^S) \\ \Omega^S(t_{ij}^S) &= \Omega^S(t_i^S), \end{cases} \quad (17)$$

(2) Addition of t_{ij}^S and deletion of t_i^S and t_j^S in T :

$$T = (T \setminus \{t_i^S, t_j^S\}) \cup \{t_{ij}^S\}. \quad (18)$$

The resulting diagnoser of the model given in Figure 5 is presented in Figure 7.

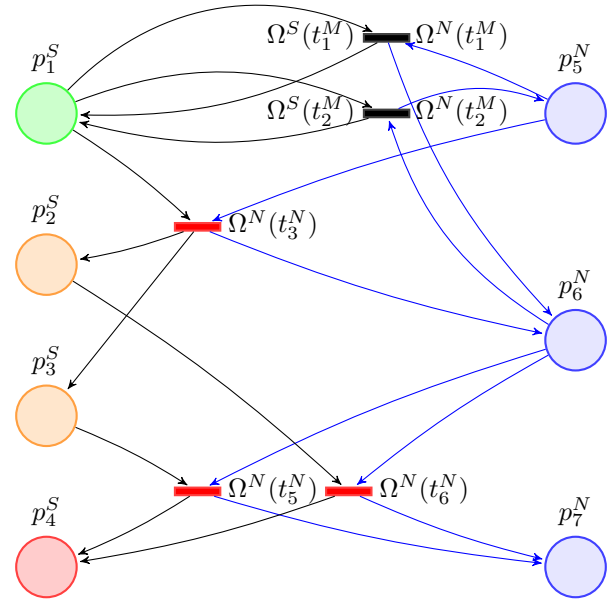


Figure 7. Example of system diagnoser using the HPPN-BL.

In Figure 7, Step 1 generates numerical conditions Ω^N to every transitions. All transitions are indeed supported by a change of dynamics that can be observed with the generation of residuals. All transitions are then upgraded into mixed transitions. As there are unobservable events, symbolic conditions associated with the occurrence of f_1 and f_2 are removed from the diagnoser model during Step 2, transforming t_3 , t_4 , t_5 and t_6 into numerical transitions. Finally, because transitions t_3 and t_4 are generating a change of dynamics from p_1^N to p_2^N , they are merged into one single numerical transition t_3^N .

Example 4 For the three-tank system example illustrated in Figure 8, transitions with the symbolic conditions $closev_{13}$ and $openv_{13}$ remain symbolic. Symbolic conditions f_1 and f_4 are deleted, as they are unobservable.

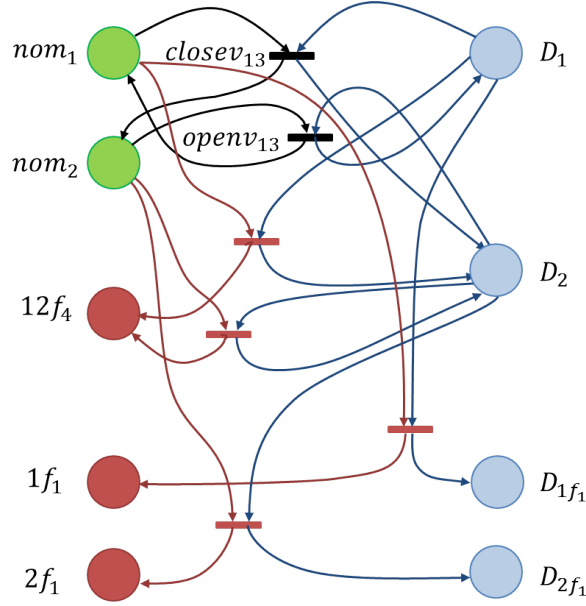


Figure 8. Three-tank system example diagnoser using the HPPN-BL.

4.2.2. Behavioral Diagnosis Result

The diagnosis result is given at each clock tick from the diagnoser state. By using the HPPN-BL and the state estimation process based on particle filtering described in Section 3.3, the diagnosis Δ_k at time k is the distribution of health mode beliefs, which depends on particle values and weights and is deduced from the marking of the diagnoser at time k :

$$\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N\}. \quad (19)$$

The marking $\hat{M}_k$ indicates the belief on the fault occurrences. It gives the same information as a classical diagnoser in terms of fault occurrences, with the same ambiguity. The difference is that in a classical diagnoser, every possible diagnosis has the same belief degree. With a HPPN-BL-based diagnoser, the ambiguity is valued by particle weights.

As a consequence, using diagnosis for health management becomes easier. In the case of classical diagnoser, it is indeed very difficult to “choose” a system belief state in case of decision making. In the case of a HPPN-BL-based diagnoser, each possible state of the system is valued, so it is easy to evaluate the most probable state at each clock tick.

Another advantage is that several types of uncertainty are taken into account.

1. Configuration pseudo-firing during numerical transition firing may represent uncertainty about the occurrence of an unobservable discrete event, especially a fault event.
2. Configuration and particle pseudo-firing during symbolic transition firing may represent uncertainty about the non-occurrence of an observed discrete event, i.e. a false ob-

servation.

3. Configuration pseudo-firing during mixed transition firing may represent uncertainty about the occurrence of an unobservable discrete event, supported by a change of continuous dynamics.
4. The noise added on particle values during the prediction step models uncertainty about system dynamic equations and continuous observations.

5. HYBRID LEVEL OF HYBRID PARTICLE PETRI NETS

The previous part describes a way to use the HPPN-BL to monitor the system health state based on its behavioral model. It could be interesting to look at a higher level of representation to illustrate a different level of dynamics, or to have a more aggregate view of the system. For instance, in health monitoring context, it is interesting to take into account the degradation dynamics. Getting some information about the degradation of the system is a significant advantage for elaborating a more precise diagnosis and to perform prognosis.

The next sections describe the hybrid level of Hybrid Particle Petri Nets formalism (HPPN-HL). The purpose of HPPN-HL is to represent some hybrid state characteristics, and not just either continuous dynamics or discrete states. For example, it will be possible to study properties such as stability or diagnosability of hybrid systems, or to monitor degradation and build a prognosis process. As stated before, this paper focuses on the degradation laws for system health monitoring, which depend on health modes. A set of dynamic equations is used to monitor the hybrid information we focus on. The update of the degradation value at each clock tick defines a degradation diagnosis function.

5.1. Hybrid Level

The HPPN-HL is defined as a tuple $\langle P^H, T^H, H, \mathcal{F}, \Omega^H \rangle$ where:

- P^H is the set of hybrid places.
- T^H is the set of hybrid transitions.
- $H \subset \mathbb{R}^n$ is the state space of the hybrid state vector.
- $\mathcal{F}$ is the set of dynamic equations of the system associated with hybrid places, representing hybrid state evolution.
- Ω^H is the set of conditions associated with the hybrid transitions.

A complete HPPN combines a behavioral level with a hybrid level and is thus defined as a tuple $\langle P, T, Pre, Post, X, C, H, \mathcal{F}, \gamma, \Omega, M_0 \rangle$.

Hybrid places are used to compose the HPPN-HL and represent possible hybrid states. A hybrid state is a couple (p_i^S, p_j^N) . For the sake of clarity, $p_l^H = (p_i^S, p_j^N)$ denotes that a hybrid

place p_l^H represents the hybrid state (p_i^S, p_j^N) . Because hybrid states are combinations of symbolic places and numerical places, the hybrid state set of a given behavioral model is always finite. However, only couples that are part of the set of output places of the same transition are considered as hybrid states:

$$p_l^H = (p_i^S, p_j^N) \in P^H \text{ if } \exists t_m \in T, (p_i^S, p_j^N) \in (Post(t_m))^2. \quad (20)$$

Hybrid states that do not satisfy Equation (20) are considered as *intermediate states*. This means there is no information in the model about them.

A hybrid place is marked by hybrid tokens $h_k^i = [s_k^i, d_k^i]$ with $i \in \{1, \dots, |M_k^H|\}$ where M_k^H is the set of all the hybrid tokens in the net at time k . A hybrid token is defined by a couple $s_k^i = (\delta_k^j, \pi_k^l)$ of tokens running in the HPPN-BL and its corresponding hybrid state vector $d_k^i \in H$. The whole marking at time k of the HPPN is $M_k = \{M_k^S, M_k^N, M_k^H\}$. Hybrid tokens are created and deleted as follows.

Creation: Because of their dependency on configurations and particles, hybrid tokens are created at the same time as the creation of a configuration or a particle. If a hybrid token h depends on a particle π that is duplicated during the prediction step of the state estimation process in a new particle π' , then h is also duplicated in h' but h' depends on the new particle π' .

Deletion: A hybrid token h depending on a configuration δ and a particle π is deleted when δ or π is deleted during the online process of the HPPN-BL.

Considering the two rules above, the HPPN-HL online process totally depends on the HPPN-BL online process but both processes are simultaneous.

Any hybrid place is linked with all the other hybrid places through a hybrid transition $t_j^H \in T^H$.

$\forall p_i \in P$, $M_k(p_i)$ is the set of tokens in p_i at time k and $m_k(p_i) = |M_k(p_i)|$ is the number of tokens in p_i at time k .

Definition 10 A hybrid transition $t_j^H \in T^H$ is fire-enabled at time k if:

$$\exists p_i^H \in Pre(t_j^H), \quad m_k(p_i^H) \geq Pre(i, j) \quad (21)$$

A hybrid place is associated with a set of dynamic equations representing a hybrid state characteristics. The idea is to let a hybrid token $h^i = [s^i, d^i]$ evolve in the HPPN-HL in accordance to symbolic and numerical places in which its associated configuration δ_k^j and its associated particle π_k^l , with $s^i = (\delta^j, \pi^l)$, are evolving.

To formally define the firing of hybrid transitions, we need to

define the following notations: $P(\delta^j) = p_j^S$ and $P(\pi^l) = p_l^N$ denote the projections of δ^j and π^l on the set of places P , $P(s^i) = (p_j^S, p_l^N)$ denotes the hybrid place of a couple $s^i = (\delta^j, \pi^l)$.

Each hybrid transition carries a set of hybrid conditions $\Omega^H(t_j^H)$ that are satisfied if $\Omega^H(t_j^H)(h^i) = 1$. Hybrid tokens h^i are moved to another hybrid place p'^H if $P(s^i) = p'^H$:

$$\forall h^i = [s^i, d^i], \quad \Omega^H(t_j^H)(h^i) = \begin{cases} 1 & \text{if } P(s^i) = p'^H \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

$\mathcal{S}_k^H(p^H)$ is the set of hybrid tokens in p^H satisfying the conditions $\Omega^H(t_j^H)$ at time k .

Definition 11 The firing of a fire-enabled hybrid transition $t_j^H \in T^H$ at time k is defined by:

$$\begin{cases} M_{k+1}^H(p^H) = M_k^H(p^H) \setminus \mathcal{S}_k^H(p^H) \\ M_{k+1}^H(p'^H) = M_k^H(p'^H) \cup \mathcal{S}_k^H(p^H). \end{cases} \quad (23)$$

An example of hybrid transition firing in a HPPN-HL is shown in Figure 9. There are two hybrid places $p_1^H = (p_1^S, p_1^N)$ and $p_2^H = (p_2^S, p_2^N)$. At time k , the two hybrid tokens $h_k^1 = [s_k^1, d_k^1]$ and $h_k^2 = [s_k^2, d_k^2]$ are following the characteristics of the hybrid state represented by p_1^H , so hybrid transitions t_1^H and t_2^H are fire-enabled. $P(s_k^1) = (p_1^S, p_1^N)$ but $P(s_k^2) = (p_2^S, p_2^N)$ so $\Omega^H(t_2^H)(h^2)$ is satisfied and h^2 is moved to p_2^H . h^2 thus follows at time $k+1$ the characteristics of the hybrid state $p_2^H = (p_2^S, p_2^N)$.

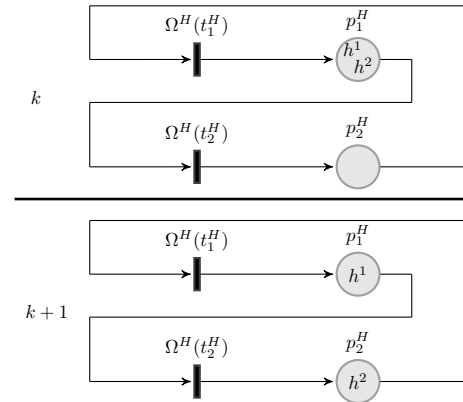


Figure 9. Illustration of firing rules of fire-enabled hybrid transitions.

Hybrid tokens evolve according to laws defined for hybrid places. HPPN-BL tokens are changing places during the prediction step of the state estimation process (see Section 3.3), and then hybrid tokens are simultaneously changing places and their values are updated as follows:

$$\forall h_{k+1|k}^i \in \hat{M}_{k+1|k}^H(p_j^H), \quad d_{k+1}^i = F_{k+1}^j(d_k^i), \quad (24)$$

where $F_k^j \in \mathcal{F}$ is the set of dynamic equations associated with the hybrid place p_j^H . Because d_{k+1}^i depends on d_k^i , the continuity of the value d^i can be ensured.

Figure 10 illustrates the evolution of the value d^2 of hybrid token h^2 of Figure 9. It shows that d_{k+1}^2 is computed with dynamic equations F_{k+1}^2 . F_{k+1}^2 is associated with p_2^H and depends on d_k^2 , the value of d^2 at time k . This ensures the continuity between F_k^1 and F_{k+1}^2 at time $k + 1$.

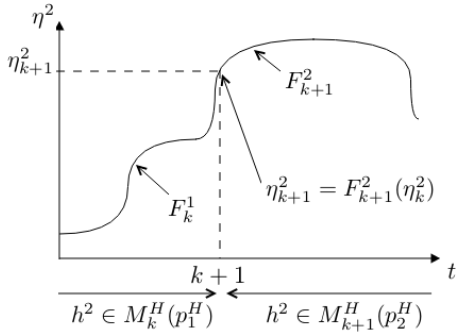


Figure 10. Illustration of the continuities of hybrid token values.

If $\mathcal{F}$ is not empty, the values d_k^i can be taken into account in the decision making process at time k that determines the marking at time $k + 1$ of the HPPN-BL.

6. APPLICATION TO DEGRADATION MONITORING

This section introduces a way to represent uncertainty about degradation for each system health mode using probability measures.

6.1. Degradation Modeling

In the context of health monitoring, the HPPN-HL formalism is used to include degradation laws in the system model. We propose to consider health modes as hybrid states of a HPPN. The system description is enriched with a set of degradation laws modeling the degradation depending on health mode stress levels. The set of degradation laws is supposed to be accurately known. $\mathcal{F} = \{F^{p_i^H}, p_i^H \in P^H\}$ is the set of degradation laws associated with hybrid places of the system. $F^{p_i^H}$ is a vector of degradation laws for each anticipated fault in the hybrid place $p_i^H = (p_i^S, p_j^N)$:

$$F^{p_i^H}(t) = \begin{bmatrix} f_1^{p_i^H}(t) \\ f_2^{p_i^H}(t) \\ \vdots \\ f_{n_f}^{p_i^H}(t) \end{bmatrix}, \quad (25)$$

where $f_j^{p_i^H}$ represents the probability distribution of the fault

f_j at any time in the hybrid place p_i^H and n_f is the number of anticipated faults.

Example 5 The right side of Figure 11 represents the HPPN-HL of the three-tank system model illustrated by Figure 6. This partial model has five health modes (see Section 4.1), thus the corresponding HPPN-HL has five hybrid places $p_1^H = (nom_1, D_1)$, $p_2^H = (nom_2, D_2)$, $p_3^H = (12f_4, D_2)$, $p_4^H = (1f_1, D_{1f_1})$, and $p_5^H = (2f_1, D_{2f_1})$. Five hybrid transitions $t_1^H \dots t_5^H$ deliver accesses to the five hybrid places when associated hybrid conditions are satisfied (Equation (22)). Not all transitions are represented in Figure 11 for the sake of readability. Fault probability distributions F^j associated with each hybrid place p_j^H are represented with the 2-parameter Weibull model. By setting $\beta = 1$, the Weibull model describes random failures and is similar to the exponential law. The second parameter η of the Weibull model depends on the system stress level in each health mode associated with a hybrid place:

$$f_j^{p_i^H}(t) = W(t, \eta_j(p_i^H)) = \int_{t_0}^t \frac{1}{\eta_j(p_i^H)} \exp\left(-\frac{t}{\eta_j(p_i^H)}\right) dt. \quad (26)$$

Fault probability distributions for leak faults f_1 , f_2 and f_3 are defined with three different value for parameter η ($\eta_a = 100000$, $\eta_b = 80000$, $\eta_c = 60000$) representing three different stress levels defined for the system health modes. Fault probability distributions associated with each hybrid place of Figure 11 are given in Table 2. All degradation laws of the three-tank system are provided in the appendix.

Fault probability distributions for f_4 and f_5 mainly evolve with the number of closings/openings. Thus, the parameter η_{13} for the valve v_{13} (resp. η_{32} for the valve v_{32}) depends on a cycle coefficient α_{13} (resp. α_{32}) that increases after each occurrence of discrete events $openv_{13}$ or $closev_{13}$ (resp. $openv_{32}$ or $closev_{32}$):

$$\begin{aligned} f_4^{p_i^H}(t) &= W(t, \eta_{13}(\alpha_{13})), \\ f_5^{p_i^H}(t) &= W(t, \eta_{32}(\alpha_{32})). \end{aligned} \quad (27)$$

At first, the system is supposed to be in health mode (nom_1, D_1) and $\alpha_{13} = \alpha_{32} = 1$, $\eta_{13} = \eta_{32} = 20000$.

6.2. Diagnosis Based on HPPN

The diagnoser extended with the HPPN-HL is called a HPPN-based diagnoser. The HPPN-based diagnoser generation step does not modify the degradation model defined in Section 6.1. It is a HPPN that monitors both the behavior and the degradation of the system.

Using a HPPN-based diagnoser, the diagnosis Δ_k of the system at time k is the complete marking of the diagnoser. Equation (28) represents the distribution of health mode beliefs

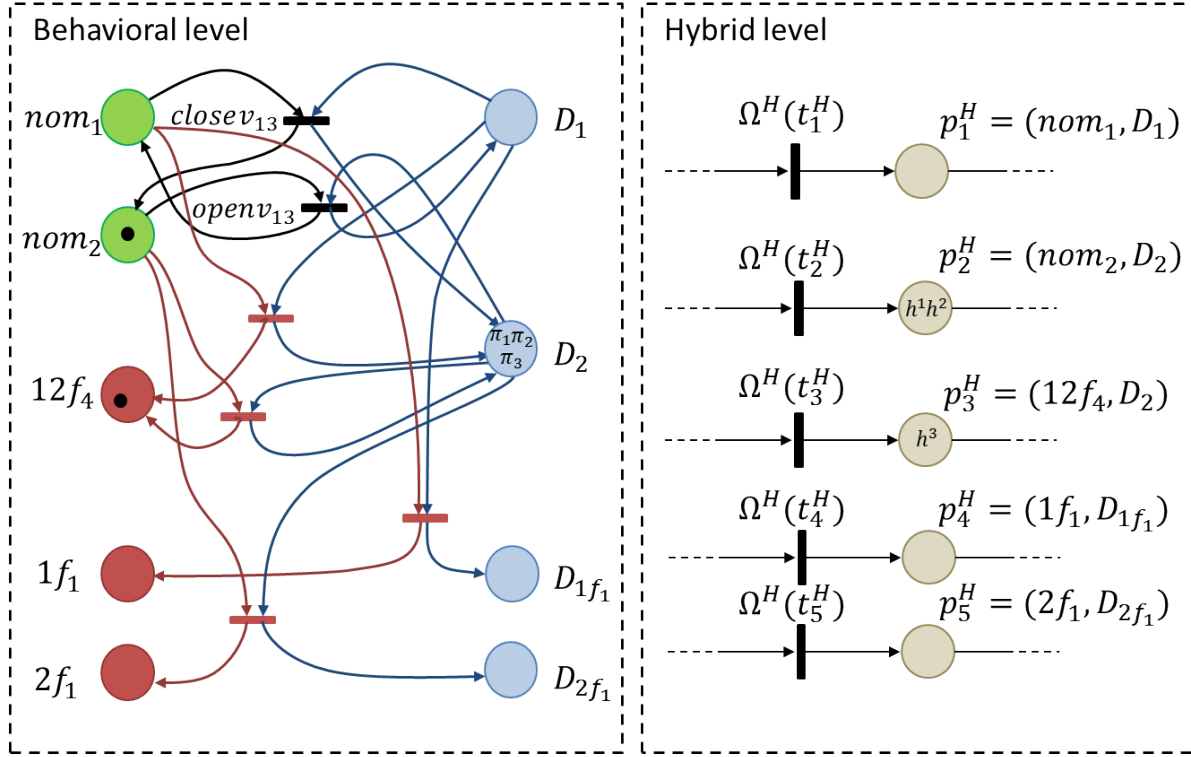


Figure 11. Three-tank system example HPPN-based partial diagnoser.

and the system degradation for each mode:

$$\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N, \hat{M}_k^H\}. \quad (28)$$

Example 6 Figure 11 shows a part of the HPPN-based diagnoser for the three-tank system. It illustrates the interactions between the HPPN-BL and the HPPN-HL of the diagnoser. At time k , two configurations and three particles are running in the HPPN-BL. One configuration is in the symbolic place nom_2 and the other one in the symbolic place $12f_4$. Three particles π^1 , π^2 and π^3 are in numerical place D_2 . Three hybrid tokens are running in the HPPN-HL. h^1 and h^2 are in the hybrid place p_2^H because they are linked to the configuration in nom_2 and respectively π^1 and π^2 . These two hybrid tokens follow degradation laws $F^{p_2^H}$. h^3 is in the hybrid place p_4^H because it is linked to the configuration in $12f_4$ and π^3 . This hybrid token follows degradation laws $F^{p_4^H}$. For this example, the result of diagnosis can be written as follows:

$$\Delta_k = \{(nom_2, D_2, h^1), (nom_2, D_2, h^2), (12f_4, D_2, h^3)\}, \quad (29)$$

where the degradation vectors of hybrid tokens h^1 , h^2 and h^3 at time k are respectively $d_k^1 = F^{p_2^H}(d_{k-1}^1)$, $d_k^2 = F^{p_2^H}(d_{k-1}^2)$ and $d_k^3 = F^{p_4^H}(d_{k-1}^3)$.

To sum up, the main steps of the proposed method are re-

minded in Algorithms 1 and 2.

Algorithm 1 Offline process

- 1: $model \leftarrow CreateHPPNModel()$
 - 2: $diagnoser \leftarrow GenerateHPPNDiagnoser(model)$
-

Algorithm 2 Online process

Input: $diagnoser, M_0$

- 1: **for all** O_k **do**
 - 2: $\hat{M}_k \leftarrow StateEstimation(diagnoser, \hat{M}_{k-1}, O_k)$
 - 3: $\Delta_k \leftarrow \hat{M}_k$
 - 4: **end for**
-

The first offline step (Algorithm 1, line 1) is to model the system considered using the HPPN framework (see Sections 4.1 and 6.1). The second offline step (Algorithm 1, line 2) is to generate the HPPN-based diagnoser from the system model (see Sections 4.2 and 6.2). Then the online process (Algorithm 2) uses consecutive observations to update the diagnoser marking and compute diagnosis (see Sections 3.3, 4.2 and 6.2).

From the HPPN-based diagnoser are provided two estimations:

- The belief on the system *health modes* takes the form of a marking $\{\hat{M}_k^S, \hat{M}_k^N\}$ of the HPPN-BL,

Table 2. Three-tank system example: health modes degradation laws.

Health mode	Valve state	Degradation laws
$p_1^H = (nom_1, D_1)$	$v_{13} = open$ $v_{32} = open$	$F^{p_1^H} \begin{cases} f_1^{p_1^H}(t) = W(t, \eta_a) \\ f_2^{p_1^H}(t) = W(t, \eta_a) \\ f_3^{p_1^H}(t) = W(t, \eta_a) \end{cases}$
$p_2^H = (nom_2, D_2)$	$v_{13} = closed$ $v_{32} = open$	$F^{p_2^H} \begin{cases} f_1^{p_2^H}(t) = W(t, \eta_c) \\ f_2^{p_2^H}(t) = W(t, \eta_a) \\ f_3^{p_2^H}(t) = W(t, \eta_a) \end{cases}$
$p_3^H = (12f_4, D_2)$	$v_{13} = closed$ $v_{32} = open$	$F^{p_3^H} \begin{cases} f_1^{p_3^H}(t) = W(t, \eta_c) \\ f_2^{p_3^H}(t) = W(t, \eta_a) \\ f_3^{p_3^H}(t) = W(t, \eta_a) \end{cases}$
$p_4^H = (1f_1, D_{1f_1})$	$v_{13} = open$ $v_{32} = open$	$F^{p_4^H} \begin{cases} f_1^{p_4^H}(t) = 1 \\ f_2^{p_4^H}(t) = W(t, \eta_a) \\ f_3^{p_4^H}(t) = W(t, \eta_a) \end{cases}$
$p_5^H = (2f_1, D_{2f_1})$	$v_{13} = closed$ $v_{32} = open$	$F^{p_5^H} \begin{cases} f_1^{p_5^H}(t) = 1 \\ f_2^{p_5^H}(t) = W(t, \eta_a) \\ f_3^{p_5^H}(t) = W(t, \eta_a) \end{cases}$

- The belief on the system *degradation state* takes the form of a marking of the HPPN-based diagnoser $\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N, \hat{M}_k^H\}$.

The marking value $\{\hat{M}_k^S, \hat{M}_k^N\}$ represents the belief on the health modes through a probability distribution. The marking value $\hat{M}_k^H$ represents the degradation state through a degradation distribution over the health modes. Because each hybrid token depends on a particle and a configuration, its degradation value is linked with the belief of its health mode. Consequently, the health mode belief and the degradation values can be considered in case of decision making in the context of health management.

7. CASE STUDY

The HPPN-based diagnoser generation has been implemented in Python 3.4. The tests described below were performed under Linux 4.9.2(x86_64 – linux – gnu) on a 1.90 GHz i5 – 3437U CPU. Simulations use one CPU and a maximum of 2GB of RAM.

7.1. Three-Tank System Diagnoser

The three-tank model has been loaded, including its behavioral description and its degradation laws. The tank sections are $S_1 = S_2 = S_3 = 0.0154m^2$ and the pipe section between tanks is $S_n = 5.10^{-5}m$. Noise is modeled as Gaussian noise. Gaussian noise with a 0 mean value and a standard deviation of 0.01m has been added to each state equation. The mea-

sure of h_1 is subjected to Gaussian noise with a 0 mean and a standard deviation of 0.01m. The measure of the water mass is subjected to Gaussian noise with 0 mean and a standard deviation of 0.1kg. The initial state is $H_0 = [H_{10}, H_{20}, H_{30}] = [0.48, 0.55, 0.58]$ and valves v_{13} and v_{32} are open. The simulation sampling period is 60s because of the slow system continuous dynamics. The complete three-tank model has 40 symbolic places, 28 numerical places and 120 transitions. When using the merging rules in the diagnoser generation, the HPPN-based diagnoser has only 40 transitions, which represents a transition number reduction of about 66%. It shows that the diagnoser generation method is efficient and will be able to manage models for real systems. The initial symbolic and numerical places are respectively nom_1 and D_1 and the initial hybrid places is $p_1^H = (nom_1, D_1)$.

7.2. Results

Three scenarios have been simulated. Scenario 1 illustrates that the method is robust to occurrences of unobservable events, such as fault events. f_1 occurs at 660s and the system ends in failure mode $(1f_1, D_{1f_1})$. Figure 12 shows the mean value of the diagnoser particles compared to the real water levels. Areas around mean values represent particle value uncertainty.

Figure 13 shows the diagnoser health mode belief degrees compared to the real one at different times of the simulation. The *black dot* indicates the real health mode of the system

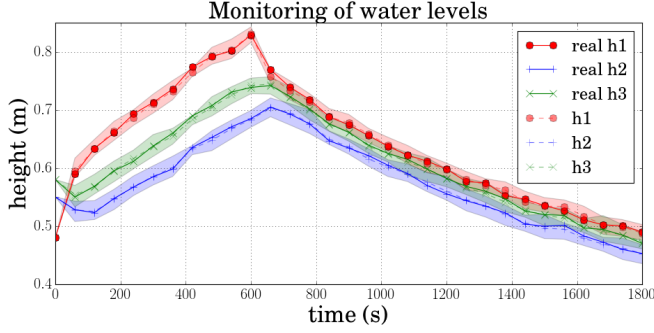


Figure 12. Scenario 1: monitoring of water levels.

at every clock tick. The *white dot* indicates the health mode with the highest belief degree and the *grey dots* indicates the other health mode belief degrees. Health mode belief degrees are computed using a weighting function, which depends on the distributions Pr_{DN} and Pr_{DS} calculated during the state estimation process (see Section 3.3).

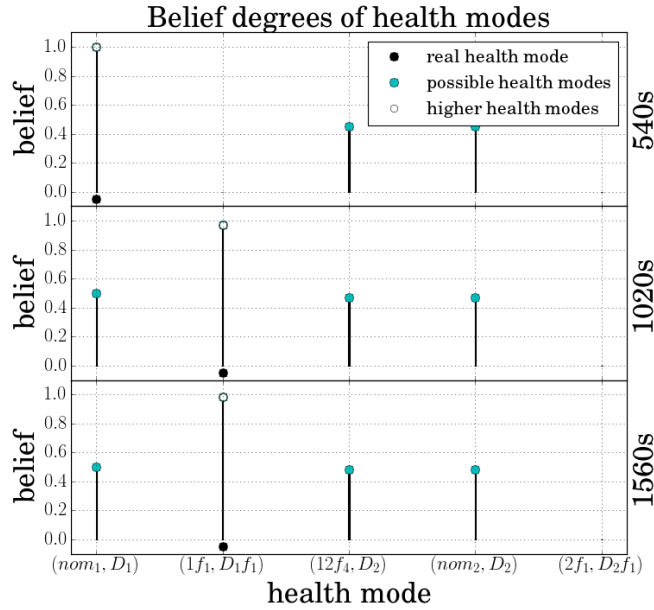


Figure 13. Scenario 1: health mode belief degrees.

Figure 13 shows that the health mode with the highest belief degree is always the real one (nom_1, D_1) at 540s and then ($1f_1, D_1f_1$) at 1020s and 1560s. The leak in tank T_1 was thus well diagnosed. The diagnoser also considers other health modes because of the similarity between continuous dynamics. However, the discrete part of the system is used to restrain the health mode distribution and exclude health modes that do not match with observations and dynamic equations, such as ($1f_1, D_1f_1$) at 540s and ($21f_1, D_2f_1$) during the whole simulation.

Scenario 2 illustrates that the proposed method is robust to oc-

currences of unobservable events where continuous dynamics are ambiguous. $close_{v_{13}}$ occurs and is observed at 660s and valve v_{13} is closed, leading the system to health mode (nom_2, D_2). Then at 1920s, f_4 occurs and the system ends in failure mode ($12f_4, D_2$). Finally, $open_{v_{13}}$ occurs and is observed at 3180s, but the health mode of the system does not change.

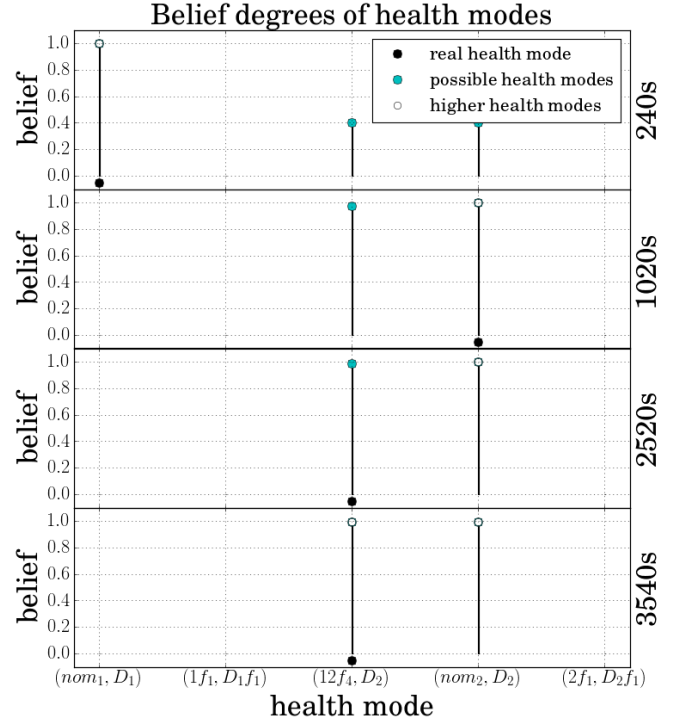


Figure 14. Scenario 2: health mode belief degrees.

Figure 14 shows the diagnoser health mode belief degrees compared to the real one at different times of the simulation. At 240s, the health mode with the highest belief degree is the real one (nom_1, D_1). At 1020s, both health modes ($12f_4, D_2$) and (nom_2, D_2) have high belief degrees because the diagnoser particles follow the same dynamic equations D_2 properly. The belief degree of (nom_2, D_2) is slightly higher than the one of ($12f_4, D_2$) because configurations in symbolic place nom_2 entirely match with the discrete observations ($close_{v_{13}}$ at 660s). Still, the possibility that the observation of $close_{v_{13}}$ at 660s could have been a false observation is kept in the marking, and thus in the diagnosis and in the computed belief degree distribution. At 2520s, the health mode with the highest belief degree is still (nom_2, D_2) because the occurrence of f_4 does not involve any change of continuous dynamics. Once again, the diagnosis carries the possibility that the system is in the real health mode ($12f_4, D_2$) with a high belief degree. Considering that f_4 is unobservable and the continuous dynamics of the two health modes are similar, it is the best diagnosis that can be pro-

duced at this time. After $open_{v_{13}}$ is observed, configurations in nom_2 do not match with the observations as well as before and then the belief degrees of the two health modes become the same. Scenario 2 shows that the diagnoser is robust to occurrences of fault events even in case of ambiguous continuous dynamics.

Scenario 3 illustrates that the proposed method is robust to knowledge-based uncertainty, such as faulty modeling of the discrete behavior of the system or unreliable continuous dynamics, and observation uncertainty, such as occurrences of unobserved observable events or false observations. In this scenario, the standard deviations of state equation noise and the noise associated with the measure of h_1 have been quintupled and the simulation sampling period has been doubled. Events $close_{v_{13}}$ and $open_{v_{13}}$ occur respectively at 720s and 2040s but none of them are observed. At 3360s, f_1 occurs, leading the system to failure mode $(1f_1, D_1f_1)$.

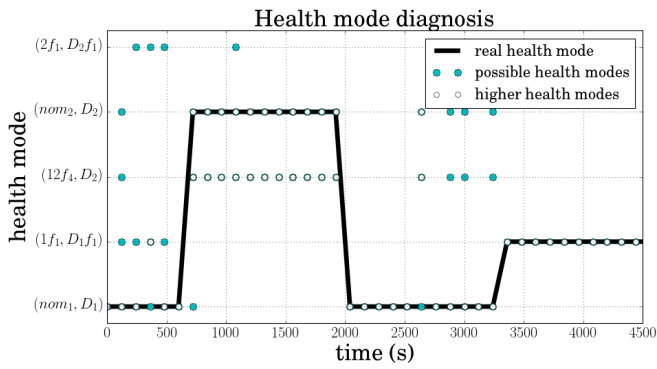


Figure 15. Scenario 3: diagnosis.

Figure 15 shows the diagnosis of the system over time. The diagnoser monitors the three-tank health mode properly even if observable events are not observed and detects the fault f_1 in only one sample period. It also shows that the amplification of noise has consequences on the belief degree distribution: at 480s and 2040s, the real health mode is not the highest one but still remains in the diagnosis result. From 720s to 1920s, the diagnosis is ambiguous: $(12f_4, D_2)$ and (nom_2, D_2) have the same belief degree due to the non observation of $close_{v_{13}}$. However, some information such as f_4 probability could be useful for decision making during this period. Figure 16 shows fault f_4 degradation monitoring by hybrid tokens in hybrid place p_1^H from 0s to 600s. Degradation is monitored by estimating fault f_4 occurrence probability and is the mean of all hybrid token values. At 600s, f_4 occurrence probability is around 0.006. This very low probability could be taken into account in case of decision making at time 720s, and then disambiguate diagnosis by preferring health mode (nom_2, D_2) .

Faults are always detected in one sampling period because HPPN consider all possibilities during the online process pre-

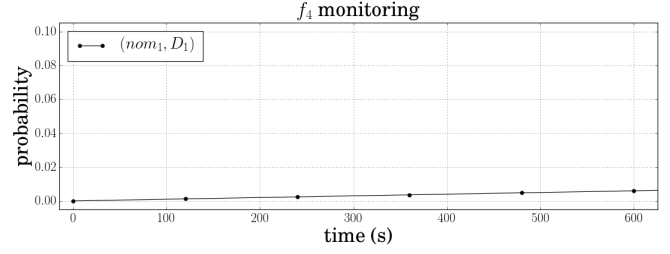


Figure 16. Scenario 3: degradation monitoring in mode (nom_1, D_1) .

Table 3. Diagnosis computation time for different scenarios.

Scenario		computation time (s)	$ M^S $	$ M^N $	$ M^H $
1	average	1.123	4	330	1848
	minimum	0.581	2	200	400
	maximum	2.595	9	696	6264
2	average	7.305	63	774	50427
	minimum	1.802	8	600	4800
	maximum	18.287	130	800	104000
3	average	8.495	118	524	66349
	minimum	0.906	8	200	1600
	maximum	118.509	2304	607	1398528
total average		6.274	65	610	44105

diction step and keep matching marking during the correction step. Diagnosis computation times and token numbers are given in Table 3. $|M^S|$, $|M^N|$ and $|M^H|$ respectively represent the diagnoser configuration, particle and hybrid token number at the time the diagnosis is performed.

Table 3 shows that the average diagnosis computation time is around 6.5s and the average number of all the diagnoser tokens at any time is 44780. These metrics point out that the diagnosis computation time increases with the diagnoser token number but remains acceptable compared to the system time constraint.

Simulation results show that HPPN-based diagnosis is robust to different types of uncertainty. In Scenario 1, the diagnoser detects and identifies properly a leak fault in tank T_1 , but always considers possibilities of other modes having almost the same dynamic equations. Scenario 2 illustrates how the diagnoser deals with undetectable fault occurrences. Finally, Scenario 3 shows that the diagnoser is robust in case of noisy knowledge and observations and introduces the need to consider degradation monitoring in case of diagnosis ambiguity.

8. CONCLUSION AND FUTURE WORK

This article illustrates the HPPN approach to process health monitoring of hybrid systems under uncertainty. The HPPN approach takes into account knowledge-based and observation-based uncertainty.

A HPPN-based diagnoser has been defined to monitor both discrete and continuous behaviors of the system. The HPPN-BL-based diagnoser is enriched to consider a level representing a hybrid property depending on the hybrid state of the system. The article focuses on the degradation property, in order to introduce concepts that will be useful to perform prognosis and health management of hybrid systems under uncertainty. In addition, diagnosis results can be used as probability distributions for decision making. Throughout the paper, concepts are illustrated with a three-tank system example. The HPPN-based diagnoser deals with event occurrence possibility and knowledge imprecision. Three scenarios have been tested to illustrate the proposed method advantage. The HPPN-based diagnoser results are consistent with the expected ones.

In future work, the HPPN-based diagnoser will be tested on a real three-tank system. The proposed method may be adapted to meet real system constraints. We also aim at formalizing and developing a prognosis process that will interleave diagnosis and prognosis methods to obtain more accurate results.

NOMENCLATURE

P	the set of places of a HPPN
P^N	the set of numerical places of a HPPN
P^S	the set of symbolic places of a HPPN
P^H	the set of hybrid places of a HPPN
T	the set of transitions of a HPPN
T^N	the set of numerical transitions of a HPPN
T^S	the set of symbolic transitions of a HPPN
T^M	the set of mixed transitions of a HPPN
T^H	the set of hybrid transitions of a HPPN
$Pre(t)$	the set of input places of a transitions $t \in T$
$Post(t)$	the set of output places of a transitions $t \in T$
X	the state space of the numerical state vector
C	the set of dynamic equation sets associated with numerical places, representing continuous state evolution of the system
H	the state space of the hybrid state vector
$\mathcal{F}$	the set of dynamic equations associated with hybrid places
Ω	the set of conditions of a HPPN
Ω^N	the set of numerical conditions of a HPPN
Ω^S	the set of symbolic conditions of a HPPN
Ω^H	the set of hybrid conditions of a HPPN
M_k	the marking of a HPPN at time k
M_k^N	the set of particles of a HPPN at time k
M_k^S	the set of configurations of a HPPN at time k
M_k^H	the set of hybrid tokens of a HPPN at time k
$M_k(p)$	the marking of the place p at time k
$m_k(p)$	the numbers of tokens in the place p at time k
$\mathcal{S}_k(p)$	the tokens in $p \in Pre(t)$ that satisfy the conditions $\Omega^N(t)$ at time k
π_k	a particle of M_k^N in a HPPN at time k with a value $x_k \in X$ and a weight $w_k \in [0, 1]$
δ_k	a configuration of M_k^S in a HPPN at time k
h_k	a hybrid token of M_k^H in a HPPN at time k with a value $d_k \in H$ and associated with a couple of tokens $s_k = (\delta_k, \pi_k)$
$P(s_k)$	the hybrid place of a couple of token $s = (\delta_k, \pi_k)$ at time k
Σ	the set of discrete events
Σ_o	the set of observable discrete events
Σ_{uo}	the set of unobservable discrete events
O_k	the set of observations at time k
O_k^N	the set of continuous observations at time k
O_k^S	the set of discrete observations at time k
Q	the set of health modes
Δ_k	the diagnosis at time k

REFERENCES

- Basile, F., Chiacchio, P., & Tommasi, G. D. (2009). Fault diagnosis and prognosis in Petri nets by using a single generalized marking estimation. In *7th ifac symposium*

- on fault detection, supervision and safety of technical processes (*safeprocess*). Barcelona, Spain.
- Bayouddh, M., Travé-Massuyes, L., & Olive, X. (2008). Hybrid systems diagnosis by coupling continuous and discrete event techniques. In *Proceedings of the IFAC World Congress* (pp. 7265–7270). Seoul, Korea.
- Boubour, R., Jard, C., Aghasaryan, A., Fabre, E., & Benveniste, A. (1997, December). A Petri net approach to fault detection and diagnosis in distributed systems. part i: application to telecommunication networks, motivations, and modelling. In *the 36th conference on decision & control*. San Diego, California, USA.
- Cabasino, M. P., Giua, A., & Seatzu, C. (2014). Diagnosability of discrete-event systems using labeled Petri nets. *IEEE Transactions on Automation Science and Engineering*, 11(1), 144–153.
- Chanthery, E., & Ribot, P. (2013). An integrated framework for diagnosis and prognosis of hybrid systems. In *3rd Workshop on Hybrid Autonomous System (HAS)*. Roma, Italy.
- Cocquempot, V., El Meznyani, T., & Staroswiecki, M. (2004). Fault detection and isolation for hybrid systems using structured parity residuals. In *5th Asian Control Conference*. (Vol. 2, pp. 1204–1212).
- Gaudel, Q., Chanthery, E., & Ribot, P. (2014). Health monitoring of hybrid systems using hybrid particle Petri nets. In *Annual conference of the prognostics and health management society 2014*.
- Gaudel, Q., Chanthery, E., Ribot, P., & Le Corrionc, E. (2014). Hybrid systems diagnosis using modified particle Petri nets. In *Proceedings of the 25th International Workshop on Principles of Diagnosis (DX'14)*. Graz, Austria.
- Genc, S., & Lafortune, S. (2007, April). Distributed diagnosis of place-bordered Petri nets. *IEEE Transactions on Automation Science and Engineering*, 4(2), 206–219.
- Henzinger, T. (1996). The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science* (pp. 278–292).
- Jianxiong, W., Xudong, X., Xiaoying, B., Chuang, L., Xiangzhen, K., & Jianxiang, L. (2013). Performability analysis of avionics system with multilayer hm/fm using stochastic Petri nets. *Chinese Journal of Aeronautics*, 26(2), 363–377.
- Koutsoukos, X., Kurien, J., & Zhao, F. (2002, August). Monitoring and diagnosis of hybrid systems using particle filtering methods. In *Proceedings of the 15th international symposium on mathematical theory of networks and systems, mtns 2003*. Notre Dame, IN.
- Lesire, C., & Tessier, C. (2005). Particle Petri nets for aircraft procedure monitoring under uncertainty. In *Applications and Theory of Petri Nets* (pp. 329–348). Springer.
- Maquin, D., Cocquempot, V., Cassar, J.-P., Staroswiecki, M., Ragot, J., et al. (1997). Generation of analytical redundancy relations for fdi purposes. In *Ifac symposium on diagnostics for electrical machines, power electronics and drives, sdemped'97* (pp. 86–93).
- Ru, Y., & Hadjicostis, C. N. (2009). Fault diagnosis in discrete event systems modeled by partially observed Petri nets. *Discrete Event Dynamic Systems*, 19(4), 551–575.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9), 1555–1575.
- Soldani, S., Combacau, M., Subias, A., & Thomas, J. (2007). On-board diagnosis system for intermittent fault: Application in automotive industry. In *7th ifac international conference on fieldbuses and networks in industrial and embedded systems* (Vol. 7-1, p. 151-158). doi: 10.3182/20071107-3-FR-3907.00021
- Staroswiecki, M., & Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5), 687–699.
- Zouaghi, L., Alexopoulos, A., Wagner, A., & Badreddin, E. (2011a). Modified particle Petri nets for hybrid dynamical systems monitoring under environmental uncertainties. In *IEEE/SICE International Symposium on System Integration (SII)* (pp. 497–502).
- Zouaghi, L., Alexopoulos, A., Wagner, A., & Badreddin, E. (2011b). Probabilistic online-generated monitoring models for mobile robot navigation using modified Petri net. In *15th International Conference on Advanced Robotics (ICAR)* (pp. 594–599).

APPENDIX

This section provides the three-tank system model. Tables 4 and 5 detail state equations and degradation laws associated with the system health modes.

Table 4. Three-tank system example: state equations and degradation laws.

	Valve state	State equations	Degradation laws for leak faults
nom_1	$v_{13} = open$ $v_{32} = open$	$D_1 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{ h_1(t) - h_3(t) } - K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
nom_2 $12f_4$	$v_{13} = closed$ $v_{32} = open$	$D_2 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
nom_3 $34f_4$ $23f_5$	$v_{13} = closed$ $v_{32} = closed$	$D_3 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{1}{S_2}(-Q_{20}(t)) \\ \dot{h}_3(t) = 0 \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
nom_4 $14f_5$	$v_{13} = open$ $v_{32} = closed$	$D_4 \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{Q_{20}(t)}{S_2}K_{32} \\ \dot{h}_3(t) = \frac{K_{13}}{S_3}\sqrt{ h_1(t) - h_3(t) } \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$
$1f_1$	$v_{13} = open$ $v_{32} = open$	$D_{1f_1} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) } - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{ h_1(t) - h_3(t) } - K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$2f_1$	$v_{13} = closed$ $v_{32} = open$	$D_{2f_1} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) - K_1\sqrt{h_1(t)} \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$3f_1$	$v_{13} = closed$ $v_{32} = closed$	$D_{3f_1} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) - K_1\sqrt{h_1(t)} \\ \dot{h}_2(t) = \frac{1}{S_2}(-Q_{20}(t)) \\ \dot{h}_3(t) = 0 \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
$4f_1$	$v_{13} = open$ $v_{32} = closed$	$D_{4f_1} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) } - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{Q_{20}(t)}{S_2}K_{32} \\ \dot{h}_3(t) = \frac{K_{13}}{S_3}\sqrt{ h_1(t) - h_3(t) } \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$
$1f_2$	$v_{13} = open$ $v_{32} = open$	$D_{1f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{ h_1(t) - h_3(t) } - K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$2f_2$ $12f_2f_4$	$v_{13} = closed$ $v_{32} = open$	$D_{2f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{32}\sqrt{ h_3(t) - h_2(t) }) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$3f_2$ $34f_2f_4$ $23f_3f_5$	$v_{13} = closed$ $v_{32} = closed$	$D_{3f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(-Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = 0 \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
$4f_2$ $14f_2f_5$	$v_{13} = open$ $v_{32} = closed$	$D_{4f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{1}{S_2}(Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{K_{13}}{S_3}\sqrt{ h_1(t) - h_3(t) } \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$
$1f_3$	$v_{13} = open$ $v_{32} = open$	$D_{1f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{ h_1(t) - h_3(t) } - K_{32}\sqrt{ h_3(t) - h_2(t) } - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$2f_3$ $12f_3f_4$	$v_{13} = closed$ $v_{32} = open$	$D_{2f_3} \begin{cases} \dot{h}_1(t) = \frac{Q_1(t)}{S_1} \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{ h_3(t) - h_2(t) } - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(-K_{32}\sqrt{ h_3(t) - h_2(t) } - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$3f_3$ $34f_3f_4$ $23f_3f_5$	$v_{13} = closed$ $v_{32} = closed$	$D_{3f_3} \begin{cases} \dot{h}_1(t) = \frac{Q_1(t)}{S_1} \\ \dot{h}_2(t) = \frac{1}{S_2}(-Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(-K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
$4f_3$ $14f_3f_5$	$v_{13} = open$ $v_{32} = closed$	$D_{4f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{ h_1(t) - h_3(t) }) \\ \dot{h}_2(t) = \frac{Q_{20}(t)}{S_2} \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{ h_1(t) - h_3(t) } - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$

Table 5. Three-tank system example: state equations and degradation laws.

	Valve state	State equations	Degradation laws for leak faults
$1f_1f_2$	$v_{13} = \text{open}$ $v_{32} = \text{open}$	$D_{1f_1f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{h_1(t) - h_3(t)} - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{h_3(t) - h_2(t)} - Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{h_1(t) - h_3(t)} - K_{32}\sqrt{h_3(t) - h_2(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$2f_1f_2$	$v_{13} = \text{closed}$ $v_{32} = \text{open}$	$D_{2f_1f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{h_3(t) - h_2(t)} - Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{-K_{32}}{S_3}\sqrt{h_3(t) - h_2(t)} \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$3f_1f_2$	$v_{13} = \text{closed}$ $v_{32} = \text{closed}$	$D_{3f_1f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(-Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = 0 \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
$4f_1f_2$	$v_{13} = \text{open}$ $v_{32} = \text{closed}$	$D_{4f_1f_2} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{h_1(t) - h_3(t)} - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(Q_{20}(t)K_{32} - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{K_{13}}{S_3}\sqrt{h_1(t) - h_3(t)} \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$
$1f_2f_3$	$v_{13} = \text{open}$ $v_{32} = \text{open}$	$D_{1f_2f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{h_1(t) - h_3(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{h_3(t) - h_2(t)} - Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{h_1(t) - h_3(t)} - K_{32}\sqrt{h_3(t) - h_2(t)} - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$2f_2f_3$	$v_{13} = \text{closed}$ $v_{32} = \text{open}$	$D_{2f_2f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{h_3(t) - h_2(t)} - Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(-K_{32}\sqrt{h_3(t) - h_2(t)} - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$3f_2f_3$	$v_{13} = \text{closed}$ $v_{32} = \text{closed}$	$D_{3f_2f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) \\ \dot{h}_2(t) = \frac{1}{S_2}(-Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
$4f_2f_3$	$v_{13} = \text{open}$ $v_{32} = \text{closed}$	$D_{4f_2f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{h_1(t) - h_3(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(Q_{20}(t) - K_2\sqrt{h_2(t)}) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{h_1(t) - h_3(t)} - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$
$1f_1f_3$	$v_{13} = \text{open}$ $v_{32} = \text{open}$	$D_{1f_1f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{h_1(t) - h_3(t)} - K_1\sqrt{h_3(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{h_3(t) - h_2(t)} - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{h_1(t) - h_3(t)} - K_{32}\sqrt{h_3(t) - h_2(t)} - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$2f_1f_3$	$v_{13} = \text{closed}$ $v_{32} = \text{open}$	$D_{2f_1f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{1}{S_2}(K_{32}\sqrt{h_3(t) - h_2(t)} - Q_{20}(t)) \\ \dot{h}_3(t) = \frac{1}{S_3}(-K_{32}\sqrt{h_3(t) - h_2(t)} - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_a)$
$3f_1f_3$	$v_{13} = \text{closed}$ $v_{32} = \text{closed}$	$D_{3f_1f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}Q_1(t) - K_1\sqrt{h_1(t)} \\ \dot{h}_2(t) = \frac{-Q_{20}(t)}{S_2} \\ \dot{h}_3(t) = \frac{1}{S_3}(-K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_c)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_b)$
$4f_1f_3$	$v_{13} = \text{open}$ $v_{32} = \text{closed}$	$D_{4f_1f_3} \begin{cases} \dot{h}_1(t) = \frac{1}{S_1}(Q_1(t) - K_{13}\sqrt{h_1(t) - h_3(t)} - K_1\sqrt{h_1(t)}) \\ \dot{h}_2(t) = \frac{Q_{20}(t)}{S_2} \\ \dot{h}_3(t) = \frac{1}{S_3}(K_{13}\sqrt{h_1(t) - h_3(t)} - K_3\sqrt{h_3(t)}) \end{cases}$	$f_1(t) = W(t, \eta_a)$ $f_2(t) = W(t, \eta_a)$ $f_3(t) = W(t, \eta_c)$