



HAL
open science

Approche intégrée de diagnostic et de pronostic pour la gestion de santé des systèmes hybrides sous incertitude

Quentin Gaudel

► **To cite this version:**

Quentin Gaudel. Approche intégrée de diagnostic et de pronostic pour la gestion de santé des systèmes hybrides sous incertitude. Automatique. INSA de Toulouse, 2016. Français. NNT : 2016ISAT0019 . tel-01417300v2

HAL Id: tel-01417300

<https://laas.hal.science/tel-01417300v2>

Submitted on 8 Jun 2017

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.

Université Fédérale



Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*

Présentée et soutenue le 30/9/16 par :

QUENTIN GAUDEL

Approche intégrée de diagnostic et de pronostic pour la gestion de santé des systèmes hybrides sous incertitude

JURY

ELODIE CHANTHERY
ALESSANDRO GIUA
RAFAEL GOURIVEAU
DIMITRI LEFEBVRE
PAULINE RIBOT
JAYANT SEN GUPTA
FRANÇOIS VERNADAT

Directrice
Rapporteur
Rapporteur
Examinateur
Directrice
Invité
Président

MCF - INSA de Toulouse
Professeur - Université AMU
MCF HDR - ENSMM Besançon
Professeur - Université du Havre
MCF - Université Toulouse 3
Ingénieur de Recherche - Airbus
Professeur - INSA de Toulouse

École doctorale et spécialité :

EDSYS : Informatique 4200018

Unité de Recherche :

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)

Équipe Diagnostic, Supervision et Conduite (DISCO)

Directeur(s) de Thèse :

Elodie CHANTHERY et Pauline RIBOT

Rapporteurs :

Alessandro GIUA et Rafael GOURIVEAU

Plan

Plan	2
Acronymes & Notations	7
Introduction	11
1 État de l'art	19
1.1 Concepts généraux	19
1.2 Architectures de supervision de système	22
1.2.1 Architectures orientées-mission	23
1.2.2 Architectures orientées-santé	26
1.3 Méthodes pour le diagnostic	28
1.3.1 Approches continues	30
1.3.2 Approches discrètes	33
1.3.3 Approches hybrides	37
1.3.4 Diagnostic avancé	38
1.4 Méthodes pour le pronostic	39
1.4.1 Approches continues	41
1.4.2 Approches discrètes	41
1.4.3 Approches hybrides	42
1.4.4 Fusion de méthodes de pronostic	43
1.5 Conclusion	43
2 Cadre de modélisation	45
2.1 Introduction	46
2.2 Exemple du robot mobile	46
2.3 Réseaux de Petri hybrides particuliers	48
2.4 Modèle de santé	56
2.5 Cas d'étude : les réservoirs d'eau	58
2.5.1 Description du système	58
2.5.2 Modélisation HPPN	62
2.6 Conclusion	67
3 Diagnostic	69
3.1 Introduction	70
3.2 Gestion des incertitudes	70
3.2.1 Incertitude symbolique	71

3.2.2	Incertitude numérique	72
3.3	Génération du diagnostiqueur	73
3.3.1	Copie du modèle	75
3.3.2	Séparation en niveau de comportement et niveau hybride	75
3.3.3	Modification des conditions symboliques	76
3.3.4	Suppression des conditions hybrides	78
3.3.5	Fusion des transitions	78
3.3.6	Suppression des transitions bouclant sur une même place hybride	80
3.4	Processus du diagnostiqueur	81
3.4.1	Prédiction	82
3.4.1.1	Pseudo-tirage des transitions	82
3.4.1.2	Mise à jour des valeurs des jetons	83
3.4.2	Correction	85
3.4.2.1	Calcul du score d'une hypothèse	86
3.4.2.2	Rééchantillonnage des jetons	90
3.5	Diagnostic	92
3.6	Algorithme stochastique de changement d'échelle	94
3.7	Gestion des performances calculatoires	96
3.7.1	Réduction de la complexité du réseau	98
3.7.2	Régulation du nombre total de jetons	99
3.8	Cas d'étude : les réservoirs d'eau	99
3.8.1	Diagnostiqueur HPPN	100
3.8.2	Résultats	102
3.8.2.1	Scénario 1	102
3.8.2.2	Scénario 2	106
3.8.2.3	Scénario 3	108
3.9	Conclusion	110
4	Pronostic	113
4.1	Introduction	114
4.2	Intégration des méthodes de diagnostic et de pronostic	115
4.3	Génération du pronostiqueur	115
4.3.1	Copie du modèle	116
4.3.2	Modification des conditions	117
4.3.3	Suppression des transitions non tirables	118
4.3.4	Suppression des bruits de l'évolution de l'état	118
4.4	Processus de pronostic	119
4.4.1	Initialisation du pronostiqueur	119
4.4.2	Processus du pronostiqueur	121
4.4.2.1	Mise à jour des valeurs des configurations avec l'entrée discrète	121
4.4.2.2	Tirage des transitions	122
4.4.2.3	Mise à jour des valeurs des jetons	123
4.5	Pronostic	124
4.6	Gestion des incertitudes	126
4.6.1	Connaissance de l'état courant	126
4.6.2	Évolution de l'état dans le futur	127
4.6.3	Connaissance des entrées futures	127
4.7	Cas d'étude : les réservoirs d'eau	128

4.7.1	Pronostiqueur HPPN	128
4.7.2	Résultats	130
4.7.2.1	Scénario 1	130
4.7.2.2	Scénario 2	133
4.7.2.3	Scénario 3	134
4.8	Vers une intégration avancée des méthodes de diagnostic et de pronostic	135
4.9	Conclusion	137
5	Cas d'étude : le rover K11	139
5.1	Introduction	140
5.2	Description	140
5.3	Modélisation du rover	143
5.4	Implémentation	144
5.5	Résultats	145
5.5.1	Scénario nominal	147
5.5.2	Scénario de faute sans replanification	150
5.5.3	Scénario de faute avec replanification	153
5.5.4	Analyse des performances	155
5.6	Conclusion	157
	Conclusions et Perspectives	161
	Liste des figures	165
	Liste des tables	171
	Références	173

Acronymes & Notations

Acronymes

ARR	Analytical Redundancy Relations,
CPN	Continuous Petri Nets,
DES	Discrete-Event System,
EOL	End Of Life,
EOM	End Of Mission,
EOP	End Of Prediction,
FDI	Fault Detection and Isolation,
FDIR	Fault Detection, Isolation and Recovery,
HA	Hybrid Automaton,
HBG	Hybrid Bond Graph,
HPN	Hybrid Petri Nets,
HPPN	Hybrid Particle Petri Nets,
IMM	Interactive Multiple-Model,
MPPN	Modified Particle Petri Nets,
PHM	Prognostics and Health Management,
POP	Partially Observed Petri Nets,
PPN	Particle Petri Nets,
RUL	Remaining Useful Life,
SIR	Sampling Importance Resampling,
SPN	Stochastic Petri Nets,
SSA	Stochastic Scaling Algorithm,
TOR	Tout ou Rien (type de capteurs).

Notations

HPPN

P	l'ensemble des places,
p	une place,
P^S	l'ensemble des places symboliques,
p^S	une place symbolique,
P^N	l'ensemble des places numériques,
p^N	une place numérique,
P^H	l'ensemble des places hybrides,
p^H	une place hybride,
T	l'ensemble des transitions,
t	une transition,
A	l'ensemble des arcs,
a	un arc,
\mathcal{A}	l'ensemble des annotations des arcs,
E	l'ensemble des labels d'événements,
v	un label d'événement,
f	un label de faute,
X	l'espace d'état du vecteur d'état continu,
D	l'espace d'état du vecteur d'état hybride,
\mathcal{C}	l'ensemble des dynamiques continues,
\mathcal{H}	l'ensemble des dynamiques hybrides,
Ω	l'ensemble des conditions,
Ω_t	l'ensemble des conditions associées à t ,
ω^S	une condition symbolique,
ω^N	une condition numérique,
ω^H	une condition hybride,
\mathbb{M}_k	le marquage au temps k ,
\mathbb{M}_k^S	le marquage symbolique au temps k ,
\mathbb{M}_k^N	le marquage numérique au temps k ,
\mathbb{M}_k^H	le marquage hybride au temps k ,
M_k	l'ensemble des jetons au temps k ,
$M_k(p)$	l'ensemble des jetons de p au temps k ,
M_k^S	l'ensemble des jetons symboliques au temps k ,
M_k^N	l'ensemble des jetons numériques au temps k ,
M_k^H	l'ensemble des jetons hybrides au temps k ,

${}^{\circ}t$	l'ensemble des places en entrée de t ,
t°	l'ensemble des places en sortie de t ,
δ_k	une configuration au temps k ,
b_k	un ensemble d'événements porté par une configuration au temps k ,
e	un événement,
π_k	un ensemble d'événements porté par une configuration au temps k ,
x_k	un vecteur d'état continu porté particule au temps k ,
h_k	un jeton hybride au temps k ,
d_k	un vecteur d'état hybride porté particule au temps k ,
S_k^t	l'ensemble des jetons acceptés par Ω_t au temps k ,
$S_k^t(p)$	l'ensemble des jetons de p acceptés par Ω_t au temps k .

Diagnosticteur HPPN

$HPPN_{\Phi}$	un modèle HPPN ,
$HPPN_{\Delta}$	un diagnosticteur HPPN,
Δ_k	le diagnostic au temps k ,
O_k	l'ensemble des observations au temps k ,
U_k^S	l'ensemble des entrées discrètes au temps k ,
u_k^N	le vecteur des entrées continues au temps k ,
Y_k^S	l'ensemble des sorties discrètes au temps k ,
y_k^N	le vecteur des sorties continues au temps k ,
O_k^{S-}	l'ensemble des événements observés jusqu'à k ,
ζ	la partition des clusters de particules,
ς	un cluster de particules.

Pronostiqueur HPPN

$HPPN_{\Pi}$	un pronostiqueur HPPN,
Π_k	le pronostic au temps k ,
U_k^+	l'ensemble des futures entrées au temps k ,
τ_p	l'horizon de prédiction.

Introduction

Les systèmes autonomes ou partiellement autonomes, tels que les aéronefs, satellites, drones, rovers, voitures, humanoïdes, etc., sont devenus si complexes qu'il devient humainement difficile d'anticiper leurs comportements, particulièrement lors de l'occurrence de pannes. Au delà de sa mission, il est donc essentiel que la conception et l'utilisation du système s'élargissent à la **gestion de la santé** (*system health management, Prognostics and Health Management* ou PHM). La gestion de santé d'un système vise au développement d'outils qui aident à la réduction des coûts liés aux indisponibilités et aux réparations, mais également à sa reconfiguration et à l'optimisation de sa mission. Par exemple, dans le cas d'un système de type rover dont la mission est d'explorer différentes zones afin d'y accomplir des expériences, la gestion de santé consiste à assurer la réalisation des expériences les plus importantes malgré un problème de déplacement (dysfonctionnement des moteurs) ou encore une diminution de l'énergie disponible (dysfonctionnement des panneaux solaires). Pour la gestion d'une flotte d'avions, la gestion de santé consiste à maximiser la disponibilité des avions tout en minimisant le coût des réparation, ce qui peut alors s'apparenter à la médecine préventive.

Une technique de **surveillance de la santé** (*health monitoring*) efficace doit être adoptée pour déterminer l'état du système à tout moment. Une méthode de **diagnostic** détermine l'état courant du système et identifie les causes probables (interaction avec l'environnement, fautes, etc.) qui peuvent conduire à cet état en raisonnant sur les observations. Une méthode de **pronostic** utilise le plan de mission courant et la connaissance sur la dégradation du système pour anticiper les comportements anormaux ou les fautes et ainsi prédire les futurs états. On associe généralement le pronostic à la prédiction de la **fin de vie** (*End Of Life* ou EOL) d'un système en service, date à laquelle le système n'est plus opérationnel, ou à sa **durée de vie résiduelle** (*Remaining Useful Life* ou RUL), durée restante jusqu'à la fin de vie. Le diagnostic et le pronostic permettent donc d'avoir un rapport sur la santé actuelle du système ainsi qu'une prédiction de l'évolution de son état dans le futur. Ces informations servent alors à reconfigurer le système et à mettre à jour les plans de mission et de maintenance, d'où l'importance de la qualité du diagnostic et du pronostic.

Intégration des méthodes de diagnostic et pronostic

Si le diagnostic et le pronostic du système semblent être corrélés, leurs calculs sont généralement dissociés, car les échelles de temps manipulées par ces deux processus sont très différentes. Pourtant, certains systèmes ont des missions à l'échelle de leur dégradation, comme par exemple les sondes spatiales, qui peuvent être soumises à des phénomènes physiques importants et qui sont souvent irréparables. Un rapprochement des méthodes semble propice à l'obtention de meilleurs résultats de surveillance de santé, et donc à de meilleures prises de décision autonomes du système. Pour respecter des contraintes de performances, il peut donc être pertinent, par exemple, de n'effectuer que des prédictions sur les états courants les plus probables déterminés par le diagnostic. De la même manière, les états pour lesquels le pronostic calcule la fin de vie la plus proche pourrait motiver la recherche d'un diagnostic plus précis ciblé.

Cette thèse a pour objectif principal d'étudier l'intérêt d'une **approche intégrée** des méthodes de diagnostic et de pronostic, et son impact sur les résultats de surveillance de santé de systèmes. Afin de répondre à cette problématique, une **architecture** de gestion de santé est définie pour assurer les missions des systèmes. Les systèmes étudiés dans ces travaux sont considérés comme des **systèmes hybrides**, de par leur complexité et l'hétérogénéité des observations. La surveillance de la santé du système est réalisée par une méthodologie basée sur les **modèles** et sur l'utilisation des observations en ligne. Un modèle est un système abstrait représentant imparfaitement le système concret. Pour appliquer la méthodologie à un cas réel, il est indispensable de considérer les **incertitudes** relatives à la modélisation, mais aussi les incertitudes relatives aux observations sur le système. Enfin, la méthodologie répond aussi à la gestion des **performances** de calcul, qui est intrinsèquement liée aux incertitudes.

Architecture de gestion de santé d'un système

Le terme "**système**", utilisé dans des nombreuses expressions telles que système informatique, système industriel, système solaire ou système endodontique, est défini dans ce manuscrit de la manière suivante.

Définition 1 (Système). Un système est une combinaison d'éléments qui interagissent entre eux et avec l'environnement, et qui est cohérente vis à vis d'un certain objectif.

Un système est décrit par sa structure et sa fonction, ainsi que son interaction avec son environnement, délimitée par des frontières spatiales et temporelles. Cette interaction peut se faire vers ou depuis ses frontières (liaisons orientées), et sous formes d'informations, d'énergie ou de matières transférées, en fonction de la discipline. Les **entrées** sont les liaisons orientées vers le système. Elles agissent sur le comportement du système et permettent de le commander. Les **sorties** sont les liaisons orientées vers l'environnement. Elles correspondent à des informations

renvoyées directement depuis le système. Un système industriel, par exemple, est généralement commandé via les entrées de ses actionneurs, tandis que les données de sortie sont renvoyées par des capteurs. Un actionneur réalise un travail physique à partir d'une énergie fournie. Un capteur mesure une grandeur physique et la transforme en grandeur utilisable.

Les entrées, comme les sorties, sont dites **continues** (ou quantitative) si elles peuvent prendre une infinité de valeurs. C'est le cas lorsqu'elles sont issues de capteurs ou d'actionneurs analogiques ou numériques. En revanche, elles sont dites **discrètes** (ou qualitative) si elles peuvent prendre un nombre fini de valeurs. C'est le cas pour les capteurs ou actionneurs *Tout Ou Rien* (TOR), qui peuvent prendre les valeurs 0 ou 1, ou bien si les données sont prétraitées (classification, symbolisation, abstraction de données continues). En général, les valeurs des sorties sont influencées par celles des entrées, mais dépendent aussi de la situation interne du système (l'**état** du système) qui est liée à ce qui s'est passé précédemment.

Lors de l'étude d'un système à des fins d'observations, de contrôle, d'extraction d'information, etc., il est commun de définir une architecture afin de mettre en perspective l'objectif des travaux.

L'architecture de gestion (ou supervision) de santé proposée dans ce manuscrit pour l'intégration du diagnostic et du pronostic est présentée en Figure 1. Un module de décision permet la

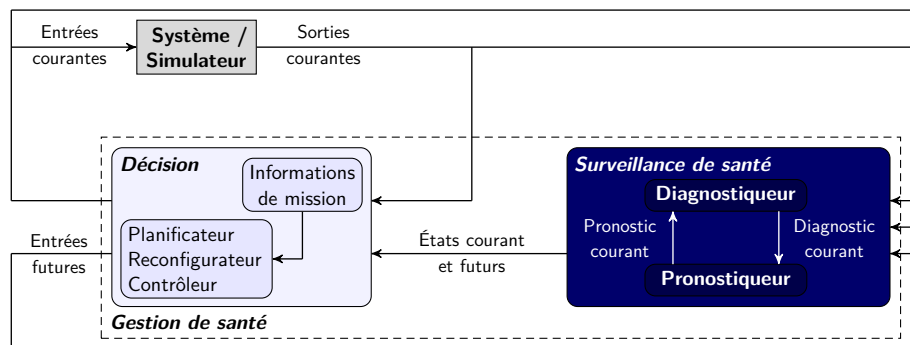


FIGURE 1 – Architecture de gestion de santé de système intégrant diagnostic et pronostic.

replanification de la mission du système, sa reconfiguration et/ou encore son contrôle de bas niveau. Un module de surveillance inclut les fonctions de diagnostic et de pronostic. En plus des données de sortie du système, chacun des deux modules de décision et de surveillance de santé utilise les données calculées par l'autre. Le **diagnostiqueur** (*diagnoser*) est le module qui calcule le diagnostic du système. Le **pronostiqueur** (*prognoser*) est le module qui calcule le pronostic du système. Le diagnostiqueur utilise le pronostic et le pronostiqueur utilise le diagnostic, illustrant l'interaction des deux modules. L'ensemble des entrées et sorties du système, discrètes et continues, est un ensemble d'**observations** du point de vue du module de surveillance de santé et sera désigné comme tel dans ce manuscrit.

Systemes hybrides

Un système est dit "hybride" si sa description fait intervenir explicitement et simultanément des dynamiques de type continu (phénomènes ou comportements continus) et discret (événementiel).

L'évolution d'un système hybride est généralement décrit par un **système multimode** (*multimode system*), pour lequel chaque mode est associé à une dynamique continue (modèle continu) qui représente un fonctionnement du système sous des conditions opérationnelles particulières. L'**état continu** du système évolue selon la dynamique continue associée au mode courant du système. Les changements de modes sont décrits par un système à événements discrets (*Discrete-Event System* ou DES). Un changement de modes est marqué par l'occurrence d'un **événement** interne ou externe au système (faute, événement de mission, interaction avec l'environnement, condition sur l'état continu, etc.). L'**état discret** du système est l'état courant du DES.

Un tel système de modes est utile pour décrire un système ayant une dynamique continue particulière pour chaque mode de fonctionnement (dynamique continue par morceau). Il n'est en revanche pas suffisant pour représenter une dynamique intrinsèquement hybride, c'est à dire une dynamique dépendante de l'état continu et d'informations discrètes, comme par exemple l'ensemble des événements survenus dans le système. Ce genre de dynamique est utile pour modéliser certains phénomènes, comme la dégradation ou la stabilité d'un système. Par exemple, pour la plupart des systèmes industriels, lorsque la dégradation n'est pas observable, elle est estimée avec des probabilités d'occurrences de fautes. Ces probabilités peuvent évoluer différemment dans chaque mode du système (on parle du niveau de stress d'un mode), mais elles peuvent également dépendre de l'analyse des fautes survenues précédemment (un enchaînement de certaines fautes peut avoir un effet catalyseur, par exemple). Pour pouvoir exprimer ces dépendances, on définit dans ces travaux une telle dynamique comme hybride. La valeur courante de cette **dynamique hybride** est appelée **état hybride**. Dans l'objectif de pouvoir modéliser un système aussi précisément que possible, le système de modes est ainsi étendu en associant une dynamique hybride à chaque mode.

Important : Dans ces travaux, un **mode** est donc la combinaison d'un état du DES, d'une dynamique continue et d'une dynamique hybride. L'**état** du système est la combinaison de ses états discret, continu et hybride.

Dans le cadre de la gestion de la santé, on s'intéresse à la modélisation des changements de dynamiques du système lorsqu'une ou plusieurs fautes surviennent. Un mode décrit l'évolution du système dans des conditions de santé spécifiques. Tant qu'aucune faute n'est survenue, le système est dans un **mode nominal** (*nominal mode*). Les fautes sont supposées permanentes, c'est-à-dire que lorsqu'une faute survient, le système entre dans un **mode dégradé** (*degraded*

mode ou *faulty mode*), et ne reviendra plus dans un mode nominal sans réparation. Il peut ainsi finir dans un **mode de défaillance** (*failure mode*), dans lequel le système n'est plus opérationnel. Cette évolution est illustrée dans la Figure 2.

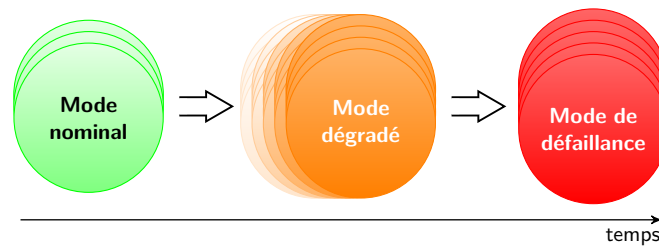


FIGURE 2 – Évolution unidirectionnelle d'un système sans maintenance ou action de réparation.

À travers le suivi de l'état du système hybride, la méthode de diagnostic détermine le mode courant du système et le chemin emprunté (**trajectoire passée**), dont la séquence de modes et l'ensemble des fautes ayant eu lieu, pour l'atteindre. En parallèle, la méthode de pronostic prédit les **trajectoires futures** possibles et particulièrement la date d'entrée dans un mode de défaillance (EOL).

Pour répondre à cette problématique, les **réseaux de Petri hybrides particuliers** (*Hybrid Particle Petri Nets* ou **HPPN**) ont été développés dans ces travaux de thèse pour capturer toutes les informations connues sur le système. Les HPPN sont un formalisme mathématique qui manipule des variables discrètes et continues, adéquates au cadre des systèmes hybrides, via l'utilisation de jetons. Dans ces travaux, le système étudié est modélisé par un HPPN. De plus, le diagnostiqueur et le pronostiqueur sont aussi des HPPN, assurant ainsi une **interface** cohérente et indispensable pour approfondir leur intégration.

Incertitudes et performances

Les dernières problématiques abordées par ces travaux sont la gestion des incertitudes et des performances de la méthode, deux aspects essentiels pour une utilisation sur des systèmes réels.

La gestion des incertitudes est abordée sous plusieurs angles. L'étape de modélisation avec les HPPN permet la spécification des incertitudes sur la connaissance du système, ainsi que sur les observations, qui peuvent être fausses, manquantes ou imprécises. Le processus du diagnostiqueur HPPN exploite les incertitudes spécifiées dans le modèle avec l'utilisation de **filtres particuliers**, mais considère aussi son écart avec la réalité. Il retourne le diagnostic du système sous forme de distribution de croyances sur les trajectoires passées du système et représenté par son marquage (i.e. l'ensemble de ses jetons). Le processus du pronostiqueur HPPN utilise ce diagnostic pour simuler l'évolution du système dans le futur et retourner le

pronostic, lui aussi sous la forme d'un marquage représentant une distribution de croyances sur les trajectoires futures du système.

Enfin, les performances des méthodes de diagnostic et de pronostic reposent principalement sur les ressources de calcul qui leur sont allouées. Un **algorithme stochastique de changement d'échelle** (*Stochastic Scaling Algorithm* ou **SSA**) est utilisé à différentes étapes de la méthodologie de surveillance de santé. Il a été conçu pour distribuer des ressources de calcul en fonction de l'importance des tâches à réaliser, tout en instaurant un compromis entre la qualité des résultats, le respect des contraintes de temps de calcul et les capacités du calculateur.

Ce manuscrit est organisé comme suit.

Le **Chapitre 1** introduit ces travaux avec un état de l'art des architectures utilisées pour la supervision des systèmes complexes, et présente les formalismes et méthodes utilisés pour la surveillance de systèmes, incluant diagnostic et pronostic, à base de modèles.

Contributions théoriques

Les Chapitres 2, 3 et 4 exposent les trois contributions théoriques majeures de ces travaux.

Le **Chapitre 2** introduit et détaille le formalisme des **réseaux de Petri hybrides particuliers**. Les réseaux de Petri, usuellement utilisés pour représenter des dynamiques discrètes, sont ici enrichis par des dynamiques continues et hybrides. Leur utilisation pour la modélisation des systèmes hybrides est ensuite décrite, puis illustrée sur un exemple simple de robot mobile, qui est réutilisé tout au long des Chapitres 2, 3, et 4.

Le **Chapitre 3** présente le **diagnostiqueur**, un objet généré à partir du modèle HPPN du système et qui est également un HPPN. Le diagnostiqueur HPPN surveille sous incertitude à la fois le comportement du système et sa dégradation. Son processus utilise à chaque instant les observations sur le système pour faire évoluer son marquage (ses jetons) sur le principe de prédiction et de correction des filtres particuliers, combiné à l'algorithme SSA. Le diagnostic courant du système est le marquage courant du diagnostiqueur, qui porte une distribution de croyances sur les trajectoires passées du système. Ce chapitre introduit et détaille également l'algorithme SSA.

Le **Chapitre 4** propose d'abord une **méthodologie de surveillance de santé intégrant diagnostic et pronostic**, et en détaille leur interface. Ensuite, le **pronostiqueur** HPPN est présenté. Il s'agit d'un HPPN, généré lui aussi avec le modèle HPPN du système, qui prédit les futures trajectoires du système en fonction des entrées correspondantes au plan de mission. La prédiction s'arrête lorsque toutes les entrées ont été simulées ou avant si toutes les trajectoires ont atteint un mode de défaillance, dans le but de réduire les temps de calcul. Le marquage du pronostiqueur à la fin de la prédiction est le pronostic du système. Celui-ci porte une distribution

de croyances sur les trajectoires futures du système, et particulièrement sur sa date d'entrée dans un mode de défaillance (EOL).

Contributions expérimentales

Pour exposer les avantages de la méthode, ce manuscrit fait rapport de contributions expérimentales. Deux systèmes sont étudiés, le système des réservoirs d'eau et le rover planétaire K11, dont les résultats en situations complexes sont donnés, expliqués et argumentés.

Le système des réservoirs d'eau est un **cas d'étude simulé** qui illustre en profondeur, à la fin des Chapitres 2, 3 et 4, les concepts théoriques présentés. Il met en valeur les avantages de la modélisation à base de HPPN, ainsi que les générations des diagnostiqueur et pronostiqueur. Les résultats des méthodes de diagnostic et pronostic sont aussi présentés à travers différents scénarios.

Le rover planétaire est un **cas d'étude réel** qui met en avant l'applicabilité de la méthode et sa robustesse face aux données réelles. Il est étudié dans le **Chapitre 5**. C'est un rover à quatre roues conçu pour tester les architectures économes en énergie, dans les conditions de l'Antarctique. Sa conception a ensuite été repensée par le NASA Ames Research Center à des fins de diagnostic et de pronostic. Il s'agit maintenant d'un banc d'essai sur lequel il est possible de provoquer (ou simuler) des fautes et des défaillances. Dans ces travaux, le K11 est étudié comme un système en opération exposé à des défaillances et réalisant des missions. Les entrées du rover sont ses commandes continues et ses sorties sont les mesures continues des capteurs. Un modèle hybride du K11 est proposé avec les HPPN, puis la méthodologie de surveillance de santé est testée sur des scénarios réels, mettant en avant l'utilisabilité des résultats.

Chapitre 1

État de l'art

Résumé

Dans ce chapitre, les concepts de diagnostic et de pronostic de systèmes sont définis. Dans le cadre de la gestion de santé des systèmes, on souligne l'intérêt d'une part d'intégrer des méthodes de diagnostic et de pronostic pour améliorer la qualité de la surveillance du système et d'autre part de les rapprocher des problématiques de contrôle et de planification. Une étude des architectures de systèmes existantes montre que ces deux points n'ont pas encore été intégrés dans une même architecture.

Concernant les techniques de diagnostic et pronostic, on verra que les formalismes à base de modèles hybrides sont les plus appropriés pour représenter les connaissances sur l'évolution de la santé des systèmes complexes.

Le processus de pronostic étant naturellement incertain, on s'intéressera particulièrement aux techniques de diagnostic qui tiennent compte des incertitudes liées au modèle et aux observations du système, tant au niveau discret qu'au niveau continu.

1.1 Concepts généraux

Ces travaux concernent la gestion de santé des systèmes, et s'orientent particulièrement vers l'intégration des modules de diagnostic et de pronostic.

La fonction de diagnostic sert à déterminer l'état de santé courant du système. Dans les travaux de [Zwingelstein 95], le diagnostic est défini de la manière suivante.

Définition 2 (Diagnostic). Le diagnostic est l'identification de la cause probable de la (ou des) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test.

Cette définition repose principalement sur le terme "défaillance". Nous choisissons la définition suivante afin de le caractériser.

Définition 3 (Défaillance). Une défaillance correspond à une cessation de l'aptitude d'une entité à accomplir une ou plusieurs fonctions requises.

On peut alors distinguer deux formes de défaillances : les défaillances partielles et les défaillances complètes. Une défaillance partielle correspond à une dégradation de l'aptitude d'un système à accomplir un certain nombre de fonctions requises. Si le système n'est plus capable de réaliser les fonctions pour lesquelles il a été conçu, on parle de défaillance complète et on dit que le système est défaillant ou en panne. Dans [Villemeur 88], une panne est définie par l'inaptitude d'une entité à accomplir une fonction requise.

Une défaillance est générée par une faute. Le concept de faute est défini dans [Isermann 97].

Définition 4 (Faute). Une faute représente une déviation non acceptable d'au moins une propriété caractéristique ou d'un paramètre du système.

Les défaillances sont généralement surveillées à l'aide de grandeurs quantitatives appelées des indicateurs. Lorsque que ces indicateurs révèlent un comportement anormal, ils deviennent des symptômes. Les symptômes traduisent les effets observables des défaillances.

Définition 5 (Symptôme). Un symptôme est l'effet ou la conséquence visible d'une défaillance.

La Définition 2 met ainsi en évidence les deux tâches du diagnostic : l'observation des symptômes de la (ou des) défaillance(s) et l'identification de la (ou des) faute(s) qui en sont à l'origine à l'aide d'un raisonnement logique fondé sur des observations sur le système.

Les comportements anormaux ou les fautes sont anticipés par un raisonnement de pronostic sur le système. Dans la littérature, on retrouve principalement deux définitions du pronostic [Brotherton *et al.* 00, Goh *et al.* 06].

Définition 6. Le pronostic consiste à calculer une prédiction de l'état d'un composant ou d'un système dans le futur.

Définition 7. Le pronostic est la capacité de prédire la durée de vie résiduelle (RUL pour *Remaining Useful Life*) de composants ou systèmes en service.

La définition 6 est très générale. Elle assimile le pronostic à une fonction de prédiction des états futurs possibles du système. La définition 7 souligne les objectifs applicatifs de la fonction de pronostic.

La durée de vie résiduelle (RUL) d'un système correspond au temps restant avant que le système ne puisse plus réaliser avec succès ses fonctions requises et doive être remplacé [Engel *et al.* 00]. La Définition 7 met en évidence la notion temporelle liée à la prédiction. Cependant, le RUL peut également s'exprimer en tant que distance (kilomètres parcourus par un véhicule) ou en nombre d'utilisations (nombre de missions pour les navettes spatiales), par exemple. Les besoins sous-jacents sont de savoir si le système peut terminer sa mission sans rencontrer de défaillance, ou encore dans un objectif de maintenance, de déterminer la meilleure date pour réparer le système. De la définition de RUL découle la notion de date de fin de vie (EOL pour *End Of Life*), qui correspond à la date à laquelle le système n'est plus opérationnel. Le RUL propose donc une date relative à la date courante alors que la date de fin de vie est une date absolue.

Deux critères de performances découlent des méthodes de diagnostic et de pronostic de systèmes : la diagnosticabilité et la pronosticabilité. La diagnosticabilité est une mesure de la capacité d'une méthode de diagnostic à diagnostiquer les fautes qui apparaissent dans un système à partir des observations disponibles sur le système. Cette propriété détermine particulièrement les fautes que l'on peut discriminer avec les observations disponibles sur le système [Sampath *et al.* 95].

Définition 8 (Diagnosticabilité). La diagnosticabilité est la capacité d'un système et de ses fonctions de surveillance et de diagnostic à exhiber dans un délai fini des observations différentes pour chaque situation de faute anticipée.

La propriété de pronosticabilité détermine les fautes et les défaillances qu'il est possible de prédire avec la connaissance disponible sur le système [Ribot 09].

Définition 9 (Pronosticabilité). La pronosticabilité est la capacité d'un système et de ses fonctions de surveillance et de pronostic à prédire dans un délai fini la date d'une défaillance anticipée.

La suite de cet état de l'art vise premièrement à étudier les architectures de gestion (ou de supervision) de santé existantes de façon à définir une architecture qui interface de manière cohérente les modules de décision et de surveillance de santé. On s'attache ensuite à étudier les différents formalismes existants pour la surveillance de santé afin de modéliser le comportement du système et sa dégradation pour effectuer du diagnostic et du pronostic.

1.2 Architectures de supervision de système

Une architecture représente la décomposition structurelle d'une tâche à effectuer. Dans la plupart des travaux passés, cette décomposition est propre à l'application (robotique, automobile, avionique, spatiale, etc.). Ces architectures correspondent à la conception d'un système dont l'objectif est la réalisation d'une mission de manière la plus autonome possible. Une mission, généralement commanditée par un opérateur, est effectuée en réalisant des actions [Chanthery 05] : des actions de mouvements, des actions sur l'environnement, etc. Le plan de la mission (ou plan d'actions) est l'ensemble ordonné des actions à réaliser. La planification est l'opération par laquelle le plan est déterminé. La notion d'autonomie est définie de plusieurs manières dans la littérature [Antsaklis & Passino 89, Alami *et al.* 98, Grabowski 15] et de nouvelles définitions apparaissent régulièrement pour pouvoir qualifier précisément l'autonomie des systèmes récents en fonction des besoins. C'est le cas par exemple dans le contexte actuel des voitures autonomes [voiture-autonome.net 15] ou encore des aéronefs autonomes [U.S Air Force 15]. De manière générale, deux concepts cohabitent derrière le terme "autonomie" : la réalisation d'une mission sans intervention humaine et la faculté d'adaptation face aux perturbations de son environnement.

L'intérêt de l'intégration d'un module de surveillance de santé (principalement diagnostic) apparaît peu à peu après les premières utilisations des systèmes avec des problématiques de maintenance et de sécurité. La notion de criticité apparaît.

La criticité d'un système fait référence aux conséquences de son dysfonctionnement. Nous reprenons la définition de [Sprauel 16], qui qualifie un système de critique lorsque son dysfonctionnement peut avoir un impact humain, environnemental ou économique, conséquent.

Définition 10 (Système critique). Un système critique est un système dont le dysfonctionnement a un impact dramatique, allant de pertes coûteuses jusqu'à des risques pour des humains.

Ce n'est qu'il y a quelques années que des travaux proposent des architectures de gestion de santé (*Prognostic and Health Management* ou PHM). De part la complexité croissante des systèmes, ces architectures intègrent dès la phase de conception les fonctions de diagnostic et de pronostic. Les objectifs vont de l'augmentation de l'autonomie d'un système (automatique, intelligence artificielle) à l'aide à la décision pour l'opérateur de maintenance, en passant par la réduction des coûts de maintenance, etc.

La Table 1.1 résume les positions des principales architectures étudiées concernant l'intégration des fonctions de diagnostic, de pronostic et de gestion de mission. Dans ce tableau, les flèches représentent les liens entre les différentes fonctions. Par exemple, la flèche \longleftrightarrow dans la colonne Mission-Diagnostic indique que le modèle du diagnostic utilise les informations relatives à la mission et que le module de planification utilise les informations issues du diagnostic.

On constate que dans tous les travaux, le modèle du diagnostic utilise les informations, sous la forme de données capteurs ou données traitées par exemple, relatives à la mission ou au contrôle du système. En revanche, l'utilisation du diagnostic pour la planification ou le contrôle n'est pas toujours mis en place dans les architectures. Concernant le couplage Mission-Pronostic, la majorité des travaux utilisent des données relatives à la mission pour effectuer le pronostic. En revanche, seul [Narasimhan *et al.* 12] effectue un couplage du pronostic vers la mission. Enfin, concernant le couplage Diagnostic-Pronostic, la majorité des travaux utilisent des informations du diagnostic pour effectuer un pronostic. Seuls les travaux de [Chanthery & Ribot 13] effectuent un couplage du pronostic vers le diagnostic.

Nous distinguerons par la suite les architectures de gestion de mission (ou orientées-mission), élaborées dans l'objectif la réalisation de la mission du système, des architectures de gestion de santé (ou orientées-santé), élaborées plus récemment pour répondre à des problématiques de type PHM.

1.2.1 Architectures orientées-mission

Historiquement, les premières architectures avaient pour objectif la réalisation d'une mission en totale autonomie, ou bien de donner au système un niveau d'autonomie plus élevé.

La plupart des architectures sont décomposées en différents niveaux hiérarchiques. La Figure 1.1 propose une architecture hiérarchique générique sur trois niveaux. Elle sert ici de référence pour mettre en avant les différences entre les architectures étudiées. Le niveau le plus élevé est le

TABLE 1.1 – Position des principales architectures concernant l'intégration de diagnostic ou de pronostic à la supervision de la mission.

Travaux	Mission - Diagnostic	Mission - Pronostic	Diagnostic - Pronostic
[Antsaklis & Passino 89]	↔		
[Sampath <i>et al.</i> 96]	↔		
[Alami <i>et al.</i> 98]	→		
[Muscettola <i>et al.</i> 98]	↔		
[Berenjii & Wang 06]	→	→	→
[Hamilton <i>et al.</i> 07]	↔		
[Camci <i>et al.</i> 07]	→	→	
[Benedettini <i>et al.</i> 08]	→		→
[Zhang <i>et al.</i> 09]	→	→	→
[Ribot 09]	→	→	→
[Olive <i>et al.</i> 11]	↔		
[Roychoudhury & Daigle 11]	→		→
[Narasimhan <i>et al.</i> 12]	↔	↔	→
[Ribot <i>et al.</i> 13]	→	→	→
[Chanthery & Ribot 13]	→	→	↔
[Daigle <i>et al.</i> 15a]		→	

niveau décisionnel, dans lequel se trouve généralement le planificateur, qui génère le plan des actions à réaliser. Ce niveau nécessite pour cela une connaissance globale de l'objectif et du contexte de la mission. Le niveau exécutif est l'interface entre le niveau décisionnel et le niveau fonctionnel. Son rôle est d'exécuter le plan de mission en utilisant les différentes fonctions disponibles sur le système (génération de trajectoire, contournement d'obstacles, etc). Enfin, le niveau fonctionnel est l'interface avec le système.

Dans le domaine de la robotique, l'architecture 3T, introduite dans [Bonasso *et al.* 95], est une architecture hiérarchique à trois niveaux, semblable à l'architecture générique présentée sur la Figure 1.1. Elle se compose d'un système délibératif, d'un système séquenceur, et d'un ensemble de compétences. Le système délibératif gère la planification de mission du robot et génère le plan des actions à réaliser. Le séquenceur instancie les actions à exécuter pour réaliser le plan envoyé par le système délibératif. Les actions sont exécutées par les compétences qui sont interfacées directement avec les capteurs et les actionneurs du robot. Aucune fonction de diagnostic (encore moins de pronostic) n'est présente. Aucune notion de panne ou de défaillance n'apparaît.

Dans le même domaine, l'architecture LAAS proposé dans [Alami *et al.* 98] est très similaire à l'architecture 3T. Elle est également définie sur trois niveaux : un niveau décisionnel, une niveau exécutif, et un niveau fonctionnel. Le niveau décisionnel gère les fonctionnalités de haut niveau. Le niveau exécutif a pour objectif d'exécuter le plan d'actions à travers l'exécution de différentes fonctions réalisées par chaque module du niveau fonctionnel. Les modules, organisés en réseau

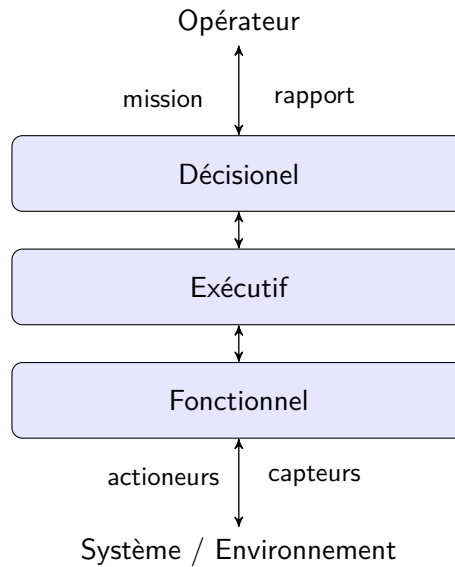


FIGURE 1.1 – Une architecture hiérarchique générique à trois niveaux : fonctionnel, exécutif et décisionnel.

dans le niveau fonctionnel, exécutent les fonctionnalités de bas niveaux. Outre l'utilisation de termes différents pour désigner les niveaux, l'architecture introduit la notion de défaillance. Une défaillance correspond à l'échec de la réalisation de la fonction d'un module. Si un module est défaillant, il est simplement arrêté en attendant une intervention extérieure. La prise de décision vis-à-vis de ces défaillances n'est pas abordée et la recherche d'un diagnostic plus précis n'est pas incluse.

Dans le domaine spatial, les auteurs de [Mussettola *et al.* 98] proposent l'architecture Remote Agent. En plus des fonctionnalités déjà introduites dans les architectures existantes, elle intègre un module dédié au diagnostic du système. L'approche considérée introduit des modes de fonctionnement (mode nominal, mode de faute, etc.). Le module de diagnostic consiste donc à déterminer le mode de fonctionnement courant du système et à effectuer sa reconfiguration si nécessaire. La reconfiguration est une réorganisation logicielle ou matérielle du système. En fonction du mode du système, différents modules de bas niveau vont être utilisés pour réaliser une tâche particulière, ou bien la tâche sera réalisée par le même module, mais de manière différente. Dans tous les cas, ce résultat de diagnostic n'est pas considéré par le planificateur. Il n'y a replanification des objectifs de la mission.

Les auteurs de [Olive *et al.* 11] proposent une architecture très similaire à l'architecture LAAS qui intègre des modules FDIR (*Fault Detection, Isolation and Recovery*) dans chacun des modules fonctionnels et également au niveau opérationnel.

L'approche FDIR est très utilisée dans le domaine spatial. Elle repose sur cinq niveaux hiérarchiques représentatifs des niveaux de criticité progressifs récapitulés dans la Table 1.2. Les

TABLE 1.2 – Description des niveaux de criticité de la stratégie FDIR.

Niveau	Description	Gestion autonome
0	Défaillance sans effet sur les performances	Oui, avec des mécanismes logiciels
1	Défaillance d'une unité de calcul ou dégradation des performances	Oui, avec la reconfiguration
2 & 3	Défaillance d'un équipement utilisé par FDIR ou défaillance non couverte par les niveaux précédents	Oui
4	Alarme matérielle ou défaillances multiples de niveau 2 ou 3	Non, passage en mode de survie (safe mode) et attente d'intervention

quatre premiers niveaux de criticité (niveaux 0 à 3) sont gérés par le système de manière autonome. Lorsque le système entre dans le dernier niveau (niveau 4), un opérateur doit prendre la main.

Bien que la stratégie FDIR soit rapide et efficace en terme d'isolation de faute, la volonté des auteurs de [Olive *et al.* 11] est de repousser la limite du dernier niveau pour rendre le système encore plus autonome en proposant une nouvelle architecture générique. Les auteurs proposent une solution basée sur deux aspects : la mise en place d'une architecture de décision et une organisation des connaissances. L'architecture de décision est une architecture hiérarchique à trois niveaux comme présentée dans la Figure 1.1. La nouvelle architecture vise la généralité pour permettre la distribution de méthodes FDIR sur les composants pour la rendre propice aux systèmes hétérogènes et répondre aux contraintes d'embarquabilité. Cependant, aucune méthode de pronostic ou autre méthode de prédiction n'est envisagée dans ce type d'architecture.

1.2.2 Architectures orientées-santé

Certains travaux prennent plus de recul par rapport aux moyens mis en place pour assurer la réalisation de la mission du système. La présence de modules de diagnostic et/ou de pronostic utilisant des connaissances globales sur le système et sur le contexte devient indispensable. La Figure 1.2 présente une forme générique de ce type d'architecture, dans laquelle les différents modules sont tous liés deux à deux afin d'augmenter l'autonomie du système. Un opérateur est inclus dans l'architecture dans un cadre d'aide à la décision pour la maintenance ou pour la mise à jour des objectifs de la mission, par exemple. Contrairement aux approches hiérarchiques, les modules de diagnostic et de pronostic partagent des informations avec les modules de décision et de contrôle. Les différents travaux étudiés par la suite réalisent généralement plusieurs de ces partages mais aucun d'entre eux ne les réalise tous.

Dans [Zhang *et al.* 09], la surveillance de santé d'un système exposé à des fautes multiples est considérée. L'approche est intéressante car les auteurs considèrent la propagation de fautes,

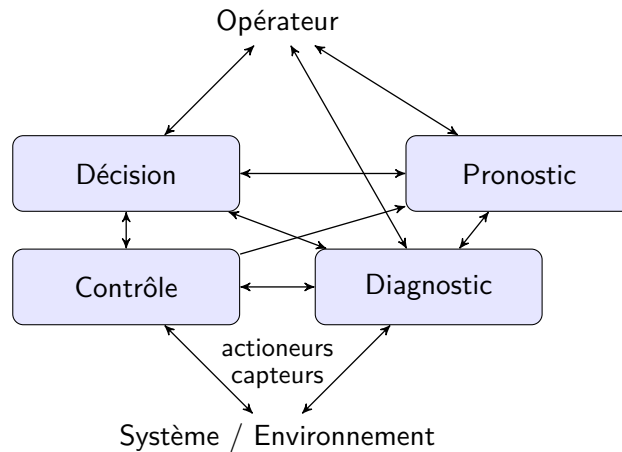


FIGURE 1.2 – Une architecture orientée-santé générique.

c'est-à-dire que l'occurrence d'une faute peut avoir une conséquence sur les prochaines occurrences des fautes. L'algorithme de pronostic, dont l'objectif est de déterminer la durée de vie résiduelle avant la défaillance du système (RUL), n'est déclenché que lorsque le système est dans un état dégradé après l'occurrence d'une faute.

Dans [Roychoudhury & Daigle 11], la surveillance de santé consiste à estimer la progression des fautes par rapport à un modèle nominal. Le pronostic se base sur le diagnostic pour produire une distribution de RUL mais les travaux ne considèrent aucun retour du pronostic vers le diagnostic.

Dans [Chanthery & Ribot 13] et [Zabi *et al.* 13], les méthodes de diagnostic et de pronostic sont intégrées dans un processus unique appelé *InterDP*. Les résultats du pronostic sont utilisés pour lever les ambiguïtés des résultats du diagnostic. L'utilisation des résultats de diagnostic et de pronostic dans un module de planification n'est cependant pas abordée.

Une architecture pour la prise de décision d'un rover autonome, utilisant les résultats de diagnostic et de pronostic pour influencer la planification de la mission est présentée dans [Narasimhan *et al.* 12]. Un module de décision est intégré à l'architecture pour maximiser les objectifs de la mission. L'architecture est décomposée en quatre modules : contrôle de bas niveau, diagnostic, pronostic et prise de décision. L'utilisation des résultats de pronostic pour la détermination du diagnostic du système n'est pas envisagée mais le pronostic utilise la connaissance sur le plan de mission du système.

À notre connaissance, il n'existe pas aujourd'hui d'architecture de gestion de santé intégrant un module de planification et de surveillance de santé, avec un retour du pronostic vers le diagnostic. C'est pour cela que nous avons proposé une nouvelle architecture de santé présentée en introduction (voir la Figure 1).

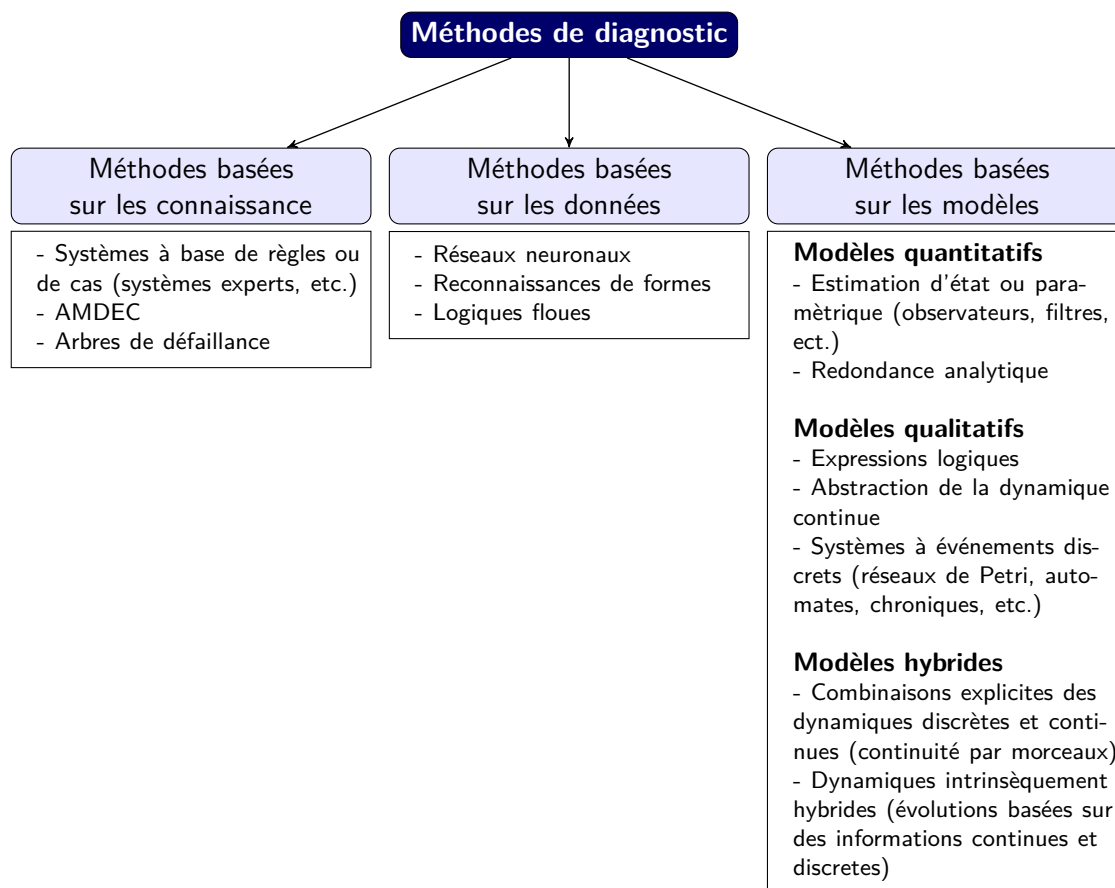


FIGURE 1.3 – Classification des méthodes de diagnostic.

1.3 Méthodes pour le diagnostic

Plusieurs approches existent pour effectuer du diagnostic de système. Elles peuvent être réparties en trois grandes catégories : les méthodes basées sur la connaissance, les méthodes basées sur les données et les méthodes basées sur les modèles (voir la Figure 1.3).

Les **approches basées sur la connaissance** utilisent une connaissance explicite des relations causales entre les symptômes, les défaillances et les fautes [Venkatasubramanian *et al.* 03]. Ces techniques sont utilisées pour identifier les causes des défaillances possibles d'un système à l'aide de relations qui se présentent sous la forme d'associations directes entre les effets et les causes. Ces relations sont souvent déterminées durant la phase de conception et proviennent d'une analyse fonctionnelle ou structurelle du système. Elles peuvent par exemple provenir d'une analyse des modes de défaillance, de leurs effets et de leur criticité (AMDEC ou FMECA) ou encore d'un arbre de défaillance, qui représente les combinaisons possibles d'événements qui peuvent engendrer une panne suite à des études de risque ou de sûreté de fonctionnement. Les approches classiques de diagnostic qui utilisent ce type de connaissance sont des systèmes à base de règles (*rule-based reasoning*) et/ou à base de cas (*case-based reasoning*) désignés généralement comme des systèmes experts [Buchanan & Shortliffe 84, Jackson 88].

Les **approches de diagnostic à base de données** (*data-based*) consistent à utiliser uniquement les informations issues des différents capteurs positionnés sur le système [Dubuisson 90, Zwingelstein 95]. L'objectif de ces méthodes est d'associer des ensembles de mesures à des états de fonctionnement connus du système. Ces approches utilisent des techniques de reconnaissance de formes, de classification, de partitionnement de données, et d'apprentissage de manière générale, pour traiter les données disponibles et construire un modèle du système fondé sur l'expérience. Un tel modèle ne provient donc pas d'une spécification physique ou structurelle du système. Il est construit avec des techniques de type réseaux neuronaux (*neural networks*) ou de logique floue (*fuzzy logic*).

Dans ce manuscrit, on différencie le qualificatif "à base de données" (*data-based*) du qualificatif "guidées par les données" (*data driven*) qui caractérise le fait qu'une méthode utilise des données consécutives sur le système pour fonctionner, comme c'est notamment le cas pour toutes les méthodes utilisant la simulation du comportement du système.

Dans cette étude, on s'intéresse particulièrement aux techniques de **diagnostic basées sur les modèles** (*model-based*). Ces méthodes sont particulièrement adaptées dans le cas où un modèle physique ou un modèle de comportement, même imprécis ou abstrait, peut être obtenu pour le système.

L'utilisation de telles techniques de diagnostic justifie l'utilisation de techniques de modélisation pour représenter le système. Nous définissons la modélisation de la manière suivante.

Définition 11 (Modélisation). Un modèle est un système abstrait de symboles représentant un phénomène concret (observé ou imaginé) de manière interprétable et reproductible. La modélisation est le processus par lequel un modèle est conçu.

Dans les approches à base de modèles, le système est représenté sous la forme d'un ou plusieurs modèles qui décrivent sa structure et son comportement (nominal ou fautif). La méthode de diagnostic compare le comportement observé du système au comportement prédit à l'aide de son modèle. L'occurrence de faute est détectée à partir d'un raisonnement sur les incohérences entre les deux comportements. La Figure 1.4 illustre ce principe. Un modèle incluant les comportements fautifs du systèmes (modèle de faute ou de dysfonctionnement) permet de localiser et d'identifier les fautes survenues.

Dans ces approches, le système est considéré comme dynamique, signifiant que son comportement dépend de son état interne, qui est lié à ce qu'il s'est passé précédemment dans le système. On parle de systèmes dynamiques par opposition au cas très particulier des systèmes statiques, dont les valeurs de sortie dépendent uniquement des valeurs d'entrée. Nous définissons un système dynamique de la manière suivante.

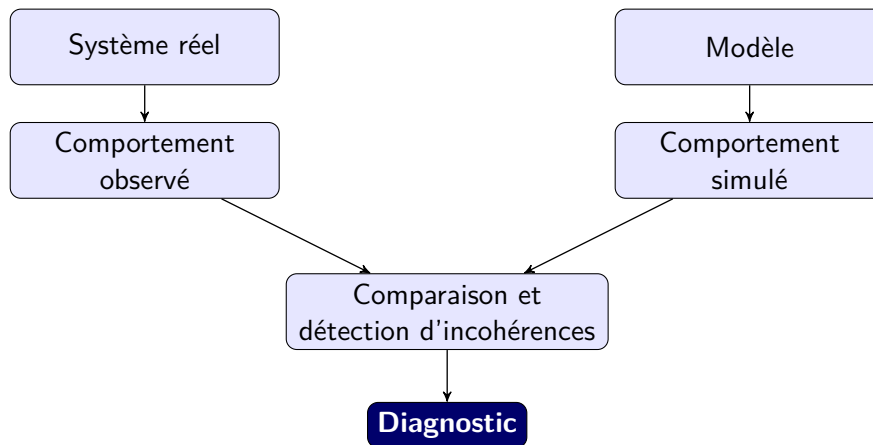


FIGURE 1.4 – Principe du diagnostic basé sur les modèles.

Définition 12 (Système dynamique). Un système dynamique est un système dont le comportement évolue au cours du temps en fonction de sa situation interne, appelée son état, qui est représentative de ce qu'il s'est passé précédemment dans le système.

On distingue généralement deux approches du diagnostic à base de modèles : l'approche FDI (*Fault Detection and Isolation*) et l'approche DX (*Diagnosis*).

1.3.1 Approches continues

L'approche FDI (*Fault Detection and Isolation*) provient de la communauté automatique et utilise des modèles quantitatifs (modèles continus) pour représenter le comportement du système. Elle se base principalement sur la représentation de systèmes continus, qui fait intervenir des variables pouvant avoir un nombre infini de valeurs.

Définition 13 (Système continu). Un système continu est un système donc la dynamique met en jeu des signaux continus.

Les approches à base de modèles continus décrivent l'évolution du système avec des équations différentielles à temps discret ou à temps continu. Le système d'équations suivant donne la forme générale d'un système continu à temps discret :

$$\begin{cases} x_{k+1} = \mathbf{f}(x_k, u_k) \\ y_k = \mathbf{h}(x_k, u_k) \end{cases}, \quad (1.1)$$

où x_k est le vecteur d'état, u_k est le vecteur des variables d'entrée, \mathbf{f} est la fonction d'évolution, y_k est le vecteur des variables de sortie, et \mathbf{h} est la fonction de sortie.

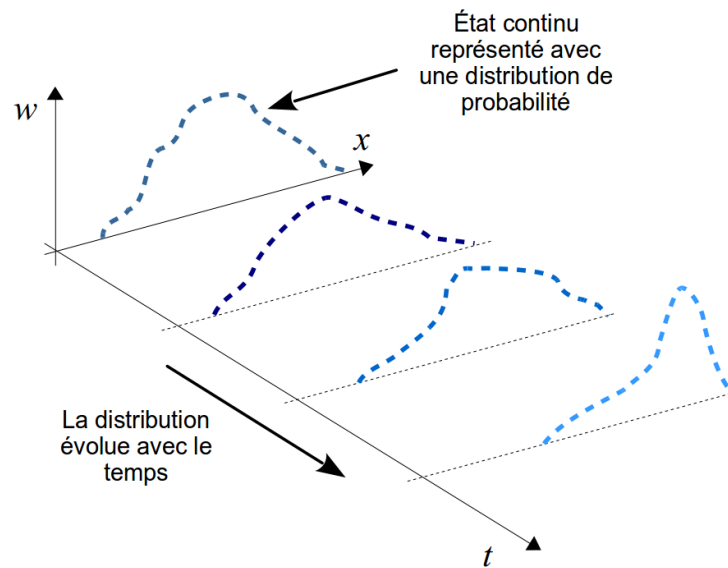


FIGURE 1.5 – Évolution au cours du temps d'une distribution de probabilité représentant l'état continu du système.

À partir du modèle continu du système, deux types de techniques sont employées pour réaliser le diagnostic du système : les techniques à base d'observateurs et les techniques à base de redondances analytiques.

Les techniques à base d'observateurs (ou estimateurs) consistent à estimer les valeurs des variables d'état à partir du modèle et d'un ensemble d'observations sur le système [Ding 14, Staroswiecki *et al.* 91, Hammouriand *et al.* 99]. Dans ces techniques, on utilise la fonction d'évolution du modèle continu (voir l'Équation 1.1) pour calculer une estimation de l'état \hat{x} qui est ensuite corrigée en fonction de la différence entre la sortie estimée \hat{y} , calculée avec la fonction de sortie, et le vecteur des sorties mesurées y .

Un certain nombre de ces observateurs, comme les filtres de Kalman [Arulampalam *et al.* 02], les filtres de Kalman étendus [Julier & Uhlmann 97] ou les filtres particuliers [Thrun *et al.* 01, van der Merwe *et al.* 00], considèrent du bruit sur l'évolution de l'état mais aussi sur les observations. L'estimation de l'état est alors manipulée comme une distribution continue de probabilité, de croyance, ou de pertinence. L'évolution d'une telle estimation distribuée est illustrée dans la Figure 1.5.

Dans les filtres de Kalman [Arulampalam *et al.* 02], cette distribution prend la forme d'une valeur centrale à laquelle est jointe une matrice de covariance de l'erreur. Les filtres de Kalman classiques se limitent à l'étude des systèmes linéaires. Les filtres de Kalman étendus permettent de travailler avec des systèmes non linéaires et utilisent des jacobiniennes (matrices des dérivées partielles) ou encore des "points sigma" (échantillon de points qui conserve la covariance de l'état) [Julier & Uhlmann 97, Julier & Uhlmann 04].

Dans les filtres particulaires, la distribution de croyances se compose d'un ensemble fini de particules, représentant chacune une valeur possible de l'état du système [van der Merwe *et al.* 00, Thrun *et al.* 01]. Durant une étape de prédiction bayésienne, chaque particule est mise à jour avec la fonction d'évolution du modèle (voir l'Équation 1.1) à laquelle est ajouté un bruit aléatoire. Une étape de correction associe alors un poids à chaque particule en fonction de l'erreur entre sa sortie estimée (calculée avec l'équation de sortie du modèle) et les observations sur le système, puis utilise ce poids pour rééchantillonner l'ensemble des particules. Le rééchantillonnage consiste à sélectionner (une ou plusieurs fois) les particules ayant les poids les plus élevés pour former la nouvelle distribution de particules estimant l'état courant du système.

Ces techniques à base de filtres sont également utilisées pour faire de l'estimation paramétrique [Liu & West 01, Yang *et al.* 07]. Ces travaux cherchent à estimer les valeurs des paramètres du système qui peuvent être considérés comme des indicateurs de défaillance. Il existe des méthodes combinant les estimations des variables d'état et des paramètres : on parle d'estimation conjointe d'état et de paramètres (*joint state-parameter estimation*) [Roychoudhury & Daigle 11, Chi *et al.* 13, Daigle & Kulkarni 14]. Techniquement, cela revient à inclure les paramètres à estimer dans le vecteur des variables d'état.

Les techniques basées sur les relations de redondance analytique (*Analytical Redundancy Relations* ou ARR) ont également été largement appliquées dans le cadre du diagnostic de systèmes continus [Cordier *et al.* 04, Ding 14]. Une relation de redondance analytique est une équation dans laquelle toutes les variables inconnues ont été remplacées par des expressions ne faisant intervenir que des variables observées. Ces méthodes se basent sur la génération de résidus du modèle pour déterminer les valeurs pertinentes à mesurer. Un résidu correspond à une différence entre le comportement simulé par le modèle et le comportement observé du système. Nous utilisons la définition suivante issue de [Gertler 98].

Définition 14 (Résidu). Un résidu est associé à une différence résultant de la comparaison de mesures de capteurs à des valeurs calculées analytiquement de la variable considérée dans le système modélisé.

Pour la détection de fautes basée sur l'utilisation de modèles, un résidu est un signal temporel, fonction des entrées et des sorties du processus, indépendant (le plus possible) du point de fonctionnement de celui-ci. En l'absence de faute, ce résidu est statistiquement nul. Lors de l'apparition d'une faute, son amplitude évolue de manière significative. L'approche la plus classique est celle dite de l'espace de parité [Staroswiecki *et al.* 91]. Les relations de parité utilisent la redondance directe au moyen de relations algébriques statiques liant les différents signaux ou la redondance temporelle issue de l'utilisation de relations dynamiques. Le terme "parité" a été emprunté au vocabulaire employé pour les systèmes logiques où la génération de bits de parité permet la détection d'erreur. Les méthodes par espace de parité consistent à

projeter les estimations des résidus dans un espace binaire afin de reconnaître ce qu'on appelle des signatures des fautes. Les signatures de fautes sont des vecteurs d'indicateurs booléens représentatifs de chaque mode de fonctionnement du système.

1.3.2 Approches discrètes

L'approche DX (*Diagnosis*) provient de la communauté de l'intelligence artificielle et utilise des modèles qualitatifs (modèles logiques ou discrets) pour décrire le système. Elle se base principalement sur l'utilisation de systèmes discrets, qui travaillent avec des variables pouvant prendre un nombre fini de valeurs. Un cas particulier des systèmes discrets sont les systèmes à événements discrets (*Discrete Event Systems* ou DES) dont la définition suivante est inspirée de [Cassandras & Lafortune 09].

Définition 15 (Système à événements discrets). Un système à événements discrets est un système dont l'espace d'état est discret, et dont l'évolution de l'état dépend de l'occurrence d'événements au fil du temps.

Plusieurs formalismes existent pour représenter les systèmes à événements discrets, tels que les automates [Sampath *et al.* 95], les réseaux de Petri [David & Alla 05], ou encore les chroniques [Dousson *et al.* 93].

Dans les DES, les approches les plus utilisées sont les approches à base de modèles de fautes. L'approche la plus connue est l'approche du diagnostiqueur introduite dans [Sampath *et al.* 95, Sampath *et al.* 96] et qui se base sur les automates.

Définition 16 (Automate). Un automate (ou un système de transitions étiquetées) est un 4-uplet $G = \langle S, E, T, s_0 \rangle$ dans lequel :

- S est un ensemble fini d'états,
- E est l'ensemble des labels d'événement apparaissant dans G ,
- $T \subseteq S \times E \times S$ est l'ensemble des transitions,
- s_0 est l'état initial.

Un diagnostiqueur est un automate obtenu à partir d'un modèle automate contenant des événements de fautes (non observables) et dont les événements déclenchant les transitions sont les événements observables du système. L'état du diagnostiqueur donne par conséquent, à tout instant, une information sur les fautes pouvant expliquer le comportement observé. Le diagnostic consiste à identifier ces événements non observables (les fautes). La détection et l'identification d'une faute se font sur le même automate. En effet, la détection d'une faute

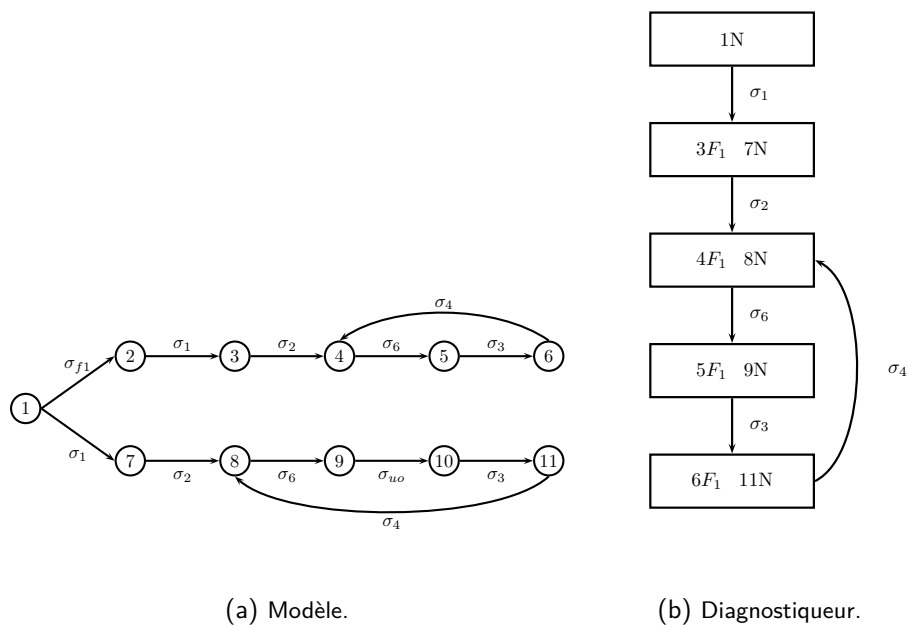


FIGURE 1.6 – Exemple d'un modèle automate et de son diagnostiqueur [Soldani 08]

se fait lorsqu'il n'existe plus d'ambiguïté entre l'hypothèse du comportement normal et celle d'un comportement de faute.

Le génération du diagnostiqueur est un processus hors-ligne et nécessite un modèle automate. Le processus de diagnostic est quant à lui effectué en ligne et consiste à parcourir le diagnostiqueur en fonction des observations reçues et à donner l'information résumée dans l'état courant du diagnostiqueur.

Sur la Figure 1.6(b), un exemple de diagnostiqueur est représenté. Il est établi à partir d'un modèle automate (Figure 1.6(a)) dont l'état initial est l'état 1. Sur cet automate modèle sont représentés des événements observables σ_i , des événements non observables σ_{uo} , et un événement de faute (non observable) σ_{f1} . Chaque état du diagnostiqueur est constitué de plusieurs états du modèle automate correspondant aux différents modes de fautes. Des labels sont associés à ces états : N pour décrire le comportement normal, F_i pour décrire qu'une faute F_i a eu lieu. À l'état initial, le diagnostiqueur est dans l'état 1 et aucune faute ne s'est produite (label N). Si σ_1 est observé, le modèle est soit dans l'état 7, correspondant à un comportement normal (7N), soit dans l'état 3, correspondant à l'occurrence d'une faute f_1 ($3F_1$). Il existe donc une ambiguïté entre le comportement normal et la possibilité de l'occurrence d'une faute (7N, $3F_1$).

Le diagnostiqueur de [Sampath *et al.* 96] est une approche centralisée chargée de surveiller, détecter et diagnostiquer les fautes du système global. Il existe de nombreux travaux sur la distribution [Su *et al.* 02] et la décentralisation [Debouk *et al.*] de l'approche diagnostiqueur.

Comme expliqué dans l'introduction, nous nous plaçons dans un cadre centralisé. Ces approches ne sont donc pas l'objet de notre étude.

Les réseaux de Petri sont souvent plus appropriés pour des travaux de modélisation car ils représentent de manière plus compacte et plus intuitive l'évolution des DES que les automates. Un réseau de Petri est un graphe biparti pour lequel un nœud peut être soit une **place**, soit une **transition**. Les places et les transitions sont connectées entre elles par des **arcs** orientés mais un arc ne peut pas lier deux places ou deux transitions. Les places peuvent contenir des **jetons** et la distribution des jetons dans les places est appelée le **marquage** du réseau.

Définition 17 (Réseau de Petri). Un réseau de Petri est un 5-uplet $R = \langle P, T, I, O, M_0 \rangle$, où :

- P est l'ensemble des places,
- T est l'ensemble des transitions,
- M_0 est le marquage initial des places : $M : P \rightarrow \mathbb{N}$, où la valeur $M_0(p)$ est le nombre de jetons dans la place p .
- I est la fonction d'entrée : $I : P \times T \rightarrow \mathbb{N}$. La valeur $I(p, t)$ est le poids de l'arc de la place p à la transition t .
- O est la fonction de sortie : $O : P \times T \rightarrow \mathbb{N}$, où la valeur $O(p, t)$ est le poids de l'arc de la transition t à la place p .

Les fonctions d'entrée I et de sortie O sont représentées sous forme matricielle. Les lignes sont associées aux places et les colonnes aux transitions. Une transition t_i peut être tirée pour un marquage M si est seulement si $M \geq I(., t_i)$ où $I(., t_i)$ est la $i^{\text{ème}}$ colonne de la matrice représentant la fonction d'entrée.

La Figure 1.7 présente un réseau de Petri ayant quatre places ($P = \{p_1, p_2, p_3, p_4\}$) et deux transitions ($T = \{t_1, t_2\}$). Graphiquement, les places sont représentées par des cercles, les transitions par des traits noirs, les arcs par des flèches et les jetons par des points noirs. Si aucune valeur n'est annotée à un arc, son poids vaut 1 par défaut. La transition t_1 est donc tirable car la place p_1 contient au moins un jeton, alors que t_2 ne l'est pas car p_2 n'en contient pas.

Nous définissons également le graphe des marquages comme étant un graphe composé de nœuds qui correspondent aux marquages accessibles, et d'arcs correspondant aux franchissements de transitions faisant passer d'un marquage à l'autre.

Le formalisme des réseaux de Petri permet de modéliser de nombreux phénomènes et de vérifier le comportement dynamique de systèmes comme les systèmes manufacturiers, les systèmes de télécommunications, les réseaux de transport, etc., car il peut représenter des évolutions parallèles et inclut des mécanismes de synchronisation.

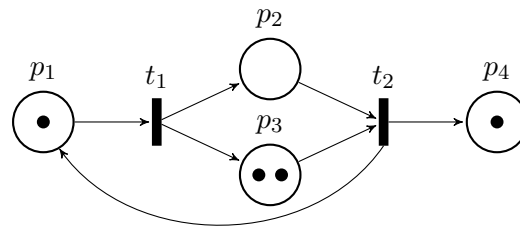


FIGURE 1.7 – Exemple d'un réseau de Petri.

Certains travaux étendent l'approche diagnostiqueur aux systèmes à événements discrets modélisés par des réseaux de Petri [Boubour *et al.* 97, Genc & Lafortune 07, Soldani *et al.* 07, Cabasino *et al.* 14]. Les auteurs de [Genc & Lafortune 07] proposent une version distribuée du diagnostiqueur. Dans [Cabasino *et al.* 14], les auteurs étudient la diagnosticabilité d'un système. Ils s'inspirent de la diagnosticabilité des automates à états finis définie dans les travaux de [Sampath *et al.* 95]. La construction d'un graphe des marquages accessibles modifié (*modified basis reachability graph*) permet d'obtenir un diagnostiqueur d'accessibilité, présenté comme un graphe.

Les travaux de [Soldani *et al.* 07] propose une approche pour la localisation de fautes intermittentes. Les auteurs s'intéressent à la gestion des observations partielles dans le cadre des systèmes à événements discrets. Ils proposent une méthode basée sur les réseaux de Petri pour modéliser le fonctionnement normal observable du système. L'étape de détection compare le flux des événements observés avec le flux des événements prédit par le modèle. Un mécanisme de localisation, basé sur l'approche du diagnostiqueur, indique les ensembles d'événements possibles expliquant les fautes. Ces travaux ne gèrent pas les incertitudes liées à la connaissance sur le système ou aux observations discrètes.

D'autres travaux s'intéressent particulièrement aux incertitudes sur les observations du système [Giua & Seatzu 02, Ru & Hadjicostis 09, Basile *et al.* 09, Jianxiong *et al.* 13]. Dans les travaux de [Giua & Seatzu 02], une approche à base d'estimateur est utilisée sur des réseaux de Petri dont le marquage est partiellement observable. L'idée est d'estimer le véritable marquage d'un réseau de Petri par un marquage calculé à partir des événements observés. Plusieurs types d'incertitude sont considérés, comme la connaissance partielle ou nulle du marquage initial, mais la structure du réseau est considérée comme connue et tous les tirages des transitions sont observables. Dans [Ru & Hadjicostis 09], les réseaux de Petri partiellement observés (*Partially Observed Petri Nets* ou POPN) sont utilisés. Les réseaux de Petri partiellement observés sont transformés en des réseaux de Petri labellisés équivalents et un moniteur est construit pour diagnostiquer les fautes et délivrer une croyance (degrés de confiance) sur leurs occurrences. Cette approche ne considère que l'incertitude au niveau du résultat du diagnostic mais pas les incertitudes liées au modèle ou encore aux observations des événements. Les auteurs

de [Basile *et al.* 09] propose de réduire l'explosion combinatoire de l'espace d'état des estimations en introduisant un marquage généralisé (jetons négatifs) pour prendre en compte l'incertitude sur le tirage des transitions. Les travaux de [Jianxiong *et al.* 13] utilise les réseaux de Petri stochastiques (*Stochastic Petri Nets* ou SPN) pour construire une modèle formel de chaque composant d'une architecture modulaire avionique intégrée. Dans toutes ces approches, aucun aspect continu n'est considéré dans le modèle du système.

1.3.3 Approches hybrides

Dans le cadre des systèmes complexes, il devient trop réducteur de considérer les systèmes comme seulement à dynamique continue ou discrète : on parle de systèmes hybrides, dont la dynamique peut être à la fois continue et discrète [Henzinger 96].

Définition 18 (Système hybride). Un système hybride est un système dynamique faisant intervenir explicitement et simultanément des dynamiques de type continu et discret.

Les systèmes hybrides sont le centre d'intérêt de nombreuses recherches dans des domaines variés, comme la modélisation, la vérification, le contrôle, et la surveillance.

Des modèles, initialement continus, ont été étendus par l'intégration d'aspects événementiels [Djeziri *et al.* 07]. De la même façon, des modèles à événements discrets ont été étendus par l'intégration de l'aspect continu, tels que les réseaux de Petri continus (*Continuous Petri Nets* ou CPN) [David & Alla 05] et hybrides (*Hybrid Petri Nets* ou HPN) [Zaidi *et al.* 06, Dotoli *et al.* 08], qui introduisent un nouveau type de place (les places continues) portant un marquage rationnel. Enfin, des modèles hybrides ont été construits par la combinaison explicite d'un modèle à événements discrets et d'un modèle continu, tels que les automates hybrides (*Hybrid Automaton* ou HA) [Bayouhd *et al.* 08, Hofbaur & Williams 02] ou les réseaux de Petri particuliers (*Particle Petri Nets* ou PPN) [Lesire & Tessier 05].

L'avantage de ces derniers modèles est qu'ils offrent une modélisation naturelle de la partie continue et de la partie discrète, propice à l'intégration du calcul de diagnostic et de pronostic [Chanthery & Ribot 13]. Dans le cadre des systèmes hétérogènes, dont les composants peuvent être soit purement discrets, soit purement continus, soit hybrides, l'extension du modèle hybride est possible.

Toutes ces modèles ont été largement utilisés et étendus pour la surveillance des systèmes hybrides.

Dans [Bayouhd *et al.* 08], les auteurs modélisent le système avec un automate hybride. Le système hybride est décrit comme un système multimode (*multi-mode system*) dans lequel à

chaque mode est associée une dynamique continue particulière. Ces travaux exploitent ensuite les relations de redondance analytique des modèles continus et l'approche par espace de parité pour générer un diagnostiqueur DES qui reconnaît les signatures associées à chacun des modes de fonctionnement.

Les auteurs de [Lesire & Tessier 05] combinent un modèle à événements discrets (réseau de Petri) et un modèle continu (équations dynamiques) dans une extension des réseaux de Petri hybrides appelée les réseaux de Petri particuliers (PPN). Ils proposent de distribuer le marquage rationnel des places continues en un ensemble de particules. Les jetons des places discrètes (les configurations) et ces particules sont ensuite utilisés dans un mécanisme de surveillance combinant le tirage possibiliste [Cardoso *et al.* 99] (duplication de jeton ou pseudo-tirage) avec un filtre particulière pour traiter toutes les incertitudes relatives au système et aux observations discrètes et continues. Ces travaux sont orientés vers la surveillance de mission et non sur la surveillance de santé.

Le formalisme des réseaux de Petri particuliers modifiés (*Modified Particle Petri Nets* ou MPPN) est proposé dans [Zouaghi *et al.* 11a] comme une extension des PPN. L'avantage principal des MPPN est qu'ils proposent d'utiliser des transitions associées à des conditions portant à la fois sur les valeurs des configurations et des particules. Dans [Zouaghi *et al.* 11a], l'application est essentiellement orientée vers la surveillance de mission et non sur la surveillance de santé. Les différents états de santé du système ne sont pas considérés. De plus, il n'y a pas de correspondance avec l'objet diagnostiqueur défini dans la littérature et le problème d'ambiguïté dans le modèle n'est pas abordé. Dans [Zouaghi *et al.* 11b], les auteurs proposent une approche pour la spécification et la surveillance dynamique de systèmes. Ils proposent d'intégrer la conception du système et la surveillance en un formalisme unifié générique pour la génération automatique de moniteur de composant à partir de la description du composant. Cependant, l'observateur généré n'est pas formellement défini. Il n'est ni comparé ni appliqué dans un contexte de gestion de santé.

1.3.4 Diagnostic avancé

Certains travaux s'intéressent particulièrement au diagnostic des systèmes avec l'intention de l'utiliser à des fins de pronostic. Ces approches considèrent le suivi de la dégradation du système. Ceci est appelé le **diagnostic avancé**.

Dans [Vianna & Yoneyama 15], la méthode de diagnostic utilise un filtre de Kalman étendu et un algorithme IMM (*Interactive Multiple-Model*) pour surveiller à la fois le comportement du système et sa dégradation dans le but d'obtenir une meilleure estimation de l'état du système comme point de départ du processus de pronostic. L'approche est cependant limitée aux systèmes continus.

Les auteurs de [Chanthery & Ribot 13] proposent d'étendre la technique du diagnostiqueur de [Bayouhd *et al.* 08] en associant à chaque mode du système hybride une dynamique de dégradation. Ces dynamiques sont limitées à des lois de vieillissement qui estiment les probabilités d'occurrences des fautes anticipées. Les probabilités évoluent différemment en fonction des facteurs de stress associés à chaque mode de fonctionnement [Kirkland *et al.* 04, Wilkinson *et al.* 04].

Définition 19 (Facteur de stress). Les facteurs de stress d'un système représentent toutes les sollicitations (normales et anormales) qui peuvent influencer l'état de santé du système.

Des exemples de facteurs de stress d'un système sont ses conditions opérationnelles (fonctionnalités indisponibles, mode de survie, etc.), les conditions environnementales (humidité, température, vibrations mécaniques, etc.), ou encore les fautes redoutées et leurs conséquences sur les composants du système. Les lois de vieillissement peuvent être alors caractérisées comme hybrides car leurs évolutions dépendent d'informations discrètes et continues. L'approche de [Chanthery & Ribot 13] ne tient cependant pas compte des incertitudes sur le modèle du système et sur ses observations, tant pour la partie continue que discrète.

Dans [Yu *et al.* 11], l'isolation des fautes est effectuée dynamiquement avec un graphe de liaison hybride (*Hybrid Bond Graph* ou HBG). La méthode propose d'utiliser une matrice de signatures de faute pour de chaque mode et introduit un retard pour permettre à chaque faute d'exprimer ses symptômes sur les résidus (particulièrement les fautes uniquement détectables avec les signaux continus). Aucune indication sur le temps d'attente n'est toutefois donnée. En parallèle de la surveillance de l'évolution de ces fautes naissantes, les dégradations de chaque composant dépendent du mode de fonctionnement courant et sont estimées avec un algorithme d'évolution différentielle hybride (*hybrid differential evolution*).

1.4 Méthodes pour le pronostic

En ce qui concerne le pronostic, deux méthodes sont principalement adoptées. La première consiste à utiliser une méthode basée sur la résolution analytique du modèle. L'idée est d'avoir une fonction de dégradation du système et de la projeter dans le futur pour connaître la date de fin de vie du système (approche purement continue). Ces techniques sont rapides en terme de performances calculatoires mais la précision des résultats est très dépendante de la qualité du modèle. La deuxième méthode consiste à simuler l'évolution du système en combinant les aspects discrets et continus de son modèle. On dira alors que le processus de pronostic est séquentiel car il considère l'évolution future du système et ses changements de modes.

Ces techniques sont généralement plus précises que les méthodes analytiques mais aussi plus coûteuses en calcul car elles simulent pas à pas l'évolution du système.

Les différentes approches pour effectuer le pronostic de système peuvent être classées de la même manière que celles pour faire du diagnostic (voir Figure 1.3). Par ailleurs, la qualité des résultats du processus de pronostic dépend de la connaissance que l'on a du système : la connaissance sur le vieillissement ou sur l'usure du système, ou plus généralement sur l'évolution du système dans le futur.

Les approches basées sur la connaissance reposent sur l'expérience et constituent une alternative à une modélisation difficile de la nature physique du système. Ces techniques dérivent des systèmes experts basés sur un ensemble de règles et d'associations, ou sur un raisonnement à base de cas [Berenji & Wang 06]. Les informations issues du domaine de la fiabilité permettent d'associer une distribution statistique aux données. Les prédictions basées sur la fiabilité reposent sur des modèles qui établissent des lois de durée de vie à partir de données issues de l'expérience [Kirkland *et al.* 04]. Une étude de fiabilité utilise principalement le traitement statistique des données pour prédire la probabilité de défaillance d'un système tout au long de sa vie et donc déterminer sa durée de vie (RUL). La probabilité de défaillance d'un système est généralement représentée avec des lois de probabilité telles que la loi normale, les lois logarithmes ou exponentielles, ou encore la loi de Weibull [Ribot 09, Zabi *et al.* 13]. Ces techniques fournissent des estimations de RUL très rapidement car elles se basent sur des résolutions analytiques des équations du modèle.

Comme les approches basées sur la connaissance, les approches de pronostic à base de données n'utilisent pas la connaissance sur la nature physique du système. Elles combinent des méthodes d'apprentissage et des méthodes d'estimation en utilisant les données issues de l'expérience et/ou les données en ligne. Les méthodes d'apprentissage permettent l'extraction des tendances en utilisant des techniques du domaine de l'intelligence artificielle telles que les réseaux neuronaux, la reconnaissance de formes, la classification ou les approches floues [Gouriveau *et al.* 13, Javed *et al.* 15]. Des estimateurs d'état et/ou de paramètres (filtres de Kalman, filtres particuliers) sont utilisés pour suivre les tendances apprises et évaluer l'état de dégradation du système à l'aide des données en ligne et d'indicateurs de vieillissement (caractéristiques mesurables sur le système) [Roemer *et al.* 05]. Le processus de pronostic est alors plus coûteux que les méthodes à base de connaissance car il repose généralement sur la simulation du modèle dans le futur. Comme pour le diagnostic, les approches de pronostic basées sur les données restent particulièrement limitées car elle dépendent de la disponibilité de données représentatives des situations de fautes ou de défaillances du système.

Les méthodes à base de modèles diffèrent des méthodes à base de données principalement sur la construction du modèle de vieillissement qui utilise la connaissance physique ou fonctionnelle du système. Comme pour le diagnostic de système, les approches de pronostic à base de

modèles utilisent des modèles discrets, continus ou hybrides. Ce sont ces approches que nous détaillerons ici.

1.4.1 Approches continues

Les approches à base de modèles continus les plus communes sont constituées de deux phases : une phase d'estimation et une phase de prédiction [Vinson *et al.* 13, Jouin *et al.* 16, Orchard *et al.* 08].

La phase d'estimation (ou d'apprentissage) consiste à utiliser des observateurs sur les données en ligne pour estimer conjointement l'état du système et les paramètres qui sont des indicateurs de la dégradation du système.

La phase de prédiction consiste à prédire l'évolution de l'état du système dans le futur à partir de l'estimation de l'état courant et d'un modèle physique de la dégradation afin de déterminer une estimation du RUL sous la forme d'une distribution de probabilité.

Les estimateurs les plus utilisés sont les filtres de Kalman étendus [Daigle *et al.* 14b] et les filtres particuliers [Saha & Goebel 11, Daigle & Goebel 13]. Ces derniers sont particulièrement appropriés pour la prédiction, car la connaissance sur l'état courant est déjà discrétisée en un ensemble d'états possibles. Le processus de pronostic devient très simple : il faut répéter l'étape de prédiction en propageant les incertitudes sur l'évolution de l'état avec le bruit aléatoire associé au modèle.

L'utilisation des filtres de Kalman étendus ou des filtres particuliers dans des méthodes de pronostic a montré son efficacité dans de nombreuses applications comme les rovers planétaires [Daigle & Sankararaman 13], les vannes pneumatiques [Daigle *et al.* 14a], ou encore les piles à combustible [Jouin *et al.* 16].

1.4.2 Approches discrètes

Les approches de pronostic qui utilisent des modèles discrets reposent principalement sur la prédictabilité des occurrences d'événement. La prédictabilité concerne l'analyse du système à événements discrets à partir de l'état courant du système pour déterminer si une faute peut encore avoir lieu ou non. Nous reprenons la définition de [Cao 89], pour décrire la prédictabilité.

Définition 20 (Prédictabilité). L'occurrence d'un événement significatif est prédictible dans le système si après l'observation d'une séquence finie d'événements, un système de surveillance peut dire avec certitude que l'événement va apparaître ou non.

Des approches de pronostic à base d'automates ont été développées dans de nombreux travaux [Genc & Lafortune 09, Kumar & Takai 10]. Dans [Genc & Lafortune 09], le principe de la prédictabilité est défini dans le contexte des DES partiellement observables. Les travaux proposent ensuite deux méthodes basées sur les automates pour vérifier la prédictabilité d'un système. La première méthode utilise un diagnostiqueur hors-ligne tandis que la seconde méthode utilise les observations en ligne et se base sur un vérificateur. Une approche de prédictabilité décentralisée est proposée dans [Kumar & Takai 10] avec la construction de pronostiqueurs locaux. Les auteurs définissent la notion de co-pronosticabilité (qui correspond dans nos termes à la co-prédictabilité) qui caractérise si un pronostiqueur local est suffisant pour prédire l'occurrence d'un événement local.

Des approches à base de réseaux de Petri ont aussi été proposées dans [Basile *et al.* 09, Madalinski & Khomenko 12, Lefebvre 14]. Les auteurs de [Basile *et al.* 09] utilisent un marquage généralisé (*generalized marking*) pour tenir compte des occurrences d'événements non observables tout en minimisant l'explosion de l'espace d'état. Dans [Lefebvre 14], le problème est étudié avec des réseaux de Petri partiellement observés (POPON). Cette approche représente les états courant et futurs possibles sous la forme de distributions de croyance. La distribution sur l'état courant est calculée avec les observations sur le système et sert de point de départ pour le calcul de la distribution sur les états futurs possibles.

La limite des approches discrètes est le manque d'information quantitative concernant l'estimation de la dégradation et la prédiction du RUL du système.

1.4.3 Approches hybrides

À notre connaissance, les approches de pronostic à base de systèmes hybrides sont peu nombreuses.

Dans [Yu *et al.* 14], les auteurs proposent de développer une méthode de pronostic séquentielle pour des systèmes hybrides dans lesquels les fautes ne sont pas détectables immédiatement. Les RUL des composants sont calculés à partir des estimations de l'état de dégradation et d'un seuil de dégradation fixé a priori. L'algorithme de pronostic est activé dès qu'une incohérence est détectée au niveau du diagnostic et simule la propagation des fautes isolées. Un module auxiliaire se déclenche lorsque qu'un changement de mode survient pendant le déroulement de l'algorithme de pronostic, pour réduire la simulation aux fautes identifiées. Ces travaux ne considèrent cependant pas en compte les incertitudes au niveau du modèle et des observations.

Dans [Daigle *et al.* 15a], un cadre de modélisation générique pour le pronostic des systèmes hybrides est proposé. L'estimation de l'état hybride du système peut être réalisée par différentes techniques déjà utilisées dans le cadre du diagnostic de systèmes hybrides. La méthode de pronostic se base sur le modèle hybride, sur l'estimation de l'état et les entrées futures

sur le système. Le processus de pronostic est séquentiel et change de mode en suivant les spécifications du modèle.

Dans [Chanthery & Ribot 13], les automates hybrides étendus avec des lois de vieillissement sont utilisés pour générer le **pronostiqueur** du système. Il s'agit d'un outil de pronostic qui détermine le RUL du système, dès qu'un nouveau diagnostic est disponible. Le diagnostic est calculé à l'aide d'un diagnostiqueur qui surveille le comportement du système avec les observations et qui estime en parallèle la dégradation du système. La méthode est démontrée sur un cas d'étude simulé dans [Zabi *et al.* 13]. La dégradation du système est représentée par des lois Weibull modélisant les probabilités d'occurrences des fautes. Lorsque la probabilité d'occurrence d'une faute est supérieure à un seuil fixé, le pronostiqueur considère un changement de mode du système, qui est alors soumis à un niveau de stress différent et donc à des lois de vieillissement différentes. Le processus de pronostic se base sur le diagnostic et utilise un raisonnement analytique sur les lois de vieillissement pour déterminer le RUL, rendant le calcul du pronostic rapide. L'approche est cependant limitée car elle ne prend pas en compte les changements de mode dus à l'évolution des entrées sur le système par exemple.

1.4.4 Fusion de méthodes de pronostic

Les approches à base de modèles n'utilisent pas particulièrement de techniques d'apprentissage issues de l'intelligence artificielle mais certains travaux récents s'intéressent particulièrement à l'intégration des modèles physiques aux méthodes basées sur les données et l'apprentissage. On parle alors de méthodes fusionnées de pronostic (*data-model-fusion prognostics*) [Liao & Köttig 14, Zhang *et al.* 09].

Ce type d'approches s'intéresse principalement à améliorer la prédiction du RUL en travaillant sur une meilleure estimation des observations futures sur le système. Dans [Daroogheh *et al.* 15], par exemple, les auteurs combinent le pronostic à base de filtres particuliers avec des réseaux neuronaux qui prédisent les observations futures sur le système.

1.5 Conclusion

L'étude des architectures de supervision de santé montre qu'il est maintenant indispensable de considérer les fonctions de diagnostic et de pronostic dès la conception de l'architecture du système. Leur intégration n'a jamais été proposée.

Par ailleurs, l'étude des formalismes utilisés pour la surveillance des systèmes montre que beaucoup de travaux ont été réalisés sur le diagnostic de systèmes hybrides, mais peu sur le pronostic de systèmes hybrides. La gestion des incertitudes au niveau du diagnostic est parfois

le sujet de travaux mais peu de travaux concernent le couplage interactif du diagnostic et du pronostic pour les systèmes hybrides.

Un nouveau formalisme peut donc être envisagé. Il doit idéalement :

- utiliser les réseaux de Petri pour bénéficier de leur capacité d'expression et leur compacité,
- pouvoir exprimer à la fois les contraintes des systèmes à événements discrets et des systèmes continus,
- pouvoir gérer les incertitudes sur la connaissance du système et sur les observations,
- pouvoir exprimer des caractéristiques purement hybrides.

Ces travaux proposent l'utilisation d'un formalisme étendu des MPPN, appelé **réseaux de Petri hybrides particuliers** (*Hybrid Particle Petri Nets* ou **HPPN**), dans un contexte de surveillance de santé des systèmes hybrides faisant face aux incertitudes inhérentes aux systèmes réels. La construction d'un diagnostiqueur et d'un pronostiqueur ainsi que leur processus sont formalisés.

Chapitre 2

Cadre de modélisation

Résumé

Ce chapitre présente le cadre de modélisation dans lequel se placent ces travaux pour étudier l'évolution de la santé des systèmes.

Dans un premier temps, le formalisme des réseaux de Petri hybrides particuliers (HPPN) est introduit. Les réseaux de Petri, usuellement utilisés pour représenter des dynamiques discrètes, sont ici enrichis par des dynamiques continues et hybrides. La structure des HPPN est détaillée, des règles de tirages sont définies et des propriétés sont énoncées.

Dans un second temps, les HPPN sont utilisés pour modéliser l'évolution de la santé d'un système hybride. Chaque mode est représenté par trois places : une place symbolique, une place numérique, et une place hybride. Les différents types de jetons représentent l'état du système, c'est-à-dire ses états discret, continu et hybride. Les transitions représentent les changements de modes. Des conditions sont associées aux transitions pour contextualiser ces changements de modes. Un changement de mode peut par exemple avoir lieu sur occurrence d'un événement (faute, événement de mission, interaction avec l'environnement, etc.) ou dès qu'une contrainte sur l'état continu et/ou sur l'état hybride est satisfaite.

La méthodologie de modélisation est appliquée sur un exemple de robot mobile qui est réutilisé dans les chapitres suivants pour illustrer les concepts.

La dernière section de ce chapitre présente un système de réservoirs d'eau, cas d'étude simulé qui permet d'illustrer en profondeur les concepts théoriques. Dans ce chapitre, il est décrit puis modélisé avec les HPPN.

2.1 Introduction

Dans un contexte de gestion de santé, la modélisation a pour objectif de représenter les changements de dynamiques du système lorsqu'une ou plusieurs fautes surviennent. Un **mode** décrit l'évolution du système dans des conditions de santé spécifiques. Tant qu'aucune faute n'est survenue, le système est dans un **mode nominal**. Lorsqu'une faute survient, le système entre dans un **mode dégradé**. Sans réparation, le système peut finir dans un **mode de défaillance**, dans lequel il n'est plus opérationnel.

Ce chapitre présente le cadre de modélisation proposé pour étudier l'évolution de la santé d'un système.

La Section 2.2 introduit les travaux avec la description d'un robot mobile, un exemple simple mais très complet, qui est réutilisé dans les chapitres suivants pour illustrer les concepts.

La Section 2.3 détaille ensuite le formalisme des **réseaux de Petri hybrides particuliers** (*Hybrid Particle Petri Nets* ou **HPPN**). Les réseaux de Petri, usuellement utilisés pour représenter des dynamiques discrètes, sont ici enrichis par des dynamiques continues et hybrides. Leur utilisation pour modéliser des systèmes hybrides est ensuite décrite, puis illustrée sur l'exemple du robot mobile.

La Section 2.4 explique comment les HPPN peuvent être utilisés pour modéliser l'évolution de la santé d'un système. La méthodologie est suivie pour modéliser l'évolution de la santé du robot mobile.

Enfin, la Section 2.5 présente un système de réservoirs d'eau, un cas d'étude simulé qui met en valeur les avantages de la modélisation à base de HPPN.

2.2 Exemple du robot mobile

Un robot mobile, dont l'évolution est décrite en Figure 2.1, est utilisé tout au long du manuscrit pour illustrer les définitions et les concepts.

Remarque 2.1 (Représentation graphique d'un système multimode). L'évolution du système est décrit avec un graphe orienté, dans lequel les nœuds représentent les modes et les arcs représentent les changements de mode. Le mode initial est pointé par l'arc n'ayant aucun nœud pour origine.

Les variables qui peuvent être observées ou estimées à partir des observations sont écrites en gras.

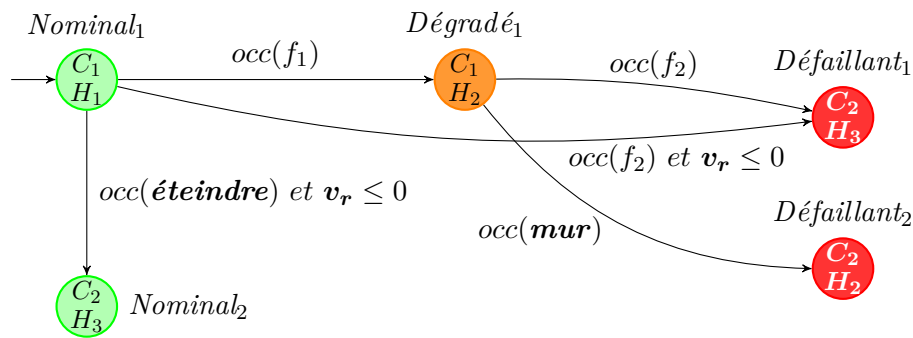


FIGURE 2.1 – Description du robot mobile.

La mission du robot est d'atteindre une zone spécifique sans rencontrer d'obstacle ni rentrer dans un mode de défaillance. Une fois cette zone atteinte, le robot est éteint. Son mode initial est *Nominal₁*. Dans ce mode, le robot se déplace dans une zone non hostile. Sa vitesse v_r , positive, peut être estimée par odométrie à partir de la dynamique continue C_1 et des observations continues issues de capteurs mesurant le déplacement des roues. Deux fautes sont anticipées, f_1 et f_2 , où f_1 représente l'entrée dans une zone hostile et f_2 , la déconnexion du moteur. La dégradation du robot est estimée avec les probabilités d'occurrences de ces fautes et la dynamique hybride H_1 , pour laquelle les probabilités augmentent simplement avec le temps.

Lorsque la commande TOR *éteindre* (événement discret et observable) se produit, le robot s'arrête et sa vitesse décroît jusqu'à atteindre la valeur 0. Le robot entre alors dans le mode *Nominal₂*, dans lequel il est éteint et où sa vitesse reste nulle (dynamique continue C_2). Puisque le robot est éteint, les probabilités d'occurrence des fautes stagnent en suivant la dynamique hybride H_3 .

La faute f_2 représente la déconnexion du moteur (débranchement de câble par exemple). Cela implique que le robot s'arrête, et donc que sa vitesse décroît jusqu'à la valeur 0. L'occurrence de f_2 mène ainsi le système dans le mode *Défaillant₁*. Une fois son moteur déconnecté, le robot a les mêmes dynamiques continue et hybride (C_2 et H_3) que s'il avait été éteint.

La faute f_1 représente l'entrée du robot dans une zone hostile. Le système rentre alors dans le mode *Dégradé₁*. Dans ce mode, le robot se déplace toujours à la vitesse v_r en suivant la dynamique continue C_1 . Cependant, les conditions physiques auxquelles il est soumis impliquent que la probabilité d'occurrence de f_2 augmente de manière plus significative que dans le mode *Nominal₁*. Ce phénomène est représenté par la dynamique hybride H_2 .

Depuis le mode *Dégradé₁*, le robot peut toujours entrer dans le mode *Défaillant₁* avec l'occurrence de la faute f_2 . Cependant, on ne fait pas coïncider ce changement de mode avec une réduction de vitesse (voir l'arc qui connecte *Dégradé₁* et *Défaillant₁*) car l'estimation de la vitesse par odométrie est moins précise dans la zone hostile.

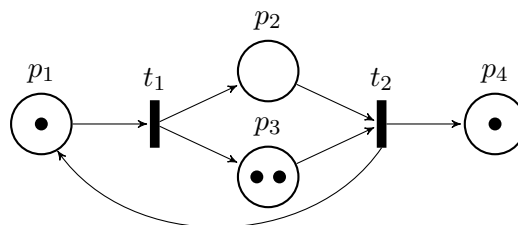
Pour finir, des obstacles sont présents dans la zone hostile. Le robot peut se retrouver bloqué par un mur (événement *mur*). Dans ce cas, la mission échoue et le robot entre dans le mode *Défaillant₂*. L'évènement *mur* n'est pas prédictible (son occurrence n'est pas estimée avec des probabilités) mais est observable grâce à un capteur environnemental TOR. Bien que la mission soit compromise et que le robot ne puisse plus bouger (C_2), son moteur est toujours allumé et donc ses lois de dégradation restent les mêmes (H_2).

2.3 Réseaux de Petri hybrides particuliers

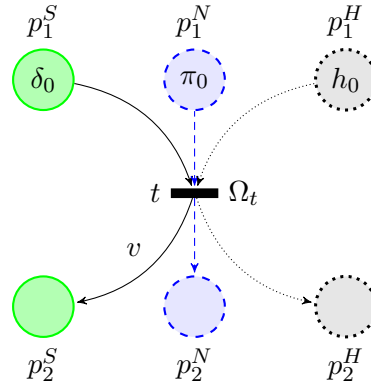
On se propose d'utiliser un réseau de Petri hybride particulière (HPPN), une extension des réseaux de Petri, pour modéliser le système.

Rappel : Réseaux de Petri

Un réseau de Petri est un graphe biparti pour lequel un nœud peut être soit une **place**, soit une **transition**. Les places et les transitions sont connectées entre elles par des **arcs** orientés mais un arc ne peut pas lier deux places ou deux transitions. Les places peuvent contenir des **jetons** et la distribution des jetons dans les places est appelée le **marquage** du réseau. Dans un réseau de Petri ordinaire, les transitions sont **tirables** (ou franchissables) si un nombre suffisant de jetons est présent dans ses places en entrée. Le **tirage** (ou franchissement) d'une transition consomme (détruit) les jetons requis et en crée de nouveaux dans ses places en sortie (s'il en existe). Graphiquement, les places sont représentées par des cercles, les transitions par des traits noirs, les arcs par des flèches et les jetons par des points noirs. Le réseau de Petri ci-dessous possède quatre places et deux transitions. La transition t_1 est tirable car la place p_1 contient au moins un jeton, alors que t_2 ne l'est pas car p_2 n'en contient pas.



La Définition 21 définit la structure complète d'un réseau de Petri hybride particulière puis la suite de la section décrit en détail chaque élément qui la compose.

FIGURE 2.2 – Exemple illustratif d'un HPPN au temps $k = 0$.

Définition 21 (HPPN). Un HPPN est un 11-uplet $\langle P, T, A, \mathcal{A}, E, X, D, \mathcal{C}, \mathcal{H}, \Omega, \mathbb{M}_0 \rangle$ réunissant les informations pour décrire des évolutions discrètes (avec les places symboliques), continues (avec les places numériques) et/ou hybrides (avec les places hybrides), et les relations des unes avec les autres (avec les transitions) :

- P est l'ensemble des places, réunissant les places symboliques P^S , les places numériques P^N et les places hybrides P^H , $P = P^S \cup P^N \cup P^H$,
- T est l'ensemble des transitions,
- $A \subset (P \times T \cup T \times P)$ est l'ensemble des arcs,
- \mathcal{A} est l'ensemble des annotations des arcs,
- E est l'ensemble des labels d'événements,
- $X \subset \mathbb{R}^{n_N}$ est l'espace d'état du vecteur d'état continu, avec $n_N \in \mathbb{N}_+$ le nombre fini de variables d'état continues,
- $D \subset \mathbb{R}^{n_H}$ est l'espace d'état du vecteur d'état hybride, avec $n_H \in \mathbb{N}_+$ le nombre fini de variables d'état hybrides,
- \mathcal{C} est l'ensemble des dynamiques continues associées aux places numériques,
- \mathcal{H} est l'ensemble des dynamiques hybrides associées aux places hybrides,
- Ω est l'ensemble des conditions associées aux transitions,
- \mathbb{M}_0 est le marquage initial du réseau.

Exemple. Une représentation graphique d'un HPPN simple est présentée en Figure 2.2.

Remarque 2.2 (Représentation d'un HPPN). Les places symboliques sont représentées par des cercles aux contours pleins, alors que les places numériques et hybrides sont représentées par des cercles discontinus et pointillés. Les transitions sont représentés par des traits noirs. Les arcs qui connectent les transitions et les places symboliques (resp. numériques et hybrides) sont représentés par des flèches pleines (resp. discontinues et pointillées). Dans l'objectif d'améliorer la lisibilité des figures, les indices de temps des jetons ne sont pas indiqués si la connaissance sur le temps courant est explicite ou si elle n'est pas nécessaire dans le contexte traité.

L'ensemble $E = E_o \cup E_{uo}$ des labels d'événements du système est l'union de l'ensemble E_o des labels des événements observables et l'ensemble E_{uo} des labels des événements non observables sur le système. Un **événement** est défini comme un couple $e = (v, k)$ où $v \in E$ est un label (ou type) d'événement et $k \in \mathbb{R}$ est la date d'occurrence de e . Par exemple, l'événement $(\text{éteindre}, 4)$ indique qu'un événement de type *éteindre* est survenu au temps 4. Un événement (v, k) est dit non observable si quel que soit k , v n'appartient pas à l'ensemble des observations discrètes du système. Considérons par exemple une faute non observable de type f_1 . L'événement $(f_1, 10)$ représente l'occurrence de f_1 au temps 10 mais il ne sera jamais observé.

Les **places** d'un HPPN sont marquées avec des **jetons**. La particularité des jetons d'un HPPN est que chacun d'eux porte une valeur.

Les **places symboliques** P^S représentent les états discrets du système et sont marquées avec des jetons symboliques, appelés **configurations**. L'ensemble des configurations au temps k est noté M_k^S . Chaque configuration du HPPN porte la trace de l'ensemble des événements qui sont survenus sur le système jusqu'au temps k et dont l'occurrence explique l'existence de la configuration. De manière plus précise, une configuration $\delta_k \in M_k^S$ est un jeton au temps k dont la valeur est l'ensemble b_k des évènements qui sont survenus jusqu'au temps k : $b_k = \{(v, \kappa) | \kappa \leq k\}$.

Les **places numériques** P^N représentent les **dynamiques continues** du système et les incertitudes associées. À chaque place numérique $p^N \in P^N$ est associé un ensemble d'équations $C_{p^N} \in \mathcal{C}$ modélisant une dynamique continue du système et les bruits associés (incertitudes sur l'évolution et sur les mesures) :

$$C_{p^N} = \begin{cases} x_{k+1} &= \mathbf{f}(x_k, u_k) + \mathbf{v}(x_k, u_k) \\ y_k &= \mathbf{h}(x_k, u_k) + \mathbf{w}(x_k, u_k) \end{cases}, \quad (2.1)$$

où $x_k \in X$ est le vecteur d'état continu, $u_k \in \mathbb{R}^{n_u}$ est le vecteur des n_u variables d'entrée continues, \mathbf{f} est la fonction d'évolution continue non bruitée, \mathbf{v} est la fonction de bruit de l'évolution continue, $y_k \in \mathbb{R}^{n_y}$ est le vecteur des n_y variables de sortie continues, \mathbf{h} est la fonction de sortie continue non bruitée, et \mathbf{w} est la fonction de bruit de la sortie continue. Les fonctions \mathbf{f} , \mathbf{v} , \mathbf{h} et \mathbf{w} sont dépendantes de la place p^N considérée. Les places numériques sont marquées avec des jetons numériques, appelés **particules**. L'ensemble des particules au temps k est noté M_k^N . Une particule $\pi_k \in M_k^N$, est un jeton dont la valeur est un état continu possible $x_k \in X$ du système au temps k .

Les **places hybrides** P^H représentent les **dynamiques hybrides** du système et les incertitudes associées. À chaque place hybride $p^H \in P^H$ est associé un ensemble d'équations $H_{p^H} \in \mathcal{H}$

modélisant une dynamique hybride du système :

$$H_{p^H} = \left\{ d_{k+1} = \mathbf{g}(d_k, b_k, x_k, u_k) + \mathbf{z}(d_k, b_k, x_k, u_k) \right\}, \quad (2.2)$$

où $d_k \in D$ est le vecteur d'état hybride, \mathbf{g} est la fonction d'évolution hybride non bruitée, et \mathbf{z} est la fonction de bruit de l'évolution hybride. Les fonctions \mathbf{g} et \mathbf{z} sont dépendantes de la place p^H considérée. Les places hybrides sont marquées avec des **jetons hybrides**. L'ensemble des jetons hybrides au temps k est noté M_k^H . Un jeton hybride $h_k \in M_k^H$ lie une configuration δ_k et une particule π_k , et sa valeur est un état hybride possible $d_k \in D$ au temps k .

L'ensemble des places du HPPN est donc :

$$\begin{aligned} P &= P^S \cup P^N \cup P^H \\ &= \{p_1^S, \dots, p_s^S\} \cup \{p_1^N, \dots, p_n^N\} \cup \{p_1^H, \dots, p_h^H\}, \end{aligned} \quad (2.3)$$

où s , n et h sont respectivement les nombres des places symboliques, numériques et hybrides.

Exemple. Dans la Figure 2.2, on a $P = \{p_1^S, p_2^S, p_1^N, p_2^N, p_1^H, p_2^H\}$.

Soit M_k l'ensemble des jetons du HPPN au temps k :

$$M_k = M_k^S \cup M_k^N \cup M_k^H, \quad (2.4)$$

où M_k^S , M_k^N et M_k^H sont respectivement les ensembles des configurations, particules, et jetons hybrides au temps k .

Exemple. Dans la Figure 2.2, on a $M_0 = \{\delta_0, \pi_0, h_0\}$.

Le marquage \mathbb{M}_k d'un HPPN au temps k est la distribution de ses jetons dans ses places :

$$\mathbb{M}_k = \mathbb{M}_k^S \cup \mathbb{M}_k^N \cup \mathbb{M}_k^H, \quad (2.5)$$

où $\mathbb{M}_k^S \in (2^{M_k^S})^s$, $\mathbb{M}_k^N \in (2^{M_k^N})^n$ et $\mathbb{M}_k^H \in (2^{M_k^H})^h$, sont respectivement les marquages symbolique, numérique et hybride au temps k .

Exemple. Dans la Figure 2.2, on a :

$$\begin{aligned} \mathbb{M}_0^S &= [[\delta_0] \ \emptyset], \\ \mathbb{M}_0^N &= [[\pi_0] \ \emptyset], \\ \mathbb{M}_0^H &= [[h_0] \ \emptyset]. \end{aligned}$$

Le marquage initial \mathbb{M}_0 d'un HPPN représente donc les conditions initiales du système (état initial et l'ensemble des événements ayant eu lieu jusqu'au temps 0).

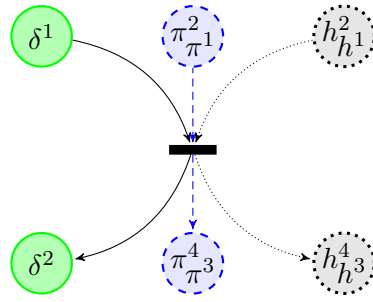


FIGURE 2.3 – Illustration des clusters de particules.

Définition 22 (Hypothèse). On définit une **hypothèse** à l'instant k comme un ensemble composé d'une configuration δ_k , de tous les jetons hybrides $\{h_k^i | i \in \{1, \dots, n_k\}\}$ liés à δ_k , et de chaque particule π_k^i liée à h_k^i à l'instant k . Un tel ensemble de jetons $\{\delta_k, \pi_k^1, \dots, \pi_k^{n_k}, h_k^1, \dots, h_k^{n_k}\}$ contient la connaissance sur l'état à l'instant k et les événements survenus sur le système.

Par exemple, si l'ensemble b_0 des événements ayant eu lieu jusqu'au temps 0 et les états continu x_0 et hybride d_0 sont initialement connus précisément, l'ensemble initial de jetons $M_0 = \{\delta_0, \pi_0, h_0\}$, où h_0 lie δ_0 et π_0 , est l'unique hypothèse initiale.

Si les conditions initiales sont ambiguës, l'ensemble initial de jetons peut être l'union de plusieurs hypothèses, e.g. $M_0 = \{\delta_0^1, \pi_0^1, h_0^1\} \cup \{\delta_0^2, \pi_0^2, h_0^2\}$, où h_0^1 lie δ_0^1 et π_0^1 , et h_0^2 lie δ_0^2 et π_0^2 .

À un instant k , une hypothèse peut aussi contenir plusieurs particules et jetons hybrides qui représentent une connaissance imprécise sur l'état continu et sur l'état hybride, e.g. $\{\delta_k^1, \pi_k^1, \dots, \pi_k^{n_k}, h_k^1, \dots, h_k^{n_k}\}$, où $n_k \in \mathbb{N}_+$ est le nombre de particules utilisées pour représenter l'état continu, et où les n_k jetons hybrides lient les n_k particules à la configuration δ_k^1 . Le nombre n_k de particules et de jetons hybrides est donc représentatif de la précision de l'hypothèse au temps k .

Définition 23 (Cluster de particules). L'ensemble des n_k particules liées à une même configuration à travers n_k jetons hybrides est appelé un **cluster**.

Exemple. Dans la Figure 2.3, h^1 et h^2 lient respectivement π^1 et π^2 à δ^1 , et, h^3 et h^4 lient respectivement π^3 et π^4 à δ^2 . Il y a donc deux hypothèses, $\{\delta^1, \pi^1, \pi^2, h^1, h^2\}$ et $\{\delta^2, \pi^3, \pi^4, h^3, h^4\}$, et deux clusters de particules, $\{\pi^1, \pi^2\}$ et $\{\pi^3, \pi^4\}$.

Propriété 2.3 (Partition de clusters de particules). À chaque instant k , l'ensemble des clusters de particules forme une partition de l'ensemble M_k^N des particules du HPPN.

Les **arcs** lient de manière directionnelle des places et des **transitions**. Les entrées d'une transition $t \in T$ sont les places desquelles part un arc pointant vers t , et les sorties de t sont les places pointées par un arc ayant pour origine t . L'ensemble des places en entrée (resp. en sortie) de t est désigné par ${}^{\circ}t$ (resp. t°). Les places en entrée (et en sortie) d'une transition peuvent être de type différents.

Exemple. C'est le cas pour la transition t de la Figure 2.2, pour laquelle ${}^{\circ}t = \{p_1^S, p_1^N, p_1^H\}$ et $t^{\circ} = \{p_2^S, p_2^N, p_2^H\}$.

Le tirage d'une transition $t \in T$ est conditionné par un jeu de conditions $\Omega_t \in \Omega$ associé à t . Le jeu Ω_t est composé d'autant de conditions qu'il y a de places en entrée de t :

$$\forall t \in T, \quad |\Omega_t| = |{}^{\circ}t|. \quad (2.6)$$

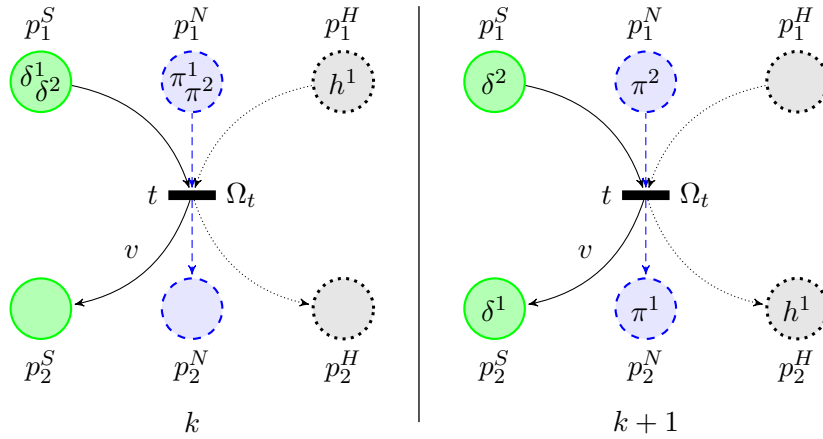
Par exemple, si t a une place de chaque type en entrée, son ensemble de conditions est $\Omega_t = \langle \omega_t^S, \omega_t^N, \omega_t^H \rangle$.

Une **condition** $\omega : M_k \rightarrow \mathbb{B}$, avec $\mathbb{B} = \{\top, \perp\}$ (l'ensemble des valeurs logiques VRAI et FAUX), peut être soit un test sur la valeur d'un jeton, soit toujours satisfait (\top), ou jamais satisfait (\perp). Une condition symbolique ω_t^S peut donc valoir \top ou \perp , ou elle peut tester l'occurrence d'un événement labellisé $v \in E$ (faute, événement de mission, interaction avec l'environnement, etc.). Dans ce dernier cas, elle prend la forme $\omega_t^S(\delta_k) = occ(b_k, v)$, qui teste si l'ensemble d'événements b_k de δ_k contient l'événement (v, k) . Avec le même raisonnement, une condition numérique ω_t^N (resp. une condition hybride ω_t^H) peut valoir \top ou \perp ou être une contrainte sur l'état continu (resp. l'état hybride) du système. Dans le dernier cas, $\omega_t^N(\pi_k) = c(x_k)$ est un test sur le vecteur d'état continu x_k de π_k .

Remarque 2.4 (Règle de franchissabilité et de franchissement des transitions). Les règles de franchissabilité et de franchissement des transitions d'un HPPN peuvent varier en fonction de son utilisation. Par exemple, dans les Chapitres 3 et 4, deux dynamiques différentes d'exécution d'un HPPN sont présentées. Ces processus d'exécution ne respectent pas les principes traditionnels des tirages (ou franchissements) des transitions. Nous proposons donc dans la suite de cette section des sémantiques de tirage par défaut pour donner une idée au lecteur de l'évolution possible du marquage d'un HPPN.

Pour éviter d'énumérer des règles de tirage pour chaque combinaison possible entre une transition et les places qui lui sont reliées, nous considérons les hypothèses suivantes :

Hypothèse 2.5 (Places en entrée d'une transition). *L'ensemble des places en entrée d'une transition est composé au minimum d'une place de n'importe quel type et au maximum d'une place de chaque type.*


 FIGURE 2.4 – Illustration du tirage d'une transition au temps k .

Hypothèse 2.6 (Places en sortie d'une transition). *L'ensemble des places en sortie d'une transition est composé d'au moins autant de places de chaque type qu'elle en possède en entrée.*

Par exemple, si une transition a une place symbolique et une place numérique en entrée, elle doit avoir au moins une place symbolique et une place numérique en sortie.

Le tirage par défaut d'une transition t au temps k est illustré dans la Figure 2.4.

On dit qu'un jeton est accepté par les conditions Ω_t au temps k s'il respecte la condition de son type. Soit $M_k(p)$ l'ensemble des jetons dans la place $p \in P$ au temps k . Nous désignons par S_k^t l'ensemble des jetons dans les places en entrée de la transition t acceptés par les conditions Ω_t au temps k . Soient $(p^S, p^N, p^H) \in (P^S \cap {}^{\circ}t) \times (P^N \cap {}^{\circ}t) \times (P^H \cap {}^{\circ}t)$ et $\omega_t^S \in \Omega_t$, $\omega_t^N \in \Omega_t$, $\omega_t^H \in \Omega_t$:

$$\begin{aligned}
 S_k^t = & \{ \delta_k \in M_k(p^S) \mid \omega_t^S(\delta_k) = \top \} \cup \\
 & \{ \pi_k \in M_k(p^N) \mid \omega_t^N(\pi_k) = \top \} \cup \\
 & \{ h_k \in M_k(p^H) \mid \omega_t^H(h_k) = \top \} .
 \end{aligned} \tag{2.7}$$

On peut alors tirer une transition tirable avec les règles suivantes.

Règle 2.7 (Transition tirable). *Une transition $t \in T$ est tirable (ou franchissable) au temps k s'il existe au moins un jeton de chacune de ses places en entrée accepté par les conditions Ω_t :*

$$\forall p \in {}^{\circ}t, \quad |S_k^t(p)| > 0. \tag{2.8}$$

Exemple. Dans la Figure 2.4, on suppose au temps k que $\omega_t^S(\delta^1) = \top$, que $\omega_t^N(\pi^1) = \top$, et que $\omega_t^H(h^1) = \top$, alors que $\omega_t^S(\delta^2) = \perp$ et que $\omega_t^N(\pi^2) = \perp$. Ainsi, la transition t est tirable au temps k car un jeton de chaque place en entrée de t est accepté.

Règle 2.8 (Tirage d'une transition). *Pour chaque type de place en entrée et en sortie de t , le tirage de $t \in T$ à l'instant k est défini par :*

$$\forall P^o \in \{P^S, P^N, P^H\}, p \in P^o \cap {}^o t, p' \in P^o \cap t^o,$$

$$\begin{aligned} M_{k+1}(p) &= M_k(p) \setminus \mathcal{S}_k^t(p), \\ M_{k+1}(p') &= M_k(p') \cup \mathcal{S}_k^t(p), \end{aligned} \tag{2.9}$$

où $\mathcal{S}_k^t(p)$ est l'ensemble des jetons de \mathcal{S}_k^t qui sont dans la place p .

Exemple. Dans la Figure 2.4, après le tirage de t , les trois jetons acceptés sont dans les places de sortie de t .

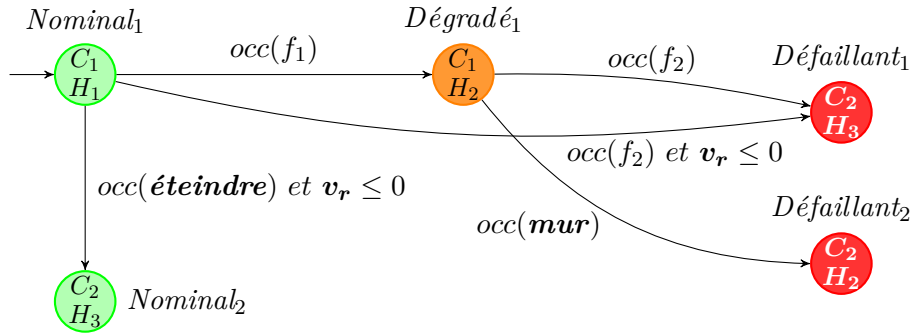
Important : Ce type de tirage constitue une différence majeure des HPPN avec les réseaux de Petri ordinaires, dans lesquels des poids sur les arcs en entrée donnent les nombres précis de jetons nécessaires au tirage d'une transition. Dans ces travaux, tous les jetons acceptés sont déplacés lors du tirage. Ce concept est également sensiblement différent des interprétations *single-server* et *infinite-server* [David & Alla 05].

Propriété 2.9 (Conservation des valeurs des jetons et liens). *Lors du tirage d'une transition, les jetons acceptés sont déplacés, leurs liens sont conservés, et leurs valeurs sont conservées ou mises à jour.*

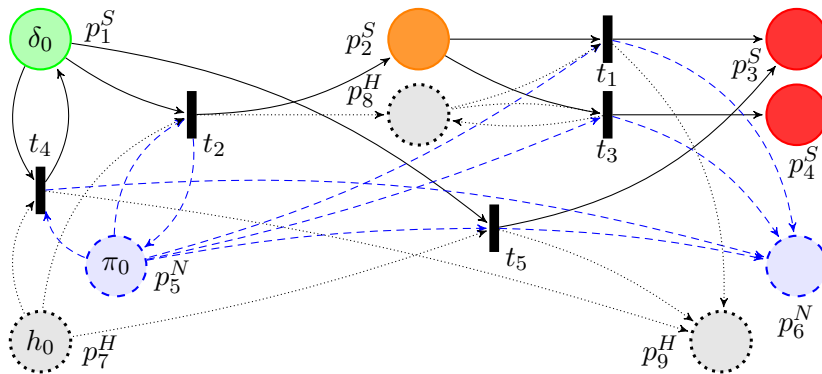
Important : La Propriété 2.9 constitue une autre différence majeure des HPPN avec les réseaux de Petri ordinaires, dans lesquels les jetons sont consommés (détruits) avant que de nouveaux jetons soient créés dans les places en sortie de la transition. La conservation des valeurs des jetons existe dans certaines extensions des réseaux de Petri, comme les réseaux de Petri colorés [David & Alla 05] par exemple. En revanche, la présence des liens entre les jetons et leur conservation lors du tirage d'une transition est spécifique aux HPPN.

Enfin, un arc $a \in A$ qui connecte une transition t à une place symbolique p^S , peut être annoté avec un label d'événement $v \in E$. Dans ce cas, une configuration δ , déplacée dans p^S après le tirage de t au temps k , voit son ensemble d'événements b mis à jour avec l'événement (v, k) . Ainsi, les **annotations** $\mathcal{A} \subset A \times E$ permettent de faire évoluer les valeurs des configurations lors du tirage des transitions.

Exemple. Dans la Figure 2.4, on suppose que h^1 lie δ^1 et π^1 au temps k . Après le tirage de t , δ^1 a pour valeur $b_{k+1}^1 = b_k^1 \cup (v, k)$, π^1 a pour valeur $x_{k+1}^1 = x_k^1$, h^1 a pour valeur $d_{k+1}^1 = d_k^1$, et h^1 lie toujours δ^1 et π^1 .



(a) Description.



(b) Modèle HPPN.

FIGURE 2.5 – Proposition de modélisation avec HPPN de l'évolution de la santé du robot mobile présenté en Section 2.2.

2.4 Modèle de santé

Avec la définition des HPPN donnée dans la Section 2.3, il est possible de modéliser un système hybride.

Chaque mode est représenté par trois places (une de chaque type). Les places symboliques représentent les différents états discrets de santé du système. Chaque dynamique continue (resp. hybride) distincte est associée à une place numérique (resp. hybride). Ainsi, une même place peut participer à la représentation de plusieurs modes. Cependant, deux modes différents ne peuvent pas être représentés avec les deux mêmes places symbolique et numérique.

Exemple du robot mobile. Un modèle HPPN du robot mobile présenté en Section 2.2 est proposé en Figure 2.5 (sa description est aussi rappelée). Les cinq modes ont été décomposés en neuf places. Deux dynamiques continues, C_1 et C_2 , sont identifiées dans la description du système en Section 2.2. Ces deux dynamiques sont donc représentées respectivement par les deux places numériques p_5^N et p_6^N . De même, les trois dynamiques hybrides identifiées, H_1 , H_2 et H_3 , sont modélisées respectivement par les trois places hybrides p_7^H , p_8^H et p_9^H . Quatre états discrets de santé du système suffisent pour modéliser la dynamique discrète du robot mobile. En effet, il n'est pas nécessaire de représenter deux états discrets différents pour les modes

$Nominal_1$ et $Nominal_2$ car ils ont deux dynamiques continues différentes. Ainsi, une seule place symbolique p_1^S représente les états nominaux. L'état dégradé est représenté par la place symbolique p_2^S . Les modes $Défaillant_1$ et $Défaillant_2$ ayant la même dynamique continue, deux places symboliques p_3^S et p_4^S sont nécessaires pour représenter les états défaillants.

Le mode initial est le mode $Nominal_1$ donc δ_0 , π_0 et h_0 sont dans les places p_1^S , p_5^N et p_7^H , respectivement. Au temps $k = 0$, on considère qu'aucun événement n'est survenu, donc $b_0 = \{\}$. La seule variable d'état estimée est la vitesse, donc $x_0 = [v_{r0}]^T$ avec $v_{r0} > 0$ car on considère que le robot est déjà en mouvement. Les probabilités initiales $\varrho_0^{f_1}$ et $\varrho_0^{f_2}$ des occurrences des fautes f_1 et f_2 sont très faibles. Ainsi, $d_0 = [\varrho_0^{f_1}, \varrho_0^{f_2}]^T$ avec $\varrho_0^{f_1} = 0.01$ et $\varrho_0^{f_2} = 0.05$.

Les transitions modélisent les changements de modes et les conditions décrivent les circonstances de ces changements. Cela implique qu'une transition $t \in T$ doit avoir une place de chaque type en entrée et une place de chaque type en sortie. Cela signifie également qu'un ensemble de trois conditions $\Omega_t = \langle \omega_t^S, \omega_t^N, \omega_t^H \rangle$ est associé à chaque transition t .

- La condition symbolique ω_t^S spécifie le label $v \in E$ de l'événement (observable ou non observable ; faute, interaction avec l'environnement ou commande discrète) qui survient au changement de mode. Si un tel label d'événement existe, l'arc a qui connecte la transition t à sa place symbolique p^S en sortie, est annoté avec v . Si aucun label d'événement n'est associé au changement de mode, la condition symbolique est mise à \top .
- La condition numérique ω_t^N spécifie la contrainte sur l'état continu qui est satisfaite au changement de mode. Si aucune condition sur l'état continu n'est à satisfaire, la condition numérique est mise à \top .
- La condition hybride ω_t^H spécifie la contrainte sur l'état hybride qui est satisfaite au changement de mode. Si aucune condition sur l'état hybride n'est à satisfaire, la condition hybride est mise à \perp .

Si ω_t^S et ω_t^N sont des conditions qui sont supposées être satisfaites en même temps, ω_t^H représente une condition alternative au changement de mode.

Important : Une interprétation généralisée est que le système change de mode si la condition logique $(\omega_t^S \wedge \omega_t^N) \vee \omega_t^H$ est satisfaite.

Dans un modèle de santé, une faute anticipée est représentée par un label d'événement $f \in E_{uo}$ non observable. Les conditions hybrides sont utilisées pour modéliser les changements de modes dûs à la dégradation du système. Par exemple, si la dégradation est modélisée par la probabilité d'occurrence d'une faute, une condition sur l'état hybride peut être une fonction booléenne qui est satisfaite si cette probabilité est supérieure à 0.9. Par exemple, dans le cas du robot mobile, on considère qu'une faute survient si sa probabilité d'occurrence atteint 0.9. Une condition sur

l'état hybride est donc une fonction booléenne qui est satisfaite si la variable du vecteur d'état hybride est supérieure à 0.9.

Exemple du robot mobile. Dans la Figure 2.5, les cinq changements de modes sont modélisés par cinq transitions. Par exemple, la transition t_4 représente le passage du mode $Nominal_1$ au mode $Nominal_2$ donc ses places en entrée sont p_1^S , p_5^N et p_7^H , et ses places en sortie sont p_1^S , p_6^N et p_9^H .

Les conditions Ω_{t_4} testent si un événement labellisé *éteindre* est survenu au temps k et si v_k a décru jusqu'à 0 :

$$\Omega_{t_4} = \langle \text{occ}(b_k, \text{éteindre}), x_k^0 \leq 0, \perp \rangle.$$

La transition t_2 représente l'occurrence de la faute f_1 , les conditions Ω_{t_2} sont donc :

$$\Omega_{t_2} = \langle \text{occ}(b_k, f_1), \top, d_k^0 > 0.9 \rangle.$$

Avec le même raisonnement, on obtient pour les transitions t_1 , t_3 et t_5 les conditions suivantes :

$$\Omega_{t_1} = \langle \text{occ}(b_k, f_2), \top, d_k^1 > 0.9 \rangle,$$

$$\Omega_{t_3} = \langle \text{occ}(b_k, \text{mur}), \top, \perp \rangle,$$

$$\Omega_{t_5} = \langle \text{occ}(b_k, f_2), x_k^0 \leq 0, d_k^1 > 0.9 \rangle.$$

2.5 Cas d'étude : les réservoirs d'eau

Le système des réservoirs d'eau est un cas d'étude simulé qui permet d'illustrer les concepts théoriques présentés dans chaque chapitre.

Dans cette partie, le système et son évolution sont décrits avant de proposer un modèle HPPN du système.

2.5.1 Description du système

Le système, présenté en Figure 2.6, est composé de trois réservoirs cylindriques connectés en série [Gaudel et al. 15a].

L'eau est versée dans le réservoir $Tank_1$ via la pompe $Pump_1$, qui délivre le flux constant q_1 . Le réservoir $Tank_3$ est le réservoir intermédiaire entre les réservoirs $Tank_1$ et $Tank_2$. Enfin, le réservoir $Tank_2$ se vide avec le flux q_2 . La mission du système est de maintenir le niveau d'eau l_2 dans le réservoir $Tank_2$ au dessus d'un niveau l_{2min} fixé.

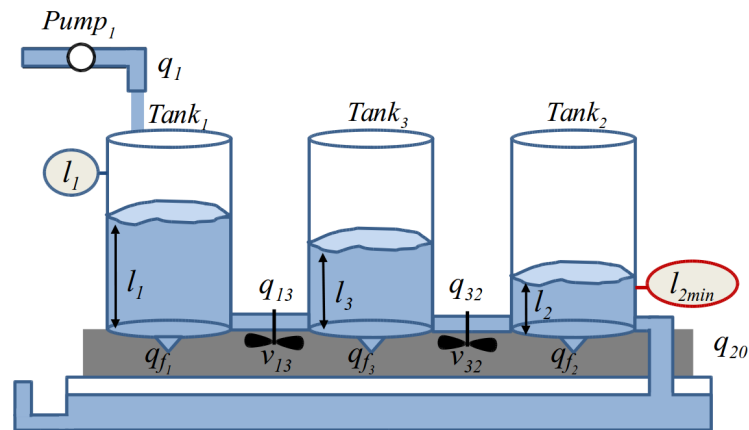


FIGURE 2.6 – Système des réservoirs d'eau.

TABLE 2.1 – Commandes, mesures et types des fautes du système des réservoirs d'eau.

Type de commande	Commentaires	Unités
Flux d'eau délivré par la pompe	Flux continu constant	$\text{m}^3 \cdot \text{s}^{-1}$
Ouverture et fermeture des vannes	Deux commandes discrètes observables par vannes	Événements
Type de mesure	Commentaires	Unités
Niveau d'eau	Mesure continue dans le réservoir $Tank_1$	m
Masse totale d'eau	Mesure continue de la masse de l'eau dans les trois réservoirs réunis	kg
Labels d'événements des fautes	Descriptions des fautes	Effets
f_0	Passage de l_2 en dessous de l_{2min}	Mène à la défaillance
f_1, f_2, f_3	Fuites d'eau dans les réservoirs	Provoquent des échappements d'eau
f_4, f_5	Blocages des vannes en position fermée	Empêchent l'eau de changer de réservoir

Les observations sur le système des réservoirs d'eau, ainsi que la liste des fautes pouvant survenir, sont présentées dans la Table 2.1.

Les seules observations sur système sont le niveau d'eau l_1 dans le réservoir $Tank_1$ et la masse totale m_{tot} de l'eau dans les trois réservoirs. Le niveau d'eau l_1 est mesuré avec une camera rendant la mesure précise à l'ordre du décimètre. La masse totale m_{tot} est mesurée avec une balance précise à l'ordre du kilogramme. Ces mesures sont relevées toutes les minutes. Les vannes v_{13} et v_{32} permettent le transfert de l'eau entre les réservoirs. Elles sont contrôlées par les actions discrètes et observables $ouvr_{v_{13}}$, $ferm_{v_{13}}$, $ouvr_{v_{32}}$ et $ferm_{v_{32}}$.

Dans sa description complète, six fautes peuvent survenir sur le système :

- f_1 , f_2 et f_3 : trois fuites, respectivement dans les réservoirs $Tank_1$, $Tank_2$ et $Tank_3$,
- f_4 et f_5 : deux blocages des vannes v_{13} et v_{32} en position fermée,
- f_0 : une faute indiquant l'échec du maintien du niveau l_2 au dessus de l_{2min} , et menant à la défaillance du système (échec de sa mission). Elle est supposée ne pouvoir survenir qu'après l'occurrence d'une autre faute.

Les fuites et les blocages des vannes peuvent survenir à n'importe quel instant mais une connaissance sur leurs origines est disponible.

L'apparition de fuites (fautes f_1 , f_2 et f_3) est fortement liée à l'usure des réservoirs. L'usure d'un réservoir n'est pas physiquement observable mais elle est causée par la pression subie par les parois. Pour chaque réservoir, quatre niveaux de stress sont identifiés :

- un stress faible, qui correspond à un fonctionnement avec une sortie d'eau possible,
- un stress moyen, qui correspond à un fonctionnement avec une entrée d'eau et une sortie d'eau impossibles,
- un stress élevé, qui correspond à un fonctionnement avec une entrée d'eau possible et une sortie d'eau impossible.

Par exemple, le niveau de stress du réservoir $Tank_1$ est faible si la vanne v_{13} est ouverte car une sortie d'eau est possible. Si v_{13} est fermée, le niveau de stress du réservoir $Tank_1$ est en revanche plus élevé car la sortie d'eau n'est plus possible et que le flux q_1 est constant (l'arrivée d'eau dans le réservoir est toujours possible).

Les blocages des vannes en position fermée (fautes f_4 et f_5) sont liés aux manipulations des vannes. Par exemple, plus la vanne v_{13} est ouverte et fermée, plus le mécanisme de la vanne se détériore rapidement et plus son blocage devient probable.

De cette connaissance sur les origines des fautes, il est possible de déduire des lois de vieillissement des composants du système qui peuvent être représentées par des lois de probabilités d'occurrences des fautes f_1 , f_2 , f_3 , f_4 et f_5 . Les évolutions de ces probabilités ont un caractère hybride car elles dépendent à la fois du temps et d'informations sur les conditions opérationnelles du système, telles que les positions des vannes ou des historiques sur leurs utilisations.

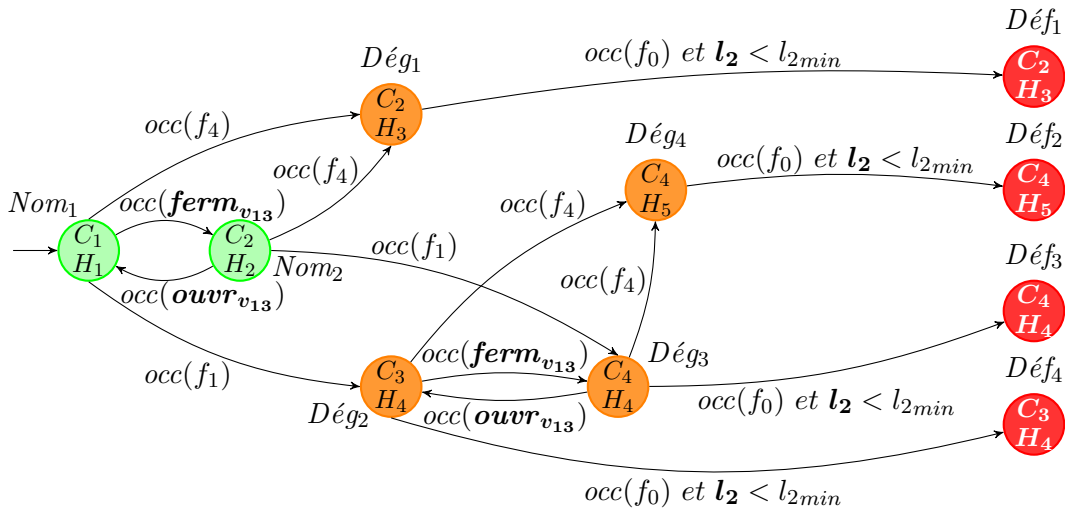


FIGURE 2.7 – Description de l'évolution de la santé du système des réservoirs d'eau.

En conséquence, ces évolutions sont considérées comme des dynamiques hybrides. Les probabilités d'occurrences des fautes f_1 , f_2 et f_3 évoluent en fonction des niveaux de stress associés à chaque réservoir et les probabilités d'occurrences des fautes f_4 et f_5 évoluent en fonction du nombre de fois que les vannes sont manipulées (ouvertes et fermées).

Pour simplifier l'étude et l'analyse des résultats, seuls les événements d'action sur la vanne v_{13} ($ouvr_{v_{13}}$ et $ferm_{v_{13}}$) et les fautes f_1 , f_4 et f_0 sont considérés dans la suite de cette étude. Une description de l'évolution de la santé du système est présentée dans la Figure 2.7.

Le système des réservoirs d'eau possède dix modes de fonctionnement. Son mode initial est Nom_1 , pour lequel les deux vannes sont ouvertes, laissant passer l'eau entre les réservoirs (dynamique continue C_1). Dans ce mode, les réservoirs ont un niveau de stress faible (dynamique hybride H_1).

Sur occurrence de la commande $ferm_{v_{13}}$, le système passe dans le mode Nom_2 , dans lequel la vanne v_{13} est en position fermée (dynamique continue C_2). Dans ce mode, les niveaux de stress des réservoirs $Tank_2$ et $Tank_3$ sont faibles mais celui du réservoir $Tank_1$ est élevé (dynamique hybride H_2).

Le système peut revenir dans le mode Nom_1 sur occurrence de la commande $ouvr_{v_{13}}$.

Depuis le mode Nom_1 (resp. Nom_2), le système peut entrer, sur occurrence de la faute f_1 , dans le mode dégradé $Dég_2$ (resp. $Dég_3$), pour lequel la dynamique continue C_3 (resp. C_4) représente l'évolution des niveaux d'eau avec une fuite dans le réservoir $Tank_1$ alors que la vanne v_{13} est ouverte (resp. fermée). Les deux modes $Dég_2$ et $Dég_3$ ont la même dynamique hybride H_4 , pour laquelle les niveaux de stress des réservoirs $Tank_2$ et $Tank_3$ sont faibles et la probabilité d'occurrence de la faute f_1 n'est plus estimée car la faute est déjà survenue.

Depuis les modes Nom_1 et Nom_2 , si la vanne v_{13} se bloque en position fermée (occurrence de la faute f_4), le système entre dans le mode dégradé $Dég_1$, dont la dynamique est la même que si la vanne avait été volontairement fermée (dynamique C_2). Dans ce mode, les niveaux de stress des réservoirs $Tank_2$ et $Tank_3$ sont faibles, celui du réservoir $Tank_1$ est élevé et la probabilité d'occurrence de la faute f_4 n'est plus estimée (dynamique hybride H_3).

De la même manière, le système peut entrer, depuis les modes $Dég_2$ et $Dég_3$, dans le mode dégradé $Dég_4$, ayant pour dynamique continue C_4 . Le mode $Dég_4$ a la dynamique hybride H_5 , pour laquelle les niveaux de stress des réservoirs $Tank_2$ et $Tank_3$ sont faibles et les probabilités d'occurrences des fautes f_1 et f_4 ne sont plus estimées.

Enfin, depuis n'importe quel mode dégradé, le système entre dans un mode de défaillance lorsque l_2 devient inférieur à l_{2min} , indicateur de l'occurrence de la faute f_0 . Le niveau l_2 n'est pas observable directement mais il peut être estimé à partir des observations continues. Vu que la faute f_0 est simplement un indicateur de défaillance et n'a aucun impact sur le comportement du système, les dynamiques continue et hybride d'un mode de défaillance sont les mêmes que celles du mode dégradé en amont.

2.5.2 Modélisation HPPN

Un modèle HPPN du système des réservoirs d'eau est proposé en Figure 2.8.

Les 10 modes ont été décomposés en 19 places (10 symboliques, 4 numériques et 5 hybrides) et 14 transitions représentent les 14 changements de mode. Pour améliorer la lisibilité, la représentation du modèle HPPN est un peu modifiée. Le nom d'une transition t et ses conditions associées Ω_t sont affichées dans le même rectangle. L'écriture des conditions est aussi simplifiée : le temps discret k n'est pas affiché et $p_0(f_i)$ représente la probabilité d'occurrence de la faute f_i . Le niveau d'eau minimum à maintenir dans le réservoir $Tank_2$ vaut $l_{2min} = 0.6$ m.

La Figure 2.9 donne la correspondance entre les modes et les places, ainsi qu'entre les conditions des changements de mode et les transitions. Elle représente le système multimode de la Figure 2.7 enrichi avec les noms des places et des transitions du modèle HPPN de la Figure 2.8.

L'état continu comprend trois variables continues représentant les niveaux l_1 , l_2 et l_3 des trois réservoirs. La Table 2.2 recense les équations d'évolutions continues non bruitées des quatre dynamiques continues C_1 , C_2 , C_3 et C_4 . La période d'échantillonnage est de 15 secondes.

Les constantes S_1 , S_2 , S_3 sont les sections des réservoirs qui valent toutes 0.0154 m². Les constantes K_{13} , K_{32} , K_{20} sont déterminées avec la section $S_n = 0.00005$ m² des tuyaux entre les réservoirs et en sortie du réservoir $Tank_2$, avec la constante gravitationnelle, ainsi qu'avec des coefficients de gain $a_{13} = 0.4753$, $a_{32} = 0.4833$ et $a_{20} = 0.1142$ de chaque tuyau.

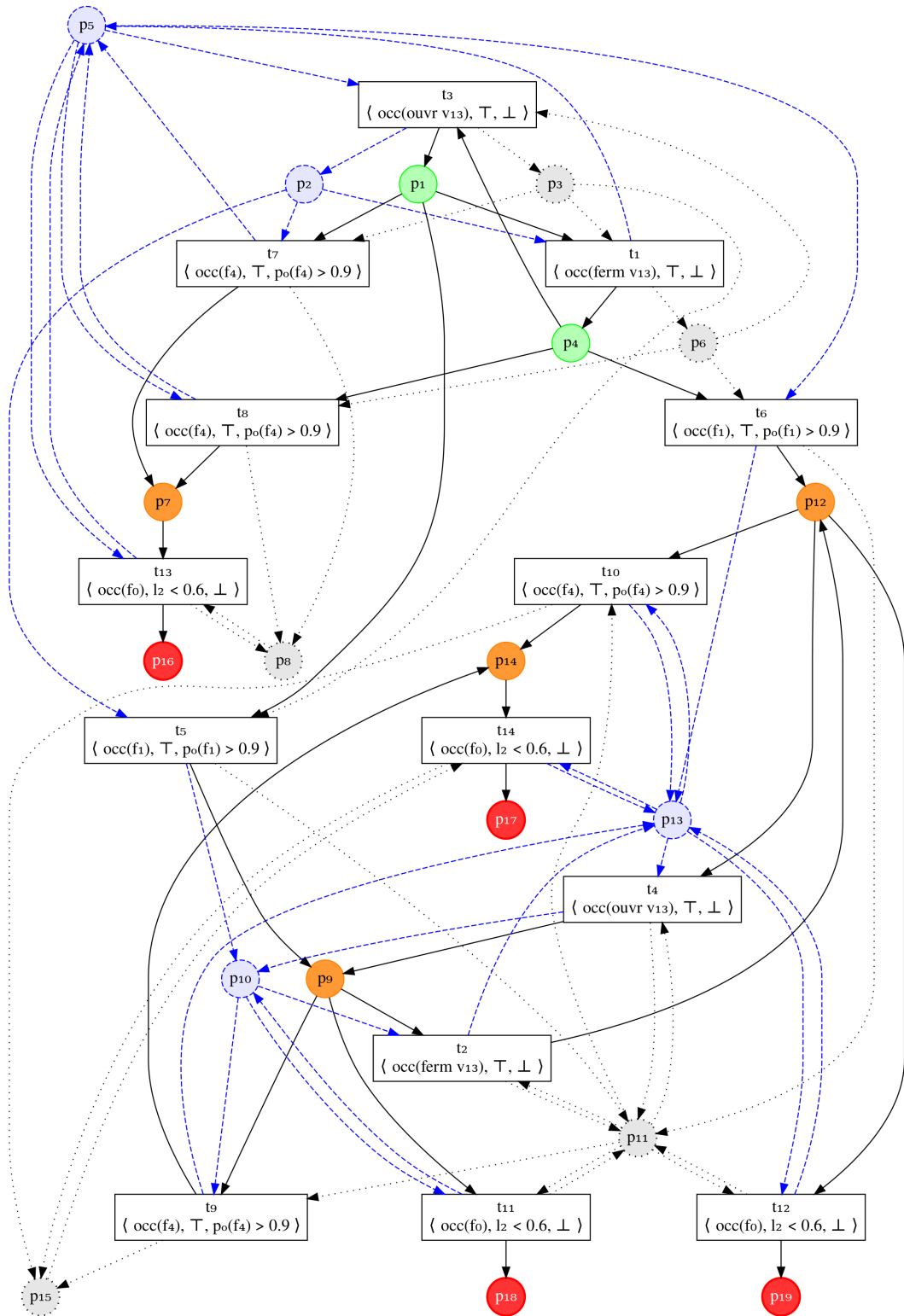


FIGURE 2.8 – Modèle HPPN de l'évolution de la santé du système des réservoirs d'eau.

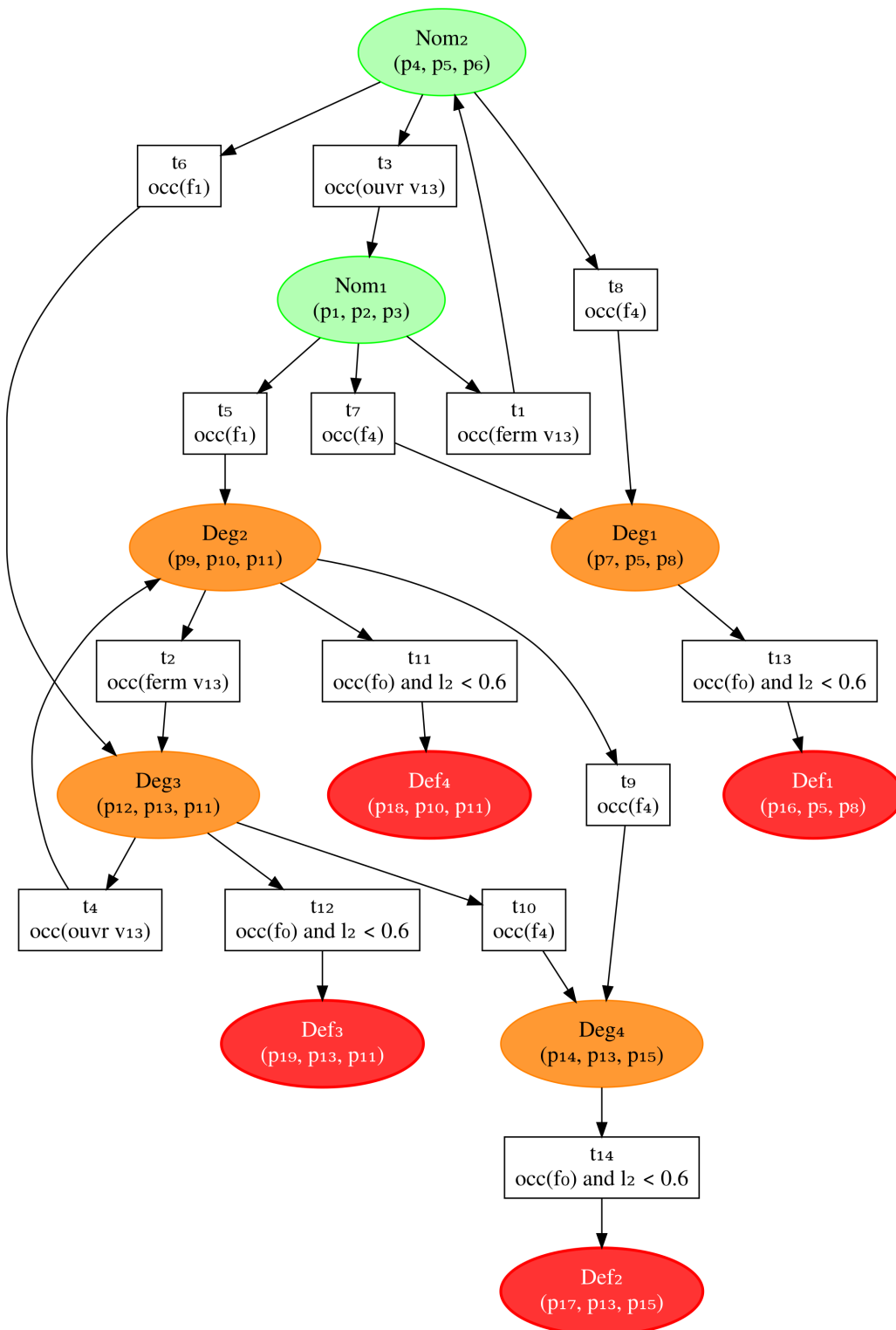


FIGURE 2.9 – Correspondance entre la description du systèmes des réservoirs d'eau en Figure 2.7 et les éléments de son modèle HPPN en Figure 2.8.

Dynamique	Commentaires	Équation d'évolution continue non bruitée
C_1	v_{13} est ouverte v_{32} est ouverte	$l_{1k+1} = l_{1k} + \frac{1}{S_1}(q_{1k} - K_{13}\sqrt{ l_{1k} - l_{3k} })$ $l_{2k+1} = l_{2k} + \frac{1}{S_2}(K_{32}\sqrt{ l_{3k} - l_{2k} } - K_{20}\sqrt{ l_{2k} })$ $l_{3k+1} = l_{3k} + \frac{1}{S_3}(K_{13}\sqrt{ l_{1k} - h_{3k} } - K_{32}\sqrt{ l_{3k} - l_{2k} })$
C_2	v_{13} est fermée v_{32} est ouverte	$l_{1k+1} = l_{1k} + \frac{1}{S_1}q_{1k}$ $l_{2k+1} = l_{2k} + \frac{1}{S_2}(K_{32}\sqrt{ l_{3k} - l_{2k} } - K_{20}\sqrt{ l_{2k} })$ $l_{3k+1} = l_{3k} + \frac{1}{S_3}(-K_{32}\sqrt{ l_{3k} - l_{2k} })$
C_3	v_{13} est ouverte v_{32} est ouverte $Tank_1$ fuit	$l_{1k+1} = l_{1k} + \frac{1}{S_1}(q_{1k} - K_{13}\sqrt{ l_{1k} - l_{3k} } - K_1\sqrt{ l_{1k} })$ $l_{2k+1} = l_{2k} + \frac{1}{S_2}(K_{32}\sqrt{ l_{3k} - l_{2k} } - K_{20}\sqrt{ l_{2k} })$ $l_{3k+1} = l_{3k} + \frac{1}{S_3}(K_{13}\sqrt{ l_{1k} - h_{3k} } - K_{32}\sqrt{ l_{3k} - l_{2k} })$
C_3	v_{13} est fermée v_{32} est ouverte $Tank_1$ fuit	$l_{1k+1} = l_{1k} + \frac{1}{S_1}(q_{1k} - K_1\sqrt{ l_{1k} })$ $l_{2k+1} = l_{2k} + \frac{1}{S_2}(K_{32}\sqrt{ l_{3k} - l_{2k} } - K_{20}\sqrt{ l_{2k} })$ $l_{3k+1} = l_{3k} + \frac{1}{S_3}(-K_{32}\sqrt{ l_{3k} - l_{2k} })$

TABLE 2.2 – Équations d'évolutions continues non bruitées du système des réservoirs d'eau.

La constante K_1 est déterminée avec la constante gravitationnelle et le coefficient de fuite $a_1 = 0.1$ du réservoir $Tank_1$ (supposé connu et constant).

Quelle que soit la dynamique continue du système, le bruit associé aux équations d'évolution continue est le même pour chaque niveau d'eau. Il est représenté par un bruit blanc suivant une loi normale d'écart type constant $\sigma = 0.002$, ce qui correspond à une incertitude sur l'évolution d'un niveau d'eau de l'ordre du centimètre par période d'échantillonnage.

L'équation de sortie continue est la même pour toutes les dynamiques continues. L'équation de la sortie continue déterminant la masse totale d'eau contenue dans les trois réservoirs est la suivante :

$$m_{totk} = \sum_{i=1}^3 S_i \times \rho_{eau} \times l_{ik}, \quad (2.10)$$

où ρ_{eau} est la densité de l'eau.

Les incertitudes sur les observations sont modélisées par des bruits blancs suivant des lois normales d'écart type $\sigma = 0.02$ pour la mesure niveau l_1 et $\sigma = 0.2$ pour celle de la masse totale d'eau m_{tot} , en accord avec la description des capteurs en Section 2.5.1.

L'état hybride comprend cinq variables continues représentant les probabilités d'occurrence des fautes f_1, f_2, f_3, f_4 et f_5 . Nous faisons le choix de représenter la distribution de probabilité d'occurrence d'une faute $f_i, \forall i \in \{1, \dots, 5\}$, avec un modèle de Weibull à 2 paramètres : la forme β_i et l'échelle η_i . En fixant $\beta_i = 1$, le modèle de Weibull décrit une faute aléatoire et est similaire à une loi exponentielle. Le second paramètre η_i est représentatif de la durée restante avant que la f_i survienne.

La forme générique de l'évolution de la probabilité d'occurrence $p_i = p_o(f_i)$ d'une faute f_i au temps k est :

$$p_{ik} = W_b(k, \eta_i), \quad (2.11)$$

où W_b est la fonction de répartition de la loi de Weibull avec $\beta = 1$:

$$W_b(k, \eta) = \int_0^k \frac{1}{\eta} e^{-\frac{k}{\eta}} dk. \quad (2.12)$$

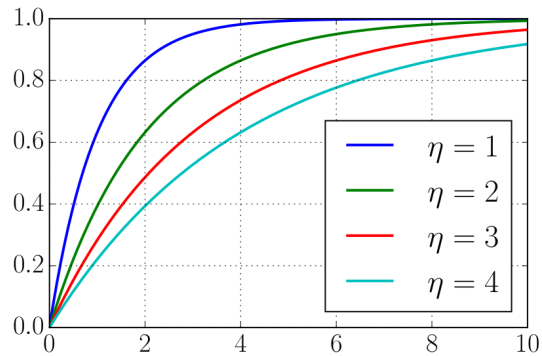


FIGURE 2.10 – Fonction de répartition de la loi de Weibull pour $\beta = 1$ et différentes valeurs de η .

De l'Équation 2.12, on déduit l'équation d'évolution hybride générique pour chaque faute f_i , $i \in \{1, \dots, 5\}$:

$$p_{ik+1} = p_{ik} - e^{\frac{-1}{\eta_{ik}}(-\eta_{ik} \ln(1 - p_{ik}) + k)} + e^{\frac{-1}{\eta_{ik}} - \eta_{ik} \ln(1 - p_{ik})}. \quad (2.13)$$

Quel que soit le mode courant, les probabilités d'occurrences des deux fautes des vannes p_4 et p_5 augmentent d'autant plus rapidement que les vannes sont manipulées. Ainsi, à chaque instant k , le paramètre d'échelle η_4 de la loi de Weibull représentant la dégradation de la vanne v_{13} (resp. η_5 pour la vanne v_{32}) dépend du coefficient α_{13} (resp. α_{32}) qui est le nombre d'occurrences des événements $ouvr_{v_{13}}$ et $ferm_{v_{13}}$ (resp. $ouvr_{v_{32}}$ et $ferm_{v_{32}}$) jusqu'à k :

$$\eta_{4k} = \frac{\eta_{13}^0}{\alpha_{13k}}, \quad (2.14)$$

$$\eta_{5k} = \frac{\eta_{32}^0}{\alpha_{32k}},$$

où η_{13}^0 (resp. η_{32}^0) est représentatif de la durée avant que la vanne v_{13} (resp. v_{32}) ne se bloque en position fermée tant qu'elle n'a pas été manipulée. En prenant $\eta_{13}^0 = \eta_{32}^0 = \eta^0$ avec $\eta^0 = 30000000$, cette durée est de l'ordre de quatre ans.

Les paramètres η_1 , η_2 et η_3 , associés aux évolutions des probabilités d'occurrence p_1 , p_2 et p_3 des trois fuites, sont constants dans le temps mais différent en fonction des niveaux de stress

Dynamique	Commentaires	Valeurs des paramètres	η_1	η_2	η_3	η_{13}^0	η_{32}^0
H_1	v_{13} est ouverte v_{32} est ouverte		η_a	η_a	η_a	η^0	η^0
H_2	v_{13} est fermée v_{32} est ouverte		η_c	η_a	η_a	η^0	η^0
H_3	v_{13} est bloquée v_{32} est ouverte		η_c	η_a	η_a	0	η^0
H_4	v_{32} est ouverte $Tank_1$ fuit		0	η_a	η_a	η^0	η^0
H_5	v_{13} est bloquée v_{32} est ouverte $Tank_1$ fuit		0	η_a	η_a	0	η^0

TABLE 2.3 – Paramètres des dynamiques hybrides du système des réservoirs d'eau.

des réservoirs dans chaque mode. On définit trois valeurs particulières du paramètre d'échelle η représentant les trois niveaux de stress des réservoirs :

- $\eta_a = 100000$: stress faible,
- $\eta_b = 80000$: stress moyen,
- $\eta_c = 60000$: stress important.

Par exemple, dans le mode Nom_1 , les paramètres η_1 , η_2 et η_3 des trois fuites sont instanciées avec la valeur η_a associée au stress le plus faible.

La Table 2.3 récapitule les valeurs des paramètres η_1 , η_2 , η_3 , η_{13}^0 et η_{32}^0 pour chaque dynamique hybride.

Les lois de probabilité étant intrinsèquement très incertaines, aucune fonction de bruit n'est ajoutée aux dynamiques hybrides.

2.6 Conclusion

Nous avons défini dans ce chapitre le cadre de modélisation basé sur les réseaux de Petri hybrides particuliers (HPPN).

Le formalisme des HPPN, proposé initialement dans [Gaudel *et al.* 14b], enrichit les réseaux de Petri avec des dynamiques continues et hybrides. Ils ont l'avantages de pouvoir exprimer à la fois des contraintes sur les systèmes à événements discrets et sur les systèmes continus mais aussi de spécifier les incertitudes sur la connaissance du système et sur les observations. La structure des HPPN et des règles de tirages ont été détaillés.

Nous les avons ensuite particulièrement utilisé pour modéliser des systèmes hybrides. La méthodologie de modélisation se base sur une description de l'évolution de la santé avec un système multimode, pour lequel à chaque mode sont associées une dynamique continue représentant le comportement du système et une dynamique hybride décrivant l'évolution de sa dégradation. Chaque mode est alors représenté par trois places dans un HPPN : une place symbolique, une place numérique, et une place hybride. L'état du système (i.e. les états discret, continu et hybride) ainsi que les événements qui sont survenus sont représentés par le marquage du HPPN. Les conditions associées aux transitions permettent de contextualiser les changements de modes du système.

Enfin, nous avons appliqué cette méthodologie à un système de réservoirs d'eau, qui servira de cas d'étude dans les Chapitres 3 et 4 pour illustrer les méthodes de diagnostic et de pronostic. Le système a été décrit et son modèle HPPN a été détaillé.

Chapitre 3

Diagnostic

Résumé

Ce chapitre présente la méthodologie de diagnostic de systèmes basée sur les HPPN.

La méthode de diagnostic a pour avantage de proposer une gestion des performances calculatoires vis-à-vis de la précision des résultats souhaitée et de différentes sources d'incertitudes. Elle repose sur la génération d'un diagnostiqueur HPPN à partir d'un modèle HPPN. Les transitions et les conditions du diagnostiqueur sont transformées pour optimiser la représentation des connaissances sur l'état du système.

Le marquage du diagnostiqueur évolue à chaque instant avec les nouvelles observations sur le système. Cette évolution se déroule en deux étapes, la prédiction et la correction, qui combinent le pseudo-tirage des transitions, des filtres particulières, et un algorithme stochastique de changement d'échelle (SSA).

Le pseudo-tirage des transitions (i.e. la duplication des jetons) permet de traiter les incertitudes symboliques, qui sont relatives au modèle discret et aux observations discrètes. Les filtres particulières gèrent les incertitudes numériques, qui concernent l'imprécision sur le modèle continu et les valeurs numériques. Enfin, l'algorithme SSA permet de réguler le nombre total de jetons en fonction du degré de pertinence des hypothèses et des trois paramètres d'échelle du diagnostiqueur.

Le diagnostic courant du système est le marquage courant du diagnostiqueur, qui porte une distribution de croyances sur les états possibles du système et les séquences de fautes menant à ces états.

Tous les concepts abordés dans ce chapitre sont appliqués sur l'exemple du robot mobile puis appliqués au système des réservoirs d'eau.

3.1 Introduction

Dans un contexte de surveillance de santé, l'objectif de la méthode de diagnostic est de suivre l'état courant du système et de déterminer les occurrences des fautes qui ont mené à cet état.

Dans ces travaux, nous proposons de générer le **diagnostiqueur** HPPN d'un système à partir de son modèle HPPN. Un diagnostiqueur est un moniteur, construit à partir d'un modèle de comportement, qui détermine le diagnostic courant du système à partir des observations [Sampath *et al.* 95].

Ici, le diagnostiqueur est un HPPN qui permet de surveiller à la fois le comportement du système et sa dégradation, tout en prenant en compte différentes sources d'incertitudes.

La Section 3.2 débute le chapitre en présentant les différentes sources d'incertitude gérées par le diagnostiqueur.

Les Sections 3.3 et 3.4 détaillent respectivement la génération du diagnostiqueur et son processus. Son processus combine l'**évolution de son marquage**, les **filtres particuliers** et un **algorithme stochastique de changement d'échelle** (*Stochastic Scaling Algorithm* ou **SSA**). À chaque instant k , le processus prédit les états possibles du système à l'instant k en se basant sur les états possibles à l'instant $k - 1$, puis utilise les observations, qui sont les entrées et sorties, discrètes (symboliques) et continues (numériques), du système au temps k , pour corriger ses prédictions. Ces deux étapes se nomment la **prédiction** et la **correction**.

Dans la Section 3.5, on définit le **diagnostic** Δ_k comme le marquage du diagnostiqueur à chaque temps k , qui porte une distribution de croyances sur les **trajectoires passées** du système, c'est-à-dire les états possibles du système et les séquences d'événements (particulièrement de fautes) menant à ces états.

La Section 3.6 détaille l'algorithme SSA, utilisé pendant l'étape de correction, qui permet de réguler le nombre total de jetons en fonction du degré de pertinence des hypothèses et des trois paramètres d'échelle du diagnostiqueur.

Une discussion des performances calculatoires du diagnostiqueur vis-à-vis de la gestion des incertitudes en Section 3.7 conclut la partie théorique de ce chapitre.

Enfin, la Section 3.8 applique la méthode de diagnostic au système des réservoirs d'eau présenté en Section 2.5.

3.2 Gestion des incertitudes

Plusieurs types d'incertitudes sont gérés par le diagnostiqueur. Une incertitude sur la connaissance du système doit être considérée car un modèle ne reflète pas parfaitement la réalité, tant

pour la partie discrète que continue. Une incertitude sur les observations doit également être prise en compte à cause de l'imprécision inhérente aux capteurs.

On distingue par la suite l'**incertitude symbolique**, relative au modèle discret et aux observations discrètes, de l'**incertitude numérique**, qui concerne l'imprécision sur le modèle continu et les valeurs numériques.

3.2.1 Incertitude symbolique

Au niveau de la connaissance du modèle discret, l'incertitude symbolique se traduit par la présence de séquences d'événements impossibles ou partiellement incorrectes. Concernant les observations discrètes, il est possible qu'un événement soit survenu mais qu'il n'ait pas été observé, on parle alors d'observation manquante. De même, il est possible d'observer un événement mais que celui n'ait pas physiquement eu lieu ; on parle alors de fausse observation.

Dans le diagnostiqueur, l'incertitude symbolique est gérée à deux niveaux :

- Toutes les transitions symboliques sont mises à VRAI durant la génération. Ainsi, plus aucune occurrence d'événement ne conditionne un changement de mode.
- Durant l'étape de prédiction du processus, le diagnostiqueur utilise le **pseudo-tirage** (tirage possibiliste) des transitions, introduit dans [Cardoso *et al.* 99], pour envisager les occurrences de chaque événement pouvant avoir lieu en accord avec la dynamique discrète. Le pseudo-tirage crée ainsi de nouvelles hypothèses.

Rappel

Une hypothèse est un ensemble de jetons composé d'une configuration, des n jetons hybrides liés à δ , et des n particules liée aux n jetons hybrides, où n est représentatif de la précision de l'hypothèse.

Dans le formalisme des HPPN, le pseudo-tirage d'une transition correspond à la duplication des jetons : les jetons dans les places en entrée de la transition ne sont pas déplacés mais sont copiés et leurs copies sont déplacées dans les places en sortie de la transition.

Règle 3.1 (Pseudo-tirage d'une transition). *Soit $t \in T$ une transition tirable. Pour chaque type de place en entrée et en sortie de t , le pseudo-tirage de $t \in T$ à l'instant $k - 1$ est défini par :*

$$\forall P^o \in \{P^S, P^N, P^H\}, p \in P^o \cap {}^o t, p' \in P^o \cap t^o,$$

$$\begin{aligned} M_k(p) &= M_{k-1}(p), \\ M_k(p') &= M_{k-1}(p') \cup S_{k-1}^t(p), \end{aligned} \tag{3.1}$$

où $S_{k-1}^t(p)$ est l'ensemble des jetons de S_{k-1}^t qui sont dans la place p .

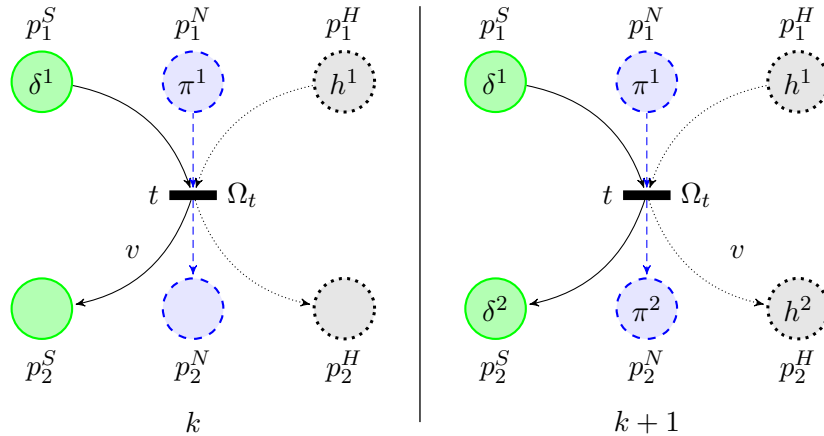


FIGURE 3.1 – Illustration du pseudo-tirage d’une transition au temps k .

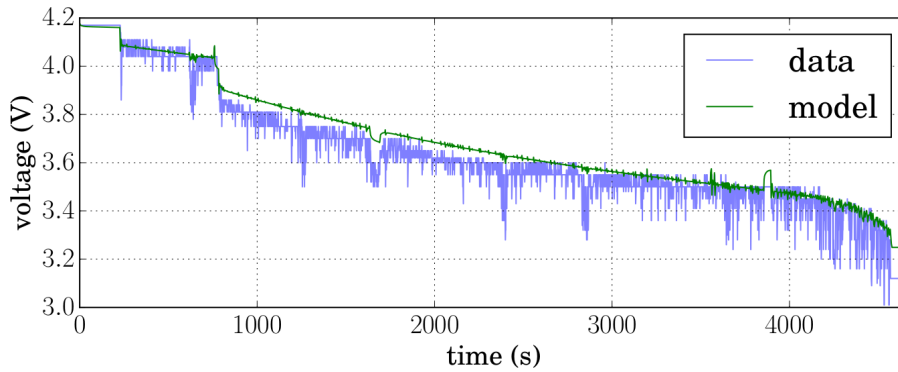


FIGURE 3.2 – Comparaison entre les données mesurées de la tension aux bornes d’une batterie et son modèle de déchargement non bruité.

Exemple. Le pseudo-tirage est illustré dans la Figure 3.1, dans lequel on suppose qu’à l’instant k , h^1 lie δ^1 et π^1 et la transition t est tirable. Après le pseudo-tirage de t , les jetons δ^1 , π^1 et h^1 n’ont pas changé de places et les jetons δ^2 , π^2 et h^2 ont été créés dans les places en sortie de t . De plus, h^1 lie toujours δ^1 et π^1 , et h^2 lie δ^2 et π^2 .

3.2.2 Incertitude numérique

Outre l’écart intrinsèque entre la réalité et le modèle continu du système, l’incertitude numérique concerne l’imprécision des valeurs numériques. C’est une problématique inévitable dans les études de cas réels. Par exemple, dans la Figure 3.2, on peut voir la différence entre les données mesurées de la tension d’une batterie et son modèle de déchargement non bruité.

L’incertitude numérique est souvent gérée avec un **estimateur** (ou observateur) [Ding 14], dont l’objectif est d’estimer l’état continu du système tout en tenant compte des bruits associés au modèle et aux mesures. Dans ces travaux, nous avons choisi d’utiliser des **filtres**

particulaires [van der Merwe *et al.* 00]. Dans notre contexte, le but d'un filtre particulaire est d'estimer l'état continu du système en se basant seulement sur les données observées.

Rappel : Filtre particulaire

Un filtre particulaire est un processus bayésien guidé par les données qui estime l'état continu d'un système sous la forme une distribution de **particules**. C'est une méthode de Monte Carlo (qui utilise une source de hasard) et séquentielle (fondée sur la simulation). Elle repose sur l'enchaînement répété de deux étapes : la prédiction et la correction. L'étape de prédiction applique l'équation d'évolution d'un modèle continu sur les particules pour prédire les états possibles à l'instant suivant. L'étape de correction utilise la fonction de sortie du modèle continu pour évaluer la cohérence de chaque particule vis-à-vis des nouvelles observations sur le système et recréer une nouvelle distribution de particule représentant l'estimation de l'état.

L'utilisation des filtres particuliers est pertinente pour l'estimation à la fois des états discret, continu et hybride car la représentation de l'estimation de l'état continu est déjà discrétisée en particules.

Par exemple dans ces travaux, un filtre particulaire est appliqué indépendamment à chaque cluster de particules grâce aux liens entre les configurations et les particules, assurés par les jetons hybrides.

Rappel : Clusters de particules

L'ensemble des particules liées à une même configuration à travers des jetons hybrides est appelé un cluster.

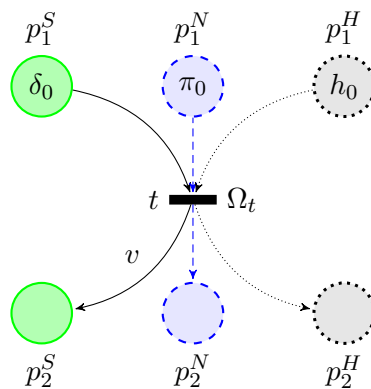
Durant l'étape de prédiction du diagnostiqueur, les valeurs des particules évoluent en fonction des dynamiques continues associées aux places numériques auxquelles les particules appartiennent. Ensuite, durant l'étape de correction, chaque cluster de particules est rééchantillonné indépendamment.

3.3 Génération du diagnostiqueur

Cette section détaille les étapes de la génération d'un diagnostiqueur $HPPN_{\Delta}$ à partir d'un modèle $HPPN_{\Phi}$.

Rappel : Structure d'un modèle $HPPN_{\Phi}$

$HPPN_{\Phi}$ est le modèle d'un système hybride. Chaque mode du système est représenté par une place symbolique, une place numérique et une place hybride, mais une place peut participer à la représentation de plusieurs modes. Les transitions modélisent les changements de modes, donc chaque transition t a trois places en entrée (une de chaque type) et trois places en sortie (une de chaque type également). En conséquence, son ensemble de conditions associées est composé d'une condition symbolique, d'une condition numérique, et d'une condition hybride : $\Omega_t = \langle \omega_t^S, \omega_t^N, \omega_t^H \rangle$. Une interprétation généralisée est que le système change de mode si la condition logique $(\omega_t^S \wedge \omega_t^N) \vee \omega_t^H$ est satisfaite.



Définition 24 (Diagnosticteur HPPN). Soit un modèle HPPN défini par le 11-uplet $HPPN_{\Phi} = \langle P_{\Phi}, T_{\Phi}, A_{\Phi}, \mathcal{A}_{\Phi}, E_{\Phi}, X_{\Phi}, D_{\Phi}, \mathcal{C}_{\Phi}, \mathcal{H}_{\Phi}, \Omega_{\Phi}, \mathbb{M}_{0\Phi} \rangle$ comme présenté en Section 2.3. Le diagnosticteur $HPPN_{\Delta}$ de $HPPN_{\Phi}$ est un 11-uplet généré à partir de $HPPN_{\Phi}$:

$$HPPN_{\Delta} = \langle P_{\Delta}, T_{\Delta}, A_{\Delta}, \mathcal{A}_{\Delta}, E_{\Delta}, X_{\Delta}, D_{\Delta}, \mathcal{C}_{\Delta}, \mathcal{H}_{\Delta}, \Omega_{\Delta}, \mathbb{M}_{0\Delta} \rangle. \quad (3.2)$$

Les éléments de $HPPN_{\Delta}$ sont déterminés avec les étapes suivantes :

- Copie du modèle, détaillée en Section 3.3.1,
- Séparation en niveau de comportement et niveau hybride, détaillée en Section 3.3.2,
- Modification des conditions symboliques, détaillée en Section 3.3.3,
- Suppression des conditions hybrides, détaillée en Section 3.3.4,
- Fusion des transitions, détaillée en Section 3.3.5,
- Suppression des transitions bouclant sur une même place hybride, détaillée en Section 3.3.6.

La suite de cette section détaille les différentes étapes de la génération du diagnosticteur.

3.3.1 Copie du modèle

La première étape est la copie du modèle :

$$HPPN_{\Delta} \leftarrow HPPN_{\Phi}. \quad (3.3)$$

Le rôle du diagnostiqueur est de surveiller les états discret, continu et hybride du système. De fait, les espaces d'états discret, continu et hybride, ainsi que les dynamiques continues et hybrides sont les mêmes que ceux du modèle. Il en résulte que l'ensemble des places, des labels d'événements, des espaces d'états et les dynamiques du diagnostiqueur restent les mêmes que ceux du modèle $HPPN_{\Phi}$:

- $P_{\Delta} = P_{\Phi}$,
- $E_{\Delta} = E_{\Phi}$,
- $X_{\Delta} = X_{\Phi}$,
- $D_{\Delta} = D_{\Phi}$,
- $\mathcal{C}_{\Delta} = \mathcal{C}_{\Phi}$,
- $\mathcal{H}_{\Delta} = \mathcal{H}_{\Phi}$.

Le marquage initial $\mathbb{M}_{0\Delta}$ du diagnostiqueur correspond au marquage initial $\mathbb{M}_{0\Phi}$ du modèle, qui contient la connaissance sur le mode, l'état et les événements survenus sur le système au temps 0. On a donc $\mathbb{M}_{0\Delta} = \mathbb{M}_{0\Phi}$.

3.3.2 Séparation en niveau de comportement et niveau hybride

La seconde étape consiste à séparer le diagnostiqueur en deux niveaux. Chaque transition $t \in T_{\Delta}$ est transformée en un couple de nouvelles transitions (t', t'') . La transition t' hérite des arcs liant t aux places symboliques et numériques, ainsi que des conditions symbolique et numérique. La transition t'' hérite des arcs liant t aux places hybrides, ainsi que de la condition hybride. On crée ainsi t' et t'' telles que :

$$\begin{aligned} {}^{\circ}t' &= {}^{\circ}t \cap (P^S \cup P^N), \\ t'^{\circ} &= t^{\circ} \cap (P^S \cup P^N), \end{aligned} \quad (3.4)$$

et :

$$\begin{aligned} {}^{\circ}t'' &= {}^{\circ}t \cap P^H, \\ t''^{\circ} &= t^{\circ} \cap P^H, \end{aligned} \quad (3.5)$$

avec les conditions $\Omega_{t'}$ et $\Omega_{t''}$ suivantes :

$$\begin{aligned} \Omega_{t'} &= \langle \omega_t^S, \omega_t^N \rangle, \\ \Omega_{t''} &= \langle \omega_t^H \rangle. \end{aligned} \quad (3.6)$$

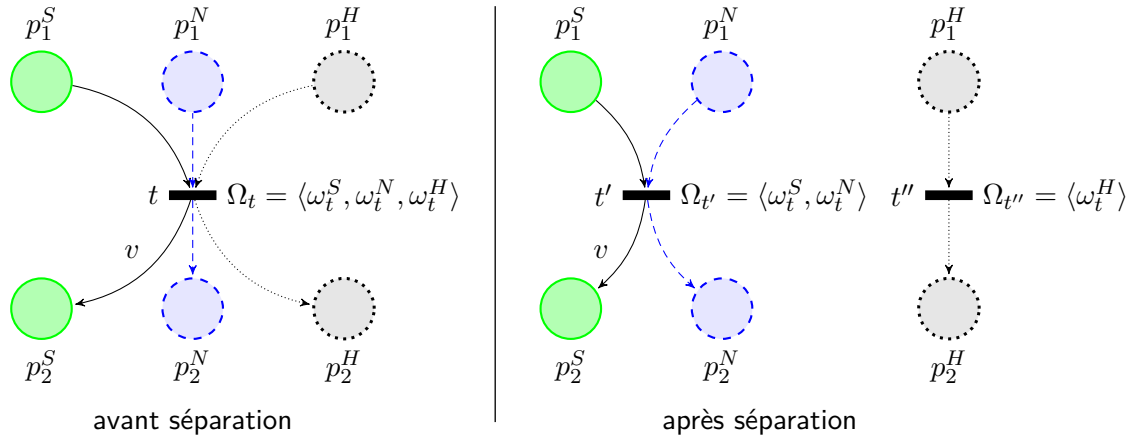


FIGURE 3.3 – Illustration de l'étape de séparation en deux niveaux du diagnostiqueur sur un HPPN simple.

Toutes les nouvelles transitions ainsi créées forment T_Δ , désigné simplement T dans la suite de la section.

Exemple. La Figure 3.3 illustre la séparation sur un HPPN simple ayant une seule transition.

Exemple du robot mobile. La Figure 3.4 illustre l'étape de séparation en deux niveaux du diagnostiqueur du robot mobile présenté en Section 2.2. Les places hybrides et les transitions associées sont isolées dans le niveau hybride.

3.3.3 Modification des conditions symboliques

La troisième étape est la mise à VRAI de toutes les conditions symboliques, en accord avec la gestion des incertitudes sur les occurrences des événements (voir Section 3.2.1) :

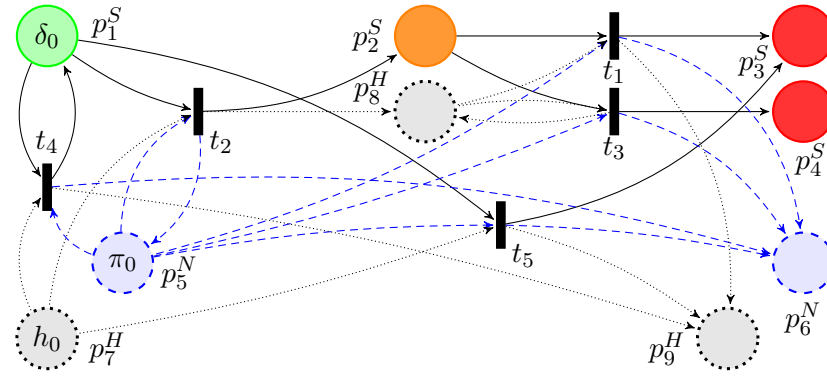
$$\forall t \in T_\Delta, \omega_t^S \in \Omega_t \Rightarrow \omega_t^S \leftarrow \top. \quad (3.7)$$

Ainsi, toutes les configurations du diagnostiqueur satisfont les conditions. Cela signifie qu'à chaque instant, le diagnostiqueur considère l'occurrence de chaque événement pouvant avoir lieu depuis le mode courant. Les annotations des arcs restent inchangées pour enregistrer l'ensemble des événements simulés dans les ensembles d'événements des configurations :

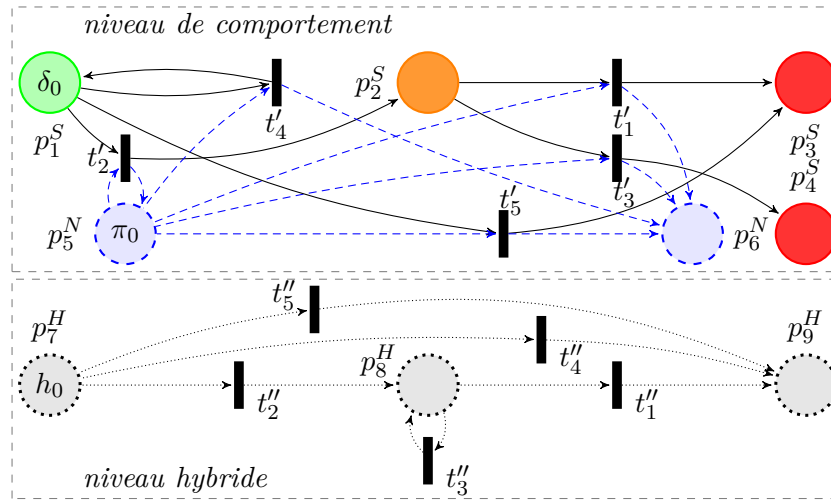
$$\mathcal{A}_\Delta = \mathcal{A}_\Phi. \quad (3.8)$$

Exemple du robot mobile. Dans l'exemple du robot mobile, les conditions associées à la transition t_2 sont :

$$\Omega_{t_2} = \langle \text{occ}(b_k, f_1), \top, d_k^0 > 0.9 \rangle.$$



(a) Avant la séparation.



(b) Après la séparation.

FIGURE 3.4 – Illustration de l'étape de séparation en deux niveaux du diagnostiqueur HPPN du robot mobile.

En conséquence, les conditions associées à la transition t'_2 sont :

$$\Omega_{t'_2} = \langle \top, \top \rangle,$$

un ensemble de conditions toujours satisfaites quels que soient δ et π , car $\Omega_{t'_2}$ ne comprend aucun test sur l'état continu. En suivant le même raisonnement, on obtient pour les conditions des autres transitions :

$$\begin{aligned} \Omega_{t'_1} &= \langle \top, \top \rangle, \\ \Omega_{t'_3} &= \langle \top, \top \rangle, \\ \Omega_{t'_4} &= \langle \top, x_k^0 \leq 0 \rangle, \\ \Omega_{t'_5} &= \langle \top, x_k^0 \leq 0 \rangle. \end{aligned}$$

3.3.4 Suppression des conditions hybrides

La quatrième étape consiste à supprimer les conditions hybrides pour que l'évolution du marquage du niveau hybride ne dépendent pas de l'état hybride :

$$\forall t \in T_{\Delta}, \omega_t^H \in \Omega_t \Rightarrow \Omega_t \leftarrow \Omega_t \setminus \{\omega_t^H\}. \quad (3.9)$$

3.3.5 Fusion des transitions

De manière à réduire la taille de l'espace des états possibles, on fusionne les transitions ayant le même ensemble de places en entrée et la même place numérique en sortie. Ces fusions sont l'objectif de la cinquième étape.

Au niveau de comportement, cela permet de créer, pendant l'étape de prédiction, des hypothèses partageant le même cluster de particules. En d'autres termes, il devient possible de surveiller plusieurs hypothèses suivant la même dynamique continue avec un unique cluster de particules au lieu d'avoir autant de clusters que d'hypothèses.

Au niveau hybride, la fusion permet de supprimer les transitions concurrentes, c'est-à-dire les transitions ayant la même place hybride en entrée et la même place hybride en sortie.

De manière plus formelle, il faut d'abord identifier les transitions qui peuvent être fusionnées. Deux transitions sont fusionnables si elles représentent le même changement de dynamique continue (mêmes places numériques en entrée et en sortie, et même condition numérique) et qu'elles ont la même place symbolique en entrée.

Définition 25 (Transitions fusionnables). Deux transitions $(t', t'') \in T^2$ sont fusionnables si et seulement si :

$$({}^{\circ}t' = {}^{\circ}t'') \wedge (t'^{\circ} \cap P^N = t''^{\circ} \cap P^H) \wedge (t'^{\circ} \cap P^H = t''^{\circ} \cap P^N) \wedge (\Omega_{t'} = \Omega_{t''}), \quad (3.10)$$

où ${}^{\circ}t$ (resp. t°) désigne l'ensemble des places en entrée (resp. en sortie) d'une transition $t \in T$.

La Définition 25 s'applique parfaitement aux transitions du niveau de comportement et à celles du niveau hybride. Pour les transitions du niveau de comportement, les intersections avec les places hybrides P^H sont toujours vides, et pour celles du niveau hybride, ceux sont les intersections avec les places numériques P^N qui sont toujours vides.

L'étape de fusion consiste à fusionner toutes les transitions fusionnables tant qu'il y a au moins deux transitions fusionnables en utilisant la définition suivante.

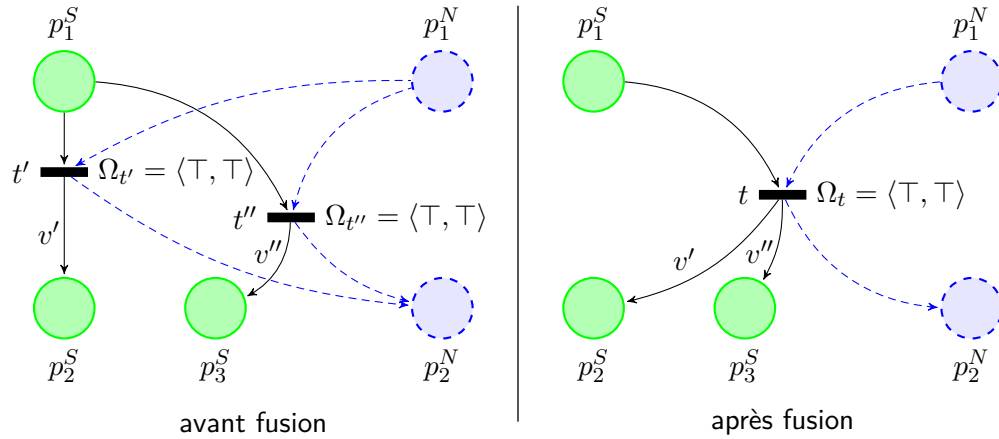


FIGURE 3.5 – Illustration de l'étape de fusion des transitions du diagnostiqueur sur un HPPN simple.

Définition 26 (Fusion de deux transitions). La fusion de deux transitions fusionnables $(t', t'') \in (T)^2$ est définie par deux étapes :

- (1) Création d'une nouvelle transition t pour laquelle :

$$\begin{aligned} \circ t &\leftarrow \circ t', \\ t^\circ &\leftarrow t'^\circ \cup t''^\circ, \\ \Omega_t &\leftarrow \Omega_{t'}. \end{aligned} \tag{3.11}$$

- (2) Introduction de t dans T et suppression de t' et t'' :

$$T \leftarrow (T \setminus \{t', t''\}) \cup \{t\}. \tag{3.12}$$

Exemple. La Figure 3.5 illustre l'étape de fusion sur un HPPN simple ayant deux transitions. Les transitions t' et t'' sont fusionnables car elles ont le même ensemble de places en entrée $\{p_1^S, p_1^N\}$, la même place numérique en sortie p_2^N , et les mêmes conditions.

Exemple du robot mobile. La Figure 3.6 illustre l'étape de fusion du diagnostiqueur du robot mobile. Pour simplifier la lecture, les transitions ont été renommées par rapport à la Figure 3.4 selon la table de correspondance suivante.

Figure 3.4	t'_1	t'_2	t'_3	t'_4	t'_5	t''_1	t''_2	t''_3	t''_4	t''_5
Figure 3.6	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}

Dans le niveau de comportement, les transitions t_1 et t_3 (resp. t_4 et t_5) ont été fusionnées en t_{13} (resp. t_{45}) puisqu'elles étaient associées à la même condition, qu'elles avaient le même ensemble de places en entrées $\{p_2^S, p_5^N\}$ (resp. $\{p_1^S, p_5^N\}$) et la même place numérique p_6^N en

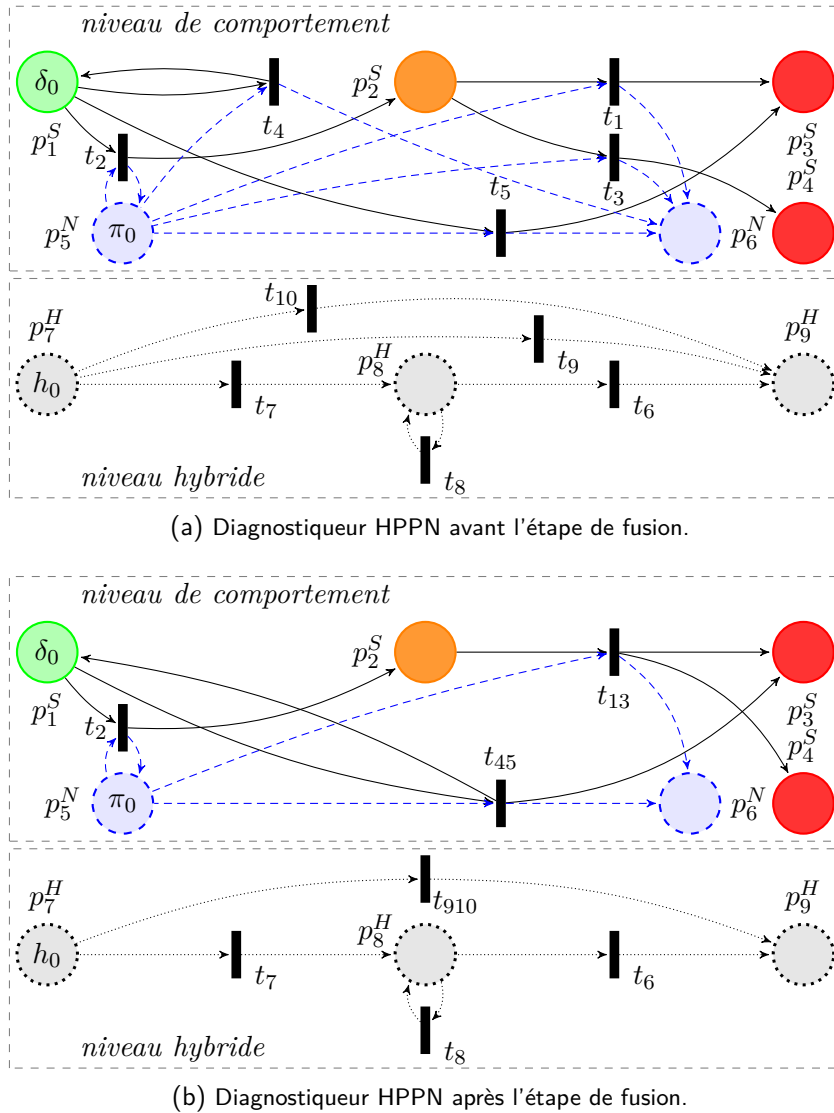


FIGURE 3.6 – Illustration de l'étape de fusion des transitions du diagnostiqueur HPPN du robot mobile.

sortie. Dans le niveau hybride, les transitions t_9 et t_{10} ont été fusionnées en t_{910} puisqu'elles étaient concurrentes.

3.3.6 Suppression des transitions bouclant sur une même place hybride

L'objectif de cette dernière étape est d'éviter le déplacement des jetons hybrides à travers une transition qui boucle sur une même place hybride. Dans un tel cas, la transition et la place hybride forment une boucle élémentaire.

Le déplacement d'un jeton hybride qui ne provoque pas de changement de place n'a aucun impact sur le suivi de la dégradation. En conséquence, les transitions bouclant sur la même

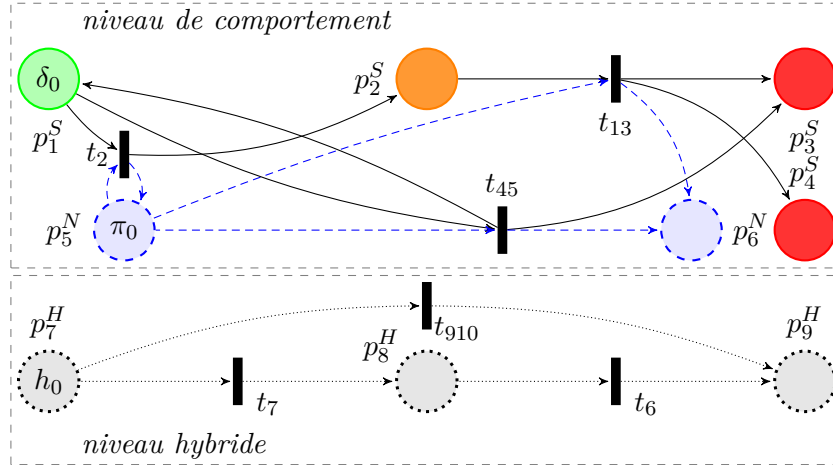


FIGURE 3.7 – Diagnostiqueur HPPN du robot présenté en Section 2.2.

place hybride sont supprimées :

$$T \leftarrow T \setminus \{t \mid t \cap P^H = t \cap P^H\}. \quad (3.13)$$

Cette étape supprime donc toutes les boucles élémentaires du niveau hybride.

Propriété 3.2 (Pureté du niveau hybride). *Le niveau hybride est un réseau de Petri structuellement pur (qui ne comporte pas de boucles élémentaires).*

Exemple du robot mobile. La Figure 3.7 présente le diagnostiqueur HPPN du robot mobile à la fin de sa génération. La transition t_8 a été supprimée car elle bouclait sur la place hybride p_8^H .

3.4 Processus du diagnostiqueur

À partir du marquage initial et des commandes initiales, le marquage du diagnostiqueur évolue à chaque nouvel ensemble d'observations pour déterminer le diagnostic courant. Cette évolution se déroule en deux étapes, la prédiction et la correction, qui combinent le pseudo-tirage des transitions, les filtres particulières et l'algorithme SSA.

La Figure 3.8 illustre comment, à chaque temps k , l'enchaînement de ces deux étapes détermine le marquage \mathbb{M}_k du diagnostiqueur en raisonnant sur les observations (entrées et sorties) du système :

$$O_k = (U_k^S, u_k^N, Y_k^S, y_k^N), \quad (3.14)$$

où $U_k^S \subset E_\Delta$ est l'ensemble des entrées discrètes, $u_k^N \in \mathbb{R}^{n_u}$ est le vecteur des entrées continues, $Y_k^S \subset E_\Delta$ est l'ensemble des sorties discrètes, et $y_k^N \in \mathbb{R}^{n_y}$ est le vecteur des sorties

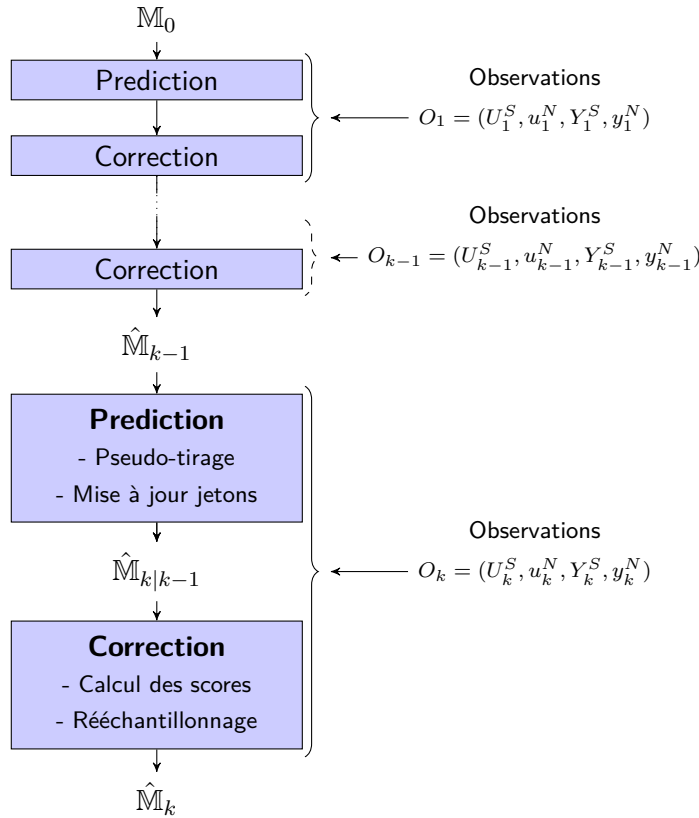


FIGURE 3.8 – Processus du diagnostiqueur HPPN.

continues au temps k . En faisant l’hypothèse qu’un seul événement peut avoir lieu durant un échantillon de temps, l’union $U_k^S \cup Y_k^S$ peut soit être vide, soit contenir un unique label d’événement $v \in E_\Delta$. Cette hypothèse est cohérente à temps discret, la période d’échantillonnage sera adaptée si besoin.

La suite de cette section présente le détail des étapes de prédiction et de correction.

3.4.1 Prédiction

L’étape de prédiction a pour objectif de déterminer le marquage $\hat{M}_{k|k-1}$ du diagnostiqueur, contenant les états prédits à l’instant k , en connaissant les états possibles au temps $k - 1$. Elle est basée sur le pseudo-tirage des transitions et sur la mise à jour des valeurs des jetons, qui simulent l’occurrence de n’importe quel événement à l’instant $k - 1$ puis l’évolution des états hybride et continu en utilisant u_{k-1}^N , le vecteur des entrées continues à l’instant $k - 1$.

3.4.1.1 Pseudo-tirage des transitions

Toutes les transitions tirables sont tirées simultanément selon des règles similaires à celles présentées dans la Section 2.3, aux différences près que l’ensemble S_{k-1}^t des jetons acceptés

par une transition $t \in T$ est déterminé de manière à respecter la condition logique $\omega_t^S \wedge \omega_t^N$ et que les jetons sont dupliqués et non déplacés.

Notons $M_{k-1}(p)$ l'ensemble des jetons dans la place $p \in P$ au temps $k - 1$.

Soient $(p^S, p^N, p^H) \in (P^S \cap {}^\circ t) \times (P^N \cap {}^\circ t) \times (P^H \cap {}^\circ t)$ et $\omega_t^S \in \Omega_t$, $\omega_t^N \in \Omega_t$, $\omega_t^H \in \Omega_t$, l'ensemble \mathcal{S}_{k-1}^t est formellement défini par :

$$\mathcal{S}_{k-1}^t = \{(\delta_{k-1}, \pi_{k-1}) \mid (\delta_{k-1}, \pi_{k-1}, h_{k-1}) \in (M_{k-1}(p^S) \times M_{k-1}(p^N) \times M_{k-1}(p^H)), \\ \text{lien}(\delta_{k-1}, \pi_{k-1}, h_{k-1}) \wedge (\omega_t^S(\delta_{k-1}) \wedge \omega_t^N(\pi_{k-1}))\} \quad (3.15)$$

si t est une transition du niveau de comportement, et :

$$\mathcal{S}_{k-1}^t = \{h_{k-1} \mid (\delta_{k-1}, \pi_{k-1}, h_{k-1}) \in (M_{k-1}(p^S) \times M_{k-1}(p^N) \times M_{k-1}(p^H)), \\ \text{lien}(\delta_{k-1}, \pi_{k-1}, h_{k-1}) \wedge (\omega_t^S(\delta_{k-1}) \wedge \omega_t^N(\pi_{k-1}))\} \quad (3.16)$$

si t est une transition du niveau hybride.

Dans les Équations 3.15 et 3.16, la fonction booléenne $\text{lien} : M(p^S) \times M(p^N) \times M(p^H) \rightarrow \mathbb{B}$ renvoie VRAI si h_{k-1} lie δ_{k-1} et π_{k-1} , et FAUX sinon.

On peut alors pseudo-tirer une transition tirable avec les règles suivantes.

Règle 3.3 (Transition tirable). *Une transition $t \in T$ est tirable (ou franchissable) au temps $k - 1$ s'il existe au moins un jeton accepté par les conditions Ω_t :*

$$|\mathcal{S}_{k-1}^t| > 0. \quad (3.17)$$

Règle 3.4 (Pseudo-tirage d'une transition). *Soit $t \in T$ une transition tirable. Pour chaque type de place en entrée et en sortie de t , le pseudo-tirage de $t \in T$ à l'instant $k - 1$ est défini par :*

$$\forall P^o \in \{P^S, P^N, P^H\}, p \in P^o \cap {}^\circ t, p' \in P^o \cap t^\circ,$$

$$\begin{aligned} M_k(p) &= M_{k-1}(p), \\ M_k(p') &= M_{k-1}(p') \cup \mathcal{S}_{k-1}^t(p), \end{aligned} \quad (3.18)$$

où $\mathcal{S}_{k-1}^t(p)$ est l'ensemble des jetons de \mathcal{S}_{k-1}^t qui sont dans la place p .

3.4.1.2 Mise à jour des valeurs des jetons

Pendant le tirage, les valeurs des configurations sont mises à jour avec les annotations des arcs \mathcal{A}_Δ avec la règle suivante.

Règle 3.5 (Mise à jour de la valeur d'une configuration). *L'ensemble d'événements b_{k-1} d'une configuration δ_{k-1} déplacée à travers un arc a pendant le tirage des transitions, est mis à jour avec le label d'événement $v \in E$ annotant a ; l'événement $(v, k-1)$ lui est ajouté :*

$$b_k \leftarrow b_{k-1} \cup \{(v, k-1)\}. \quad (3.19)$$

Propriété 3.6. *L'ensemble d'événements b_k d'une configuration ne contient que des événements survenus jusqu'à $k-1$, car un événement survenant à l'instant $k-1$ change l'état discret du système à l'instant k .*

Après le tirage des transitions, les valeurs des jetons hybrides puis celles des particules sont mises à jour en fonction des places auxquelles appartiennent les jetons et des entrées continues à l'instant $k-1$.

Règle 3.7 (Mise à jour de la valeur d'un jeton hybride). *Après le tirage des transitions, le vecteur d'état hybride d_{k-1} d'un jeton hybride est mis à jour grâce à l'équation d'évolution associée à la place hybride p^H à laquelle le jeton hybride appartient (voir Équation 2.1), en utilisant les valeurs de la configuration et de la particule qu'il lie :*

$$d_k = \mathbf{g}(d_{k-1}, b_k, x_{k-1}, u_{k-1}^N) + \mathbf{z}(d_{k-1}, b_k, x_{k-1}, u_{k-1}^N), \quad (3.20)$$

où u_{k-1}^N est le vecteur des entrées continues à l'instant $k-1$, \mathbf{g} est la fonction d'évolution hybride non bruitée et \mathbf{z} est la fonction de bruit de l'évolution hybride.

Utiliser b_k n'est pas un problème car l'ensemble b_k ne contient que des événements survenant strictement avant k (voir Propriété 3.6).

Règle 3.8 (Mise à jour de la valeur d'une particule). *Après le tirage des transitions, le vecteur d'état numérique x_{k-1} d'une particule est mis à jour avec l'équation d'évolution associée à la place numérique p^N à laquelle la particule appartient (voir Équation 2.2) :*

$$x_k = \mathbf{f}(x_{k-1}, u_{k-1}^N) + \mathbf{v}(x_{k-1}, u_{k-1}^N), \quad (3.21)$$

où u_{k-1}^N est le vecteur des entrées continues à l'instant $k-1$, \mathbf{f} est la fonction d'évolution continue non bruitée et \mathbf{v} est la fonction de bruit de l'évolution continue.

Exemple du robot mobile. La Figure 3.9 illustre l'étape de prédiction du diagnostiqueur du robot mobile à partir de son marquage $\hat{\mathbb{M}}_{k-1}$ à l'instant $k-1$. Pour alléger la lecture, nous considérons qu'il y a une unique hypothèse $\{\delta^1, \pi^1, \pi^2, h^1, h^2\}$ dans le marquage $\hat{\mathbb{M}}_{k-1}$. La place symbolique p_1^S est vide, donc les transitions t_2 et t_{45} ne sont pas tirables. La transition t_{13} est tirable car les δ^1 , π^1 et π^2 appartiennent à la même hypothèse.

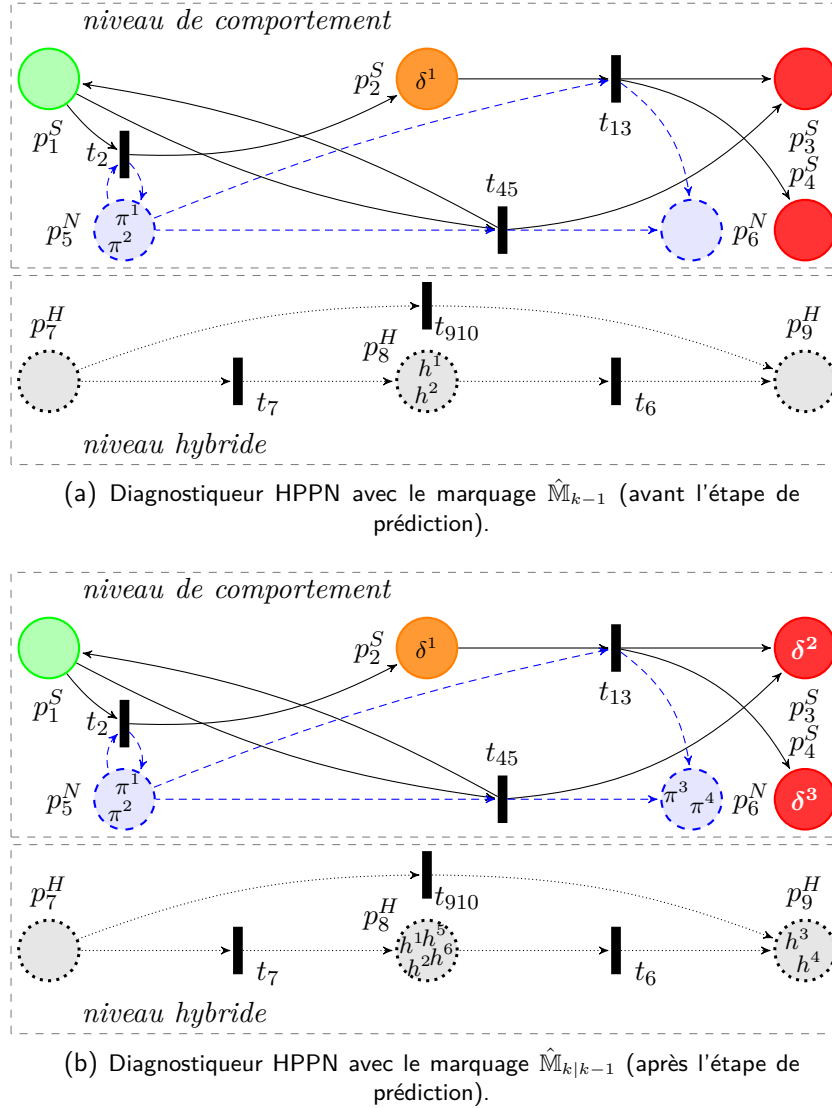


FIGURE 3.9 – Illustration de l'étape de prédiction du processus du diagnostiqueur HPPN du robot mobile.

Après l'étape de prédiction, le marquage est $\hat{M}_{k|k-1}$. L'ensemble des jetons du réseau inclut trois hypothèses : $\{\delta^1, \pi^1, \pi^2, h^1, h^2\}$, qui représentent le fait que le système n'a pas changé de mode, et $\{\delta^2, \pi^3, \pi^4, h^3, h^4\}$ et $\{\delta^3, \pi^3, \pi^4, h^5, h^6\}$, qui représentent le fait que le système est soit passé dans le mode *Défaillant*₁, soit passé dans le mode *Défaillant*₂. Comme ces deux modes ont la même dynamique continue et qu'ils partagent la même place numérique, les deux hypothèses partagent le même cluster de particules $\{\pi^3, \pi^4\}$.

3.4.2 Correction

L'étape de correction détermine le marquage estimé $\hat{M}_{k|k}$ à partir du marquage prédit $\hat{M}_{k|k-1}$ (voir Figure 3.8).

Soit $\hat{M}_{k|k-1}$ l'ensemble des jetons inclus dans le marquage $\hat{M}_{k|k-1}$. L'étape de correction est basée sur le calcul d'un score pour chaque hypothèse de $\hat{M}_{k|k-1}$ en fonction des observations et sur le rééchantillonnage des jetons en fonction des scores.

L'idée est de calculer un poids pour chaque configuration et pour chaque cluster de particules. Ces poids sont ensuite combinés pour donner le score de chaque hypothèse. Lors du rééchantillonnage, les hypothèses ayant les scores les plus grands ont tendance à être conservées alors que les hypothèses ayant les scores plus faibles ont tendance à être écartées. Les détails de ces calculs font l'objet des sous-sections suivantes.

3.4.2.1 Calcul du score d'une hypothèse

Les scores des hypothèses sont calculés avec les poids des configurations et des particules.

Définition 27 (Poids d'une configuration). Soit O_k^{S-} l'ensemble des événements ayant été observés jusqu'au temps k :

$$O_k^{S-} = \{(v, \kappa) | v \in (U_\kappa^S \cup Y_\kappa^S), \kappa \leq k\}. \quad (3.22)$$

Le poids $w_{\delta_k} \in \mathbb{R}_+$ d'une configuration δ_k est évalué grâce à une fonction w^S qui donne le degré de pertinence d'une configuration en comparant son ensemble d'événements b_k à l'ensemble O_k^{S-} :

$$w_{\delta_k} = w^S(b_k, O_k^{S-}). \quad (3.23)$$

Plus b_k est "semblable" à O_k^{S-} , plus la configuration aura un degré de pertinence élevé.

Cette évaluation peut être plus ou moins complexe en fonction des connaissances sur le système. Il est possible par exemple de considérer la probabilité qu'un événement ait bien eu lieu s'il a été observé, ou encore d'augmenter significativement le poids si une séquence d'événements particulière est reconnue. La fonction d'évaluation du poids d'une configuration dépend donc du modèle et pourra être définie par un expert.

Si aucune fonction d'évaluation n'est connue pour le système étudié, le poids d'une configuration δ_k est évalué par défaut comme la fonction gaussienne, d'espérance 0 et d'écart type 1, de la distance entre son ensemble d'événements b_k et O_{k-1}^{S-} :

$$w_{\delta_k} = g_{(0,1)}(\text{dist}(b_k, O_{k-1}^{S-})), \quad (3.24)$$

où dist est le cardinal de la différence symétrique entre deux ensembles A et B :

$$\text{dist}(A, B) = |(A \setminus B) \cup (B \setminus A)|, \quad (3.25)$$

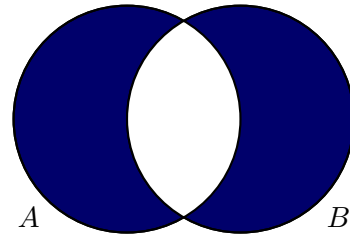


FIGURE 3.10 – En foncé : différence symétrique de deux ensembles A et B , i.e. $(A \setminus B) \cup (B \setminus A)$.

et g la fonction gaussienne, avec $\mu = 0$ et $\sigma = 1$:

$$g_{(0,1)}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \quad (3.26)$$

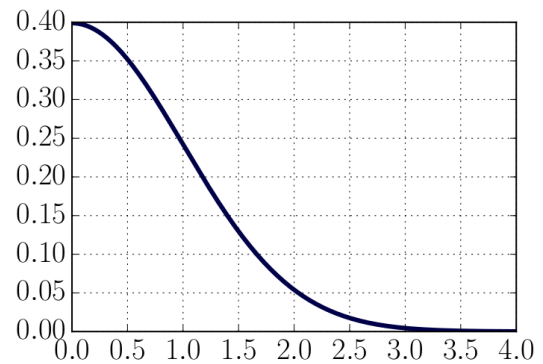


FIGURE 3.11 – Fonction gaussienne pour $\mu = 0$ et $\sigma = 1$.

Cette évaluation est naïve mais peu coûteuse. Elle accorde un poids plus important aux configurations qui prennent en compte le moins d'événements non observés. Notons que l'événement discret observé au temps k n'est pas utilisé dans l'Équation 3.24, car les valeurs des configurations contiennent uniquement des événements survenus jusqu'au temps $k - 1$ (voir Propriété 3.6). Ceci est logique car les événements survenus à l'instant k n'ont pas encore de conséquence.

Définition 28 (Poids d'une particule). Soit \hat{y}_k l'estimation de la sortie de la particule π_k à l'instant k :

$$\hat{y}_k = \mathbf{h}(x_k, u_k^N), \quad (3.27)$$

où \mathbf{h} est la fonction de sortie non bruitée associée à la place à laquelle π_k appartient (voir Équation 2.2), et u_k^N le vecteur des entrées continues au temps k .

Le poids $w_{\pi_k} \in \mathbb{R}_+$ de la particule π_k est évalué en comparant l'estimation de sa sortie \hat{y}_k avec y_k^N , le vecteur des sorties continues au temps k . Cette évaluation dépend

de la manière dont est modélisé le bruit sur les observations continues (fonction w de l'Équation 2.2) et peut varier d'une dynamique continue à l'autre.

Comme l'évaluation du poids d'une configuration, l'évaluation du poids d'une particule dépend du modèle et pourra donc être définie par un expert.

Par défaut, on se propose de modéliser le bruit sur les observations continues par un bruit gaussien. Le poids d'une particule π_k est alors la valeur de la gaussienne d'espérance μ_k et d'écart type σ_k , deux paramètres représentatifs du bruit à l'instant k , appliquée à la différence $(\hat{y}_k - y_k^N)$:

$$w_{\pi_k} = g_{(\mu_k, \sigma_k)}(\hat{y}_k - y_k^N), \quad (3.28)$$

où g la fonction gaussienne :

$$g_{(\mu, \sigma)}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x - \mu)^2}{2\sigma^2}}. \quad (3.29)$$

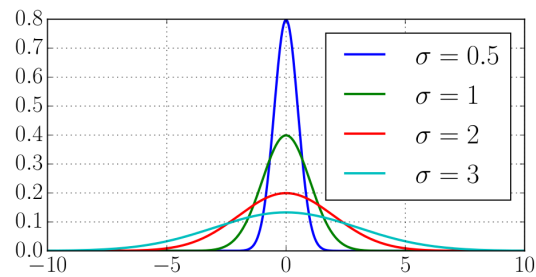


FIGURE 3.12 – Fonction gaussienne pour $\mu = 0$ et différentes valeurs de σ .

À partir des poids des particules calculés avec la Définition 28, on peut calculer les poids des clusters de particules.

Dans la suite de la section, l'indice k des objets manipulés au temps k est omis pour alléger la lecture.

Soit ζ la partition (définie dans la Propriété 2.3) de l'ensemble de toutes les particules $\hat{M}_{k|k-1}^N$. La partition ζ contient tous les clusters de particules :

$$\zeta = \left\{ \left\{ \pi^i, i \in \{1, \dots, n\} \right\} \mid \exists \delta, \exists \{h^j, j \in \{1, \dots, n\}\} \right. \\ \left. \text{tels que les } n \text{ jetons hybrides } \{h^j\} \text{ lient } \delta \text{ aux } n \text{ particules } \{\pi^i\} \right\} \quad (3.30)$$

Par définition, $\bigcup_{\varsigma \in \zeta} \varsigma = \hat{M}_{k|k-1}^N$ et $\forall (\varsigma, \varsigma') \in \zeta^2, \varsigma \neq \varsigma', \varsigma \cap \varsigma' = \emptyset$.

Définition 29 (Poids d'un cluster de particules). Le poids d'un cluster de particules $\varsigma \in \zeta$ est la moyenne des poids des particules du cluster :

$$w_{\varsigma} = \frac{1}{|\varsigma|} \sum_{\pi \in \varsigma} w_{\pi}. \quad (3.31)$$

En utilisant la moyenne, et non la somme par exemple, on empêche le nombre de particules du cluster d'influencer son poids.

Enfin, le score de chaque hypothèse est calculé avec les poids normalisés de sa configuration et de son cluster de particules.

La normalisation des poids des configurations (resp. clusters de particules) permet de relativiser les séquences d'événements (resp. estimations de l'état continu) possibles les unes par rapport aux autres en fonction de leurs cohérences.

- Les poids des configurations sont normalisés avec la somme des poids des autres configurations de $\hat{M}_{k|k-1}^S$:

$$\forall \delta \in \hat{M}_{k|k-1}^S, \quad \tilde{w}_{\delta} = \frac{w_{\delta}}{\sum_{\delta \in \hat{M}_{k|k-1}^S} w_{\delta}}. \quad (3.32)$$

- Les poids des clusters sont normalisés avec la somme des poids de tous les clusters de particules de ζ .

$$\forall \varsigma \in \zeta, \quad \tilde{w}_{\varsigma} = \frac{w_{\varsigma}}{\sum_{\varsigma \in \zeta} w_{\varsigma}}. \quad (3.33)$$

Définition 30 (Score d'une hypothèse). Soit une hypothèse représentée par la configuration δ , les n particules $\{\pi^1, \dots, \pi^n\}$ et les n jetons hybrides $\{h^1, \dots, h^n\}$ qui les lient. Le score de l'hypothèse est la somme pondérée des poids normalisés de δ et du cluster de particules $\varsigma = \{\pi^1, \dots, \pi^n\}$:

$$Score(\{\delta, \pi^1, \dots, \pi^n, h^1, \dots, h^n\}) = \alpha \times \tilde{w}_{\delta} + (1 - \alpha) \times \tilde{w}_{\varsigma}, \quad (3.34)$$

où $\alpha \in [0, 1]$ est un paramètre du diagnostiqueur indiquant la confiance globale de la partie discrète du modèle par rapport à la partie continue.

Propriété 3.9 (Bornes des scores des hypothèses). *Le score d'une hypothèse est toujours compris entre 0 et 1.*

3.4.2.2 Rééchantillonnage des jetons

L'algorithme SSA (détaillé en Section 3.6) affecte une nouvelle précision n_k^ζ à chaque cluster de particules $\varsigma \in \zeta$ au temps k en fonction du meilleur des scores des hypothèses qu'il représente, mais aussi de trois paramètres d'échelle du diagnostiqueur, ρ_Δ^{min} , ρ_Δ^{max} et ρ_Δ^{tot} .

Rappel : Précision d'un cluster et d'une hypothèse

La précision d'un cluster de particules est son cardinal. Par extension, la précision d'une hypothèse est son nombre de particules, qui est également son nombre de jetons hybrides.

Le paramètre ρ_Δ^{min} est le nombre minimum de particules que l'on s'accorde pour surveiller une hypothèse. Les hypothèses de scores les plus faibles posséderont après le rééchantillonnage un nombre de particules proche (dans la limite supérieure) de ρ_Δ^{min} ou ne seront pas rééchantillonnées.

Le paramètre ρ_Δ^{max} est le nombre suffisant de particules pour surveiller une hypothèse. Les hypothèses de scores les plus élevés posséderont après le rééchantillonnage un nombre de particules proche (dans la limite inférieure) de ρ_Δ^{max} .

Propriété 3.10. *La précision d'un cluster est toujours comprise entre ρ_Δ^{min} et ρ_Δ^{max} :*

$$\forall \varsigma \in \zeta, \quad \rho_\Delta^{min} \leq n_k^\zeta \leq \rho_\Delta^{max}. \quad (3.35)$$

Le paramètre ρ_Δ^{tot} est le nombre maximum de particules (ou de jetons hybrides) que l'on s'accorde pour surveiller l'ensemble des hypothèses.

Propriété 3.11. *La somme des précisions est toujours inférieure ou égale à ρ_Δ^{tot} :*

$$\sum_{\varsigma \in \zeta} n_k^\zeta \leq \rho_\Delta^{tot}. \quad (3.36)$$

Ici, l'algorithme SSA détermine la nouvelle précision de chaque cluster car un cluster peut représenter l'état continu de plusieurs hypothèses. On lui associe ainsi le meilleur score des hypothèses qu'il représente pour assurer que l'hypothèse ayant le score le plus élevé aura une précision en accord avec son degré de pertinence.

Exemple. Dans la Figure 3.13, le cluster $\{\pi^1, \pi^2, \pi^3\}$ représente en même temps l'hypothèse $\{\delta^1, \pi^1, \pi^2, \pi^3, h^1, h^2, h^3\}$ ayant un score de 0.1 et l'hypothèse $\{\delta^2, \pi^1, \pi^2, \pi^3, h^4, h^5, h^6\}$ ayant un score de 0.8. On associera le score 0.8 au cluster pour que l'algorithme SSA accorde à la seconde hypothèse une précision représentative de son score. On choisira de garder un score de 0.8 (et non de 0.1) pour le cluster $\{\pi^1, \pi^2, \pi^3\}$ en entrée de l'algorithme SSA afin de conserver la précision de l'hypothèse ayant le score le plus élevé.

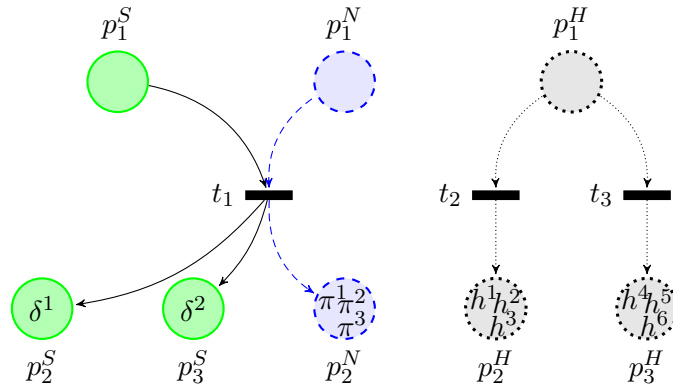


FIGURE 3.13 – Illustration de deux hypothèses partageant le même cluster de particules.

Enfin, chaque cluster de particules $\varsigma \in \zeta$ est localement rééchantillonné avec ses n_k^ς particules par un rééchantillonnage par importance (*Sampling Importance Resampling* ou SIR) [van der Merwe *et al.* 00]. Ce rééchantillonnage local choisit, suivant une loi normale, une ou plusieurs fois les particules avec les poids les plus importants, et délaisse celles avec les poids les plus faibles. Les jetons hybrides liés aux particules dupliquées sont également dupliqués, alors que ceux liés à des particules détruites sont détruits. Enfin, les configurations isolées, c'est-à-dire celles qui ne sont plus liées à aucun jeton hybride, sont détruites. Le marquage du diagnostiqueur est alors $\mathbb{M}_k = \hat{\mathbb{M}}_{k|k}$.

Remarque 3.12 (Caractère borné du diagnostiqueur). Après les étapes de prédiction et de correction au temps k , le nombre de jetons dans le diagnostiqueur est forcément inférieur ou égal à $3 \times \rho_\Delta^{tot}$. En effet, l'algorithme SSA rééchantillonne au plus ρ_Δ^{tot} particules et la situation la plus complexe est qu'il rééchantillonne ρ_Δ^{tot} hypothèses ayant chacune un seul jeton de chaque type.

Propriété 3.13 (Caractère borné du diagnostiqueur). *Le diagnostiqueur HPPN $_\Delta$ est $3 \times \rho_\Delta^{tot}$ -borné.*

Ce mécanisme de correction met en évidence le fait que les liens entre les jetons (introduits par les jetons hybrides et illustrés par la notion d'hypothèse) empêchent la distribution de particules d'être déséquilibrée par le pseudo-tirage des jetons. Dans un filtre particulaire, le nombre de particules définit la précision du filtre mais est aussi un facteur de performance calculatoire. Les paramètres ρ_Δ^{min} et ρ_Δ^{max} sont respectivement les précisions minimale et suffisante accordées au diagnostiqueur pour surveiller une hypothèse. Les paramètres d'échelle du diagnostiqueur instaurent donc un compromis entre le nombre d'hypothèses à surveiller, la précision accordée à chacune, et la puissance de calcul disponible (ρ_Δ^{tot} peut être choisi pour répondre à des contraintes de performance de calcul).

Exemple du robot mobile. La Figure 3.14 illustre l'étape de correction du diagnostiqueur du robot mobile à partir de son marquage $\hat{\mathbb{M}}_{k|k-1}$. Les particules π^1 et π^2 prédisent que le

robot a une vitesse non nulle car elles suivent la dynamique C_1 (le robot est en marche). Les particules π^3 et π^4 , quant à elles, prédisent que le robot a une vitesse nulle car elles suivent la dynamique C_2 (le robot est à l'arrêt).

Supposons maintenant que les poids des jetons sont calculés avec les fonctions par défaut. Avec les observations continues aux temps k , on accorde des poids très faibles aux particules π^1 et π^2 et des poids très élevés aux particules π^3 et π^4 . Aucun événement n'est observé au temps k , donc δ^1 obtient un poids sensiblement plus important que δ^2 et δ^3 . Enfin, on choisit les paramètres d'échelle du diagnostiqueur : $(\rho^{min}, \rho^{max}, \rho^{tot}) = (1, 2, 3)$. En conséquence, l'algorithme SSA, lors de l'étape de correction, accorde une précision de 1 au cluster $\{\pi^1, \pi^2\}$, et une précision de 2 au cluster $\{\pi^3, \pi^4\}$. Dans le cluster $\{\pi^1, \pi^2\}$, uniquement π^1 est rééchantillonné, et dans le cluster $\{\pi^3, \pi^4\}$, π^4 est rééchantillonné deux fois, formant le nouveau cluster $\{\pi^4, \pi^{4'}\}$. Au niveau hybride, h^2 , h^3 et h^5 sont supprimés et $h^{4'}$ et $h^{6'}$ sont créés. Après l'étape de correction, le marquage à l'instant k est $\hat{\mathbb{M}}_{k|k} = \mathbb{M}_k$.

3.5 Diagnostic

Le diagnostic Δ_k au temps k du système est le marquage du diagnostiqueur au temps k :

$$\Delta_k \triangleq \mathbb{M}_k. \quad (3.37)$$

Le marquage \mathbb{M}_k porte plusieurs informations :

- à travers la position des jetons dans les places et leur poids, il représente une distribution de croyances sur l'état courant du système.
- à travers l'information portée par les configurations (les ensembles d'événements b_k), il donne une représentation de la trajectoire pour atteindre l'état courant.

Il indique donc les croyances sur les états discret, continu et hybride, et sur les événements qui sont survenus (dont les fautes).

Exemple du robot mobile. Après l'étape de correction (voir Figure 3.14 (B)), on obtient le diagnostic au temps k :

$$\Delta_k = [\emptyset \quad [\delta^1] \quad [\delta^2] \quad [\delta^3] \quad [\pi^1] \quad [\pi^4, \pi^{4'}] \quad \emptyset \quad [h^1, h^6, h^{6'}] \quad [h^4, h^{4'}]],$$

qui contient trois hypothèses.

La première hypothèse, $\{\delta^1, \pi^1, h^1\}$, de faible score, considère que la faute f_1 est survenue et que le système a toujours une vitesse non nulle. Les jetons sont dans les places p_2^S , p_5^N et p_8^H , ce qui permet de déduire que le système est dans le mode $Dégradé_1$ au temps k .

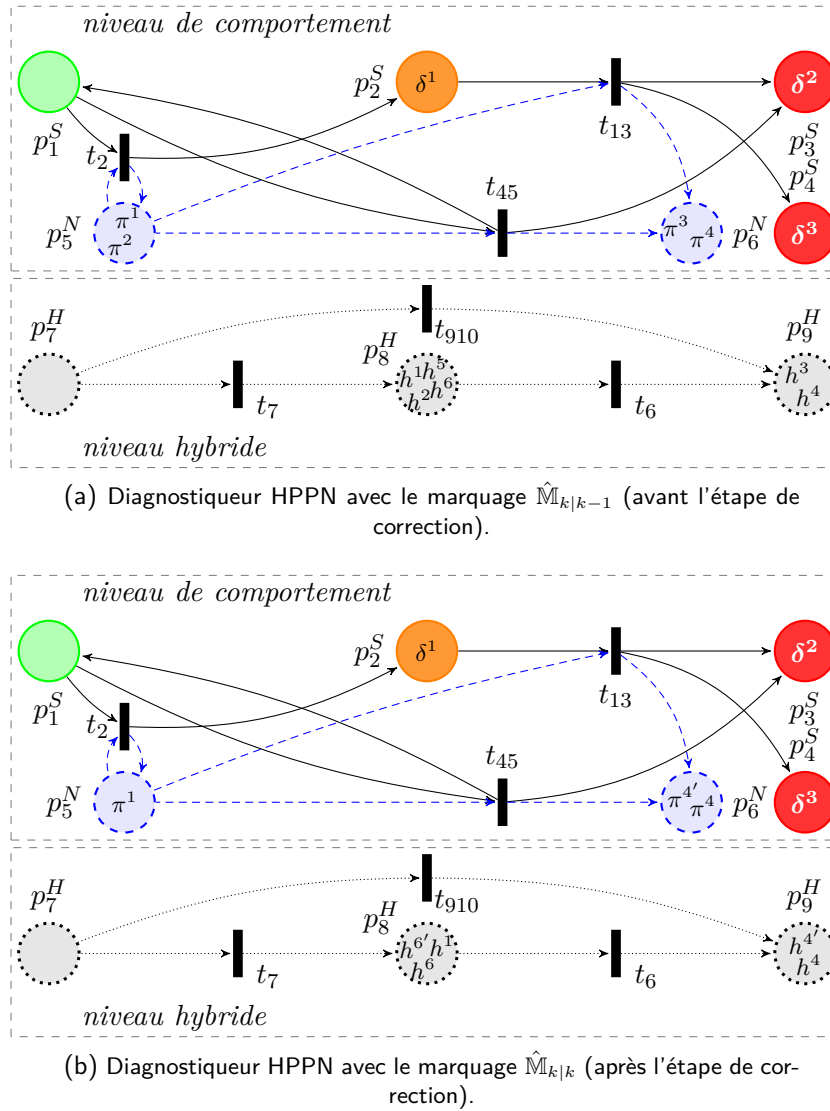


FIGURE 3.14 – Illustration de l'étape de correction du processus du diagnostiqueur HPPN du robot mobile.

Dans l'hypothèse $\{\delta^2, \pi^4, \pi^{4'}, h^4, h^{4'}\}$, ayant un score élevé, la faute f_1 puis la faute f_2 sont survenues et le système a une vitesse nulle. Le mode courant est $Défaillant_1$ car les jetons sont dans les places p_3^S , p_6^N et p_9^H .

La troisième hypothèse, $\{\delta^2, \pi^4, \pi^{4'}, h^6, h^{6'}\}$, qui a également un score élevé, considère que la faute f_1 est survenue, puis l'événement *wur* est survenu, même s'il n'a pas été observé. Le système a une vitesse nulle et est dans le mode $Défaillant_2$, car les jetons sont dans les places p_4^S , p_6^N et p_8^H .

Rappelons enfin que dans les trois hypothèses, les jetons hybrides contiennent les probabilités d'occurrences des fautes. Ici, les jetons h^4 et $h^{4'}$ ne contiennent plus d'information utile pour le diagnostic car ils sont dans la place p_9^H pour laquelle le système est défaillant, mais h^1 , h^6 et $h^{6'}$, dans la place p_8^H estiment toujours la probabilité d'occurrence de la faute f_2 .

Important : On peut maintenant préciser les contributions qu'apporte le diagnostiqueur HPPN par rapport à un diagnostiqueur classique. Les résultats du diagnostiqueur HPPN incluent les résultats d'un diagnostiqueur classique en termes d'occurrences de fautes. Dans un diagnostiqueur classique, cependant, chaque hypothèse de diagnostic a le même degré de croyance. Dans un diagnostiqueur HPPN, les hypothèses portent davantage d'incertitudes. Les ambiguïtés sont évaluées selon les valeurs des jetons et leurs places. Les performances relatives à la gestion des incertitudes sont optimisées grâce à la représentation des connaissances et au processus du diagnostiqueur. En outre, la dégradation du système est estimée (avec les jetons hybrides) durant le processus de diagnostic. L'utilisation du diagnostiqueur HPPN pour la gestion de santé devient donc aisée, particulièrement pour le processus de pronostic.

3.6 Algorithme stochastique de changement d'échelle

Cette section détaille l'algorithme stochastique de changement d'échelle (SSA), utilisé lors de l'étape de correction du diagnostiqueur. L'algorithme est ici présenté dans un cadre général, car il est ensuite réutilisé dans le Chapitre 4.

L'algorithme SSA détermine, à partir d'un ensemble fini de valeurs distribuées sur une plage de départ, une nouvelle distribution de valeurs réparties sur une plage d'arrivée spécifique. L'objectif est d'une part de changer l'intervalle dans lequel sont comprises les valeurs traitées et d'autre part de changer leur cardinal. Ce changement d'échelle utilise une source de hasard et **trois paramètres d'échelle** : ρ^{min} , ρ^{max} et ρ^{tot} (notés ρ_{Δ}^{min} , ρ_{Δ}^{max} et ρ_{Δ}^{tot} pour le diagnostiqueur). Les paramètres ρ^{min} et ρ^{max} spécifient respectivement les bornes minimum et maximum de la plage d'arrivée, et ρ^{tot} spécifie la somme maximale des valeurs de la nouvelle distribution.

Une version de l'algorithme SSA est présentée en Algorithme 1. Dans cette version, il est présenté comme un solveur de problème de répartition de ressources.

Dans ce problème, il y a n **tâches** à réaliser, ayant des **poids** plus ou moins élevés en fonction de leurs importances. Les tâches sont réalisées avec des **ressources**. Plus le nombre de ressources allouées à une tâche est grand, mieux la tâche sera réalisée. Le problème consiste à répartir les ρ^{tot} ressources disponibles aux n tâches en fonction de leurs poids.

Deux contraintes s'ajoutent. Premièrement, une tâche ne peut être réalisée que si on lui alloue au minimum ρ^{min} ressources. Deuxièmement, allouer ρ^{max} ressources à une tâche permet d'obtenir sa meilleure réalisation (en allouer plus n'améliore pas sa réalisation).

La solution proposée s'effectue sur deux étapes :

- la distribution aléatoire (suivant une loi normale en fonction des poids des tâches) de toutes les ressources disponibles,
- la remise à disposition des ressources des tâches qui n'en n'ont pas assez (souscharge) et des ressources inutiles aux tâches qui ont en trop (surcharge).

Ces deux étapes s'enchaînent jusqu'à ce que plus aucune ressource ne soit disponible ou que la répartition reste inchangée après les deux étapes.

Algorithme 1 Changement stochastique d'échelle

Entrée : W ▷ liste des poids des n tâches
Entrée : $\rho^{min}, \rho^{max}, \rho^{tot}$ ▷ paramètres d'échelle
Sortie : R ▷ liste des nombres de ressources affectées aux tâches

- 1: $J \leftarrow (1, \dots, n)$ ▷ liste des ID des tâches à pouvoir (toutes au départ)
- 2: $R \leftarrow (0 | \forall i \in t)$ ▷ liste des nombres de ressources affectées aux tâches (0 au départ)
- 3: $r \leftarrow \rho^{tot}$ ▷ nombre de ressources restantes (toutes disponibles au départ)
- 4:
- 5: **tant que** $r > 0$ **et** $|J| > 0$ **faire** ▷ il reste des ressources et des tâches à pourvoir
- 6:
- 7: $W^J \leftarrow (W_i | \forall i \in J)$ ▷ poids des tâches à pourvoir
- 8: **pour** $o \in \{1, \dots, r\}$ **faire** ▷ répéter r fois
- 9: $J_c \leftarrow Choisir(J, W^J)$ ▷ choix stochastique d'une tâche en fonction des poids
- 10: $R_{J_c} \leftarrow R_{J_c} + 1$ ▷ affecte une ressource à la tâche
- 11: **fin pour**
- 12: **pour chaque** $R_i \in R$ **faire**
- 13: **si** $R_i > \rho^{max}$ **alors** ▷ non respect de la borne maximum
- 14: $R_i \leftarrow \rho^{max}$ ▷ remise à disposition de la surcharge
- 15: **fin si**
- 16: **si** $R_i < \rho^{min}$ **alors** ▷ non respect de la borne minimum
- 17: $R_i \leftarrow 0$ ▷ remise à disposition de la souscharge
- 18: **fin si**
- 19: **fin pour**
- 20:
- 21: $J \leftarrow (J_i | \forall J_i \in J, R_{J_i} < \rho^{max})$ ▷ restriction aux tâches restantes à pourvoir
- 22: $r^n \leftarrow \rho^{tot} - \sum_{R_i \in R} R_i$ ▷ nouveau nombre de ressources restantes
- 23: **si** $r^n = r$ **alors** ▷ pas de ressources distribuées
- 24: $r \leftarrow 0$ ▷ arrêt de l'algorithme
- 25: **sinon**
- 26: $r \leftarrow r^n$ ▷ mise à jour du nombre de ressources restantes
- 27: **fin si**
- 28: **fin tant que**

Pour l'algorithme SSA utilisé durant le processus du diagnostiqueur, les n tâches à réaliser correspondent aux n clusters de particules à redistribuer pendant le rééchantillonnage local. Les poids utilisés sont les scores calculés avec l'Équation 3.34, le score d'un cluster de particules étant le meilleur score de toutes les hypothèses auxquelles il appartient. Les ressources correspondent aux particules. Plus le nombre de particules allouées à un cluster est grand, mieux le mode sera estimé. Ceci revient à déterminer la précision accordée à chaque cluster, et par extension à chacune des hypothèses.

Pour illustrer comment l'algorithme détermine les précisions, on considère l'exemple d'un échantillon de 200 hypothèses. Pour faciliter la compréhension, les hypothèses sont identifiées par des entiers et on suppose qu'elles ont des clusters de particules distincts. La Figure 3.15(a) présente les scores de l'échantillon distribués entre 0 et 1. Les Figures 3.15(b), (c), (d), (e) et (f) montrent les précisions des 200 hypothèses choisies par l'algorithme SSA avec différents jeux de paramètres d'échelle $(\rho^{min}, \rho^{max}, \rho^{tot})$.

L'exemple montre que l'algorithme SSA accorde une précision élevée (ou la précision maximum) aux hypothèses ayant les scores les plus élevés, et une précision faible (ou aucune précision) à celles ayant les scores les plus faibles, tout en respectant les contraintes d'échelle.

L'algorithme SSA peut retourner des résultats non satisfaisants avec des valeurs de paramètres d'échelle et des distributions de scores particulières. Par exemple, avec les paramètres (20,120,1000), il n'affecte que des précisions nulles aux 200 hypothèses, car le paramètre $\rho^{tot} = 1000$ est trop faible par rapport au nombre d'hypothèses et à la précision minimum $\rho^{min} = 20$. Dans de tels cas, un mécanisme de secours alloue la précision maximale ρ^{max} aux hypothèses ayant les scores les plus élevés tant que cela est possible.

On verra par la suite que l'algorithme SSA est également utilisé durant l'initialisation du pronostiqueur (présentée en dans le Chapitre 4). Il servira à déterminer la précision à accorder à la simulation du futur de chaque hypothèse.

3.7 Gestion des performances calculatoires

Un des objectifs dans ces travaux est de considérer le maximum d'incertitudes à propos du modèle du système et de ses observations. Pour y répondre, le processus du diagnostiqueur HPPN combine le pseudo-tirage de ses jetons pour créer de nouvelles hypothèses (basées sur des changements potentiels de modes) et les filtres particuliers (sur chaque cluster de particules) pour estimer et corriger les différents états possibles du système.

La création de toutes les hypothèses avec le pseudo-tirage provoque une explosion combinatoire, inhérente aux systèmes à événements discrets, du nombre total de jetons. De plus, le nombre de jetons utilisé pour estimer les états continu et hybride, est un facteur de performance calculatoire car chaque jeton est mis à jour durant l'étape de prédiction.

Deux étapes dans la méthodologie de diagnostic proposée permettent de gérer les performances calculatoires :

- la **séparation en deux niveaux** lors de la génération du diagnostiqueur permet de réduire la complexité du réseau,
- l'utilisation de l'**algorithme SSA** durant l'étape de correction régule le nombre de jetons total du réseau.

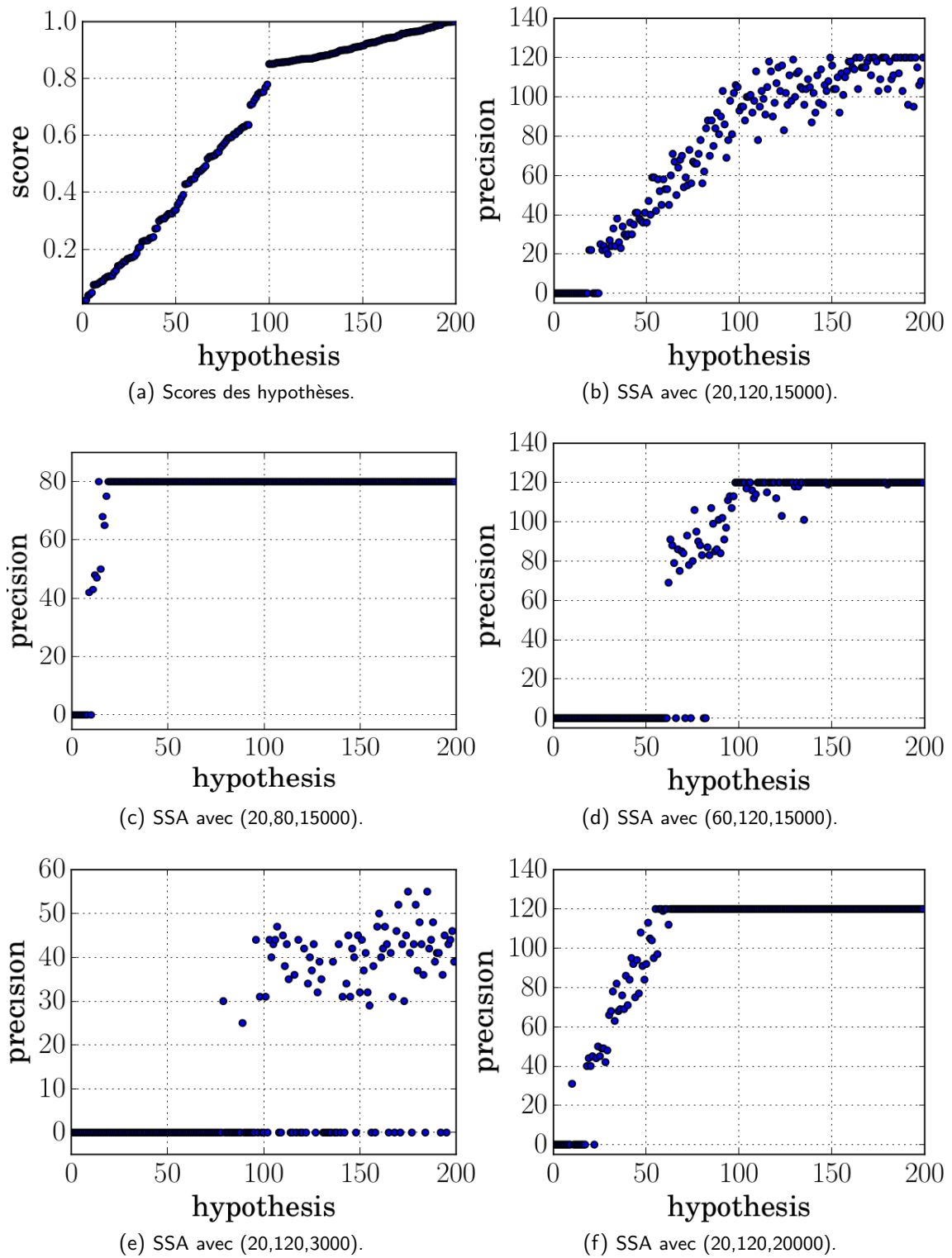


FIGURE 3.15 – Illustration du calcul des précisions associées à 200 hypothèses avec l’algorithme SSA avec différents paramètres d’échelle (ρ^{min} , ρ^{max} , ρ^{tot}).

3.7.1 Réduction de la complexité du réseau

Durant sa génération, le diagnostiqueur est scindé en deux niveaux, formant ainsi deux réseaux de Petri séparés. L'intérêt de cette étape est de séparer les informations que l'on observe (ou que l'on estime et corrige avec les observations) des informations que l'on estime mais que l'on ne peut corriger directement avec les observations.

Le **niveau de comportement** surveille les états discret et continu du système avec les observations du système. Le suivi de l'état discret s'effectue par l'estimation des événements survenus suivant la dynamique discrète. Le suivi de l'état continu s'effectue par l'estimation des variables d'état continues suivant la dynamique continue courante.

Le **niveau hybride** suit l'évolution de l'état hybride avec la dynamique hybride courante du système et les estimations des états discret et continu.

Cette séparation a deux avantages. Premièrement, elle permet de réduire la complexité du réseau en parallélisant l'évolution du marquage hybride. Deuxièmement, elle permet d'optimiser la représentation des états possibles, avec notamment la fusion des transitions du niveau de comportement (détaillée en Section 3.3.5), ce qui autorisera plusieurs hypothèses à partager le même cluster de particules.

Remarque 3.14 (PPN et MPPN). Initialement, le niveau de comportement est inspiré des réseaux de Petri particuliers modifiés (MPPN) [Zouaghi *et al.* 11a], une extension des réseaux de Petri particuliers (PPN) [Lesire & Tessier 05], deux formalismes utilisés pour la surveillance de mission d'un système hybride sous incertitude.

Les PPN ont introduit la combinaison des réseaux de Petri avec le filtrage particulier dans le cadre des systèmes hybrides. C'est une extension des réseaux de Petri hybrides dans laquelle la représentation du marquage continu est distribuée en particules et celle du marquage discret, en configurations, deux types de jetons dont les noms ont été conservés dans ces travaux. Les places symboliques et numériques y sont également définies, et les transitions peuvent avoir plusieurs types de places en entrée et sortie. Dans les PPN, une transition ne peut avoir qu'une seule condition, soit symbolique, soit numérique. Les MPPN étendent les PPN en permettant d'avoir deux conditions (une symbolique et une numérique) sur chaque transition pour modéliser l'occurrence d'un événement simultanément à la satisfaction d'une contrainte sur l'état continu.

Les processus des PPN et MPPN utilisent les configurations et les particules directement pour estimer les états discret et continu. Cela signifie qu'ils ont été nativement définis comme des diagnostiqueurs. L'application des MPPN à la surveillance de santé a d'ailleurs été faite dans [Gaudel *et al.* 14a]. Le processus utilise le pseudo-tirage, combiné à un filtre particulier appliqué à l'ensemble des particules du réseau, pour prédire les états discret et continu, puis les corriger avec les observations sur le système. Le principal problème de ce processus est

que le pseudo-tirage déséquilibre la distribution de croyances sur l'état continu. En effet, si la dynamique continue du système est trop lente par rapport à la dynamique discrète, les particules dupliquées pour représenter un passage d'un nouveau mode seront autant cohérentes avec les observations que les particules originelles. Si ce phénomène se répète un trop grand nombre de fois, les particules dans le nouveau mode deviennent trop nombreuses et le rééchantillage ne sélectionnera plus les particules originelles, provoquant une croyance en un changement de mode incohérent avec les observations.

Outre l'introduction des places, conditions et jetons hybrides pour suivre une nouvelle dynamique, le processus du diagnostiqueur HPPN corrige ce problème en exécutant des filtres particulaires locaux. Ceci est rendu possible grâce aux liens entre les configurations et les particules, fournis par les jetons hybrides.

3.7.2 Régulation du nombre total de jetons

L'utilisation d'un algorithme stochastique de changement d'échelle (SSA) permet de réguler le nombre total de jetons dans le diagnostiqueur en changeant dynamiquement les précisions des hypothèses.

Durant l'étape de correction du diagnostiqueur, un score pour chaque hypothèse est calculé à partir des observations sur le système. L'algorithme SSA attribue alors de meilleures précisions aux hypothèses ayant les scores les plus élevés et des précisions plus faibles aux autres, en se basant sur trois paramètres d'échelle : ρ_{Δ}^{min} , ρ_{Δ}^{max} et ρ_{Δ}^{tot} .

Les paramètres ρ_{Δ}^{min} et ρ_{Δ}^{max} sont représentatifs respectivement des précisions minimale et suffisante accordées au diagnostiqueur pour surveiller une hypothèse. Le paramètre ρ_{Δ}^{tot} est quant à lui représentatif du nombre maximum de jetons à utiliser pour surveiller toutes les hypothèses réunies.

Les paramètres d'échelle du diagnostiqueur instaurent donc un compromis entre le nombre d'hypothèses à surveiller, la précision accordée à chacune, et la puissance de calcul disponible (ρ_{Δ}^{tot} peut être choisi pour répondre à des contraintes de performance de calcul).

3.8 Cas d'étude : les réservoirs d'eau

Dans cette partie, le diagnostiqueur HPPN du système des réservoirs d'eau présenté en Section 2.5 est généré.

Différents scénarios sont ensuite testés pour illustrer les incertitudes prises en compte par le diagnostiqueur HPPN, et notamment sa robustesse aux observations imprécises, manquantes

et fausses. Pour chacun des scénarios, à l'instant $k = 0s$ (0 seconde), le mode du système est Nom_1 :

- les vannes v_{13} et v_{32} sont ouvertes et n'ont jamais été manipulées : $b_0 = \emptyset$,
- l'état continu est $x_0 = [l_{10}, l_{20}, l_{30}] = [0.60, 0.55, 0.58]$,
- l'état hybride est $d_0 = [p_{10}, p_{20}, p_{30}, p_{40}, p_{50}] = [0, 0, 0.001, 0, 0]$.

3.8.1 Diagnostiqueur HPPN

À partir du modèle HPPN spécifié en Section 2.5.2, le diagnostiqueur HPPN du système a été généré. Il est présenté en Figure 3.16. On peut voir qu'il possède les mêmes 19 places que le modèle, séparées en deux niveaux.

Le niveau de comportement du diagnostiqueur ne comprend plus que 12 transitions grâce à l'étape de fusion, ce qui correspond à un gain de 14% par rapport aux 14 transitions du modèle. Les transitions t_1 et t_7 (resp. t_2 et t_9) ont été fusionnées car elles représentaient toutes les deux un changement de la dynamique continue C_1 à la dynamiques C_2 (resp. C_3 à C_4) depuis le même mode Nom_1 (resp. $Dég_2$).

Concernant le niveau hybride, la génération du diagnostiqueur a permis de réduire le nombre de transitions à 7, soit un gain de 50%. Les transitions t_{18} et t_{22} ont été fusionnées car elles liaient toutes les deux la place hybride p_{11} à la place hybride p_{15} et les 6 autres transitions ont été enlevées car elles bouclaient sur une même place hybride.

Rappel

Le diagnostiqueur utilise trois paramètres d'échelle ρ_{Δ}^{min} , ρ_{Δ}^{max} et ρ_{Δ}^{tot} . Les paramètres ρ_{Δ}^{min} et ρ_{Δ}^{max} sont représentatifs des précisions minimale et suffisante accordées au diagnostiqueur pour surveiller une hypothèse. Le paramètre ρ_{Δ}^{tot} est quant à lui représentatif du nombre maximum de jetons à utiliser pour surveiller toutes les hypothèses réunies.

Le score d'une hypothèse est calculé avec l'Équation 3.34, qui dépend d'un paramètre α représentatif de la confiance relative entre la partie discrète et la partie continue du modèle.

L'état continu est représenté par seulement 3 variables (les trois niveaux d'eau). Nous fixons donc les paramètres d'échelle du diagnostiqueur à :

$$(\rho^{min}, \rho^{max}, \rho^{tot}) = (5, 20, 200),$$

pour pouvoir surveiller entre 10 et 40 clusters de particules avec des précisions variants entre 5 et 20. La précision de l'hypothèse initiale est $n_0 = 20$.

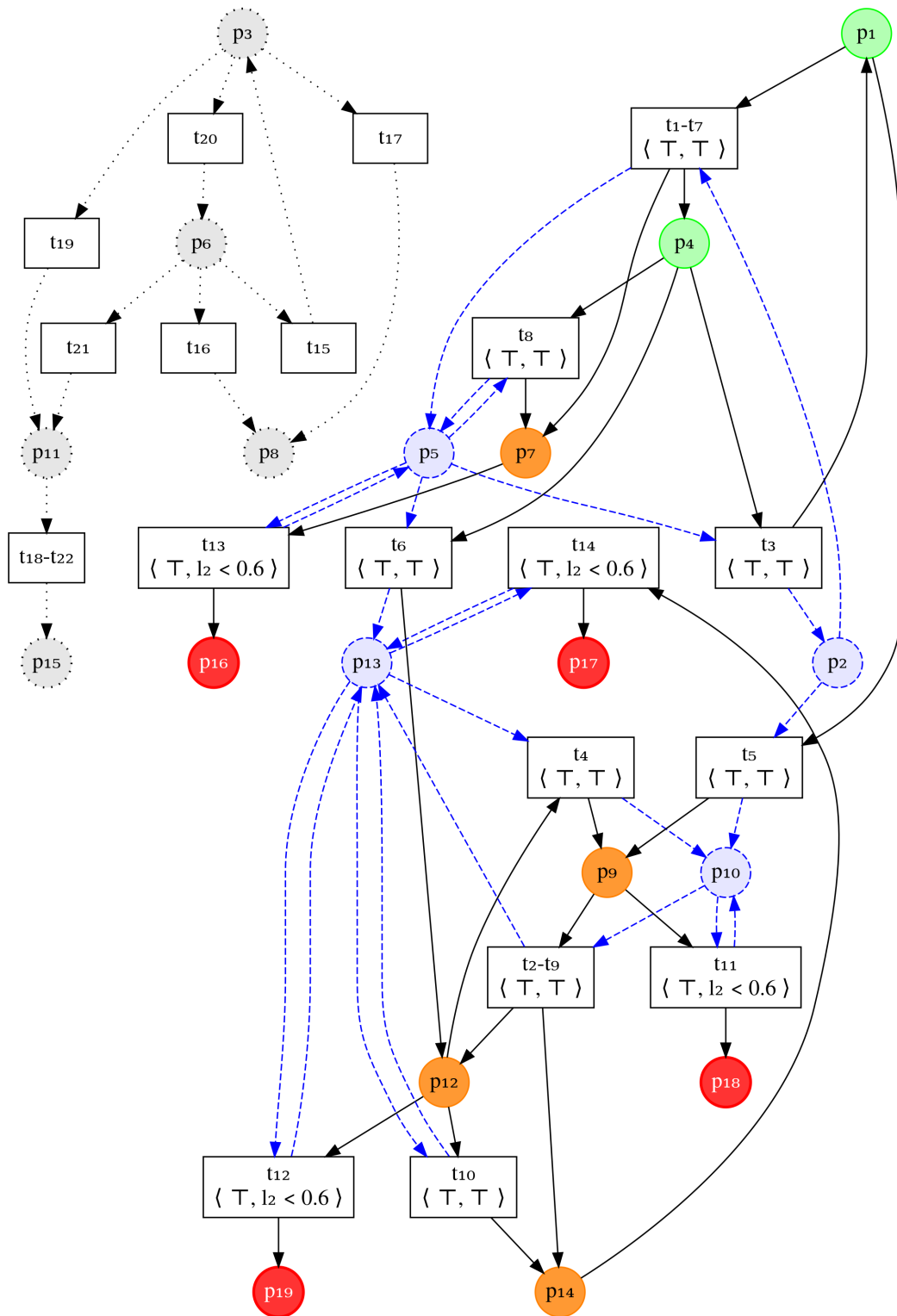


FIGURE 3.16 – Diagnostiqueur HPPN du système des réservoirs d'eau.

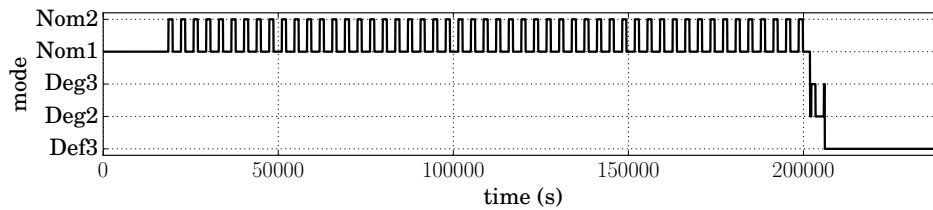


FIGURE 3.17 – Mode du système simulé à chaque instant pour le scénario 1.

On accorde une confiance égale entre les parties discrète et continue du modèle : $\alpha = 0.5$.

3.8.2 Résultats

Nous avons défini trois scénarios qui mettent en avant les avantages de la méthode de diagnostic basée sur les HPPN. Dans tous les scénarios, à chaque instant k , le débit de l'eau versée dans le réservoir $Tank_1$ vaut $q_{1k} = 0.0000023 \text{ m}^3 \cdot \text{s}^{-1}$. Après 310 minutes de fonctionnement, la vanne v_{13} est fermée toutes les heures pendant 20 minutes pour effectuer un traitement de l'eau du réservoir $Tank_1$. Au bout d'un certain temps, le réservoir $Tank_1$ se met à fuir (faute f_1) et le niveau d'eau du réservoir $Tank_2$ passe sous le seuil l_{2min} (faute f_0) quelques instants après.

Avec un simulateur, nous avons généré trois scénarios d'observations.

3.8.2.1 Scénario 1

Pour ce scénario, les observations continues issues du simulateur ont été générées avec deux fois plus de bruits que les spécifications du modèle et les observations discrètes contiennent toutes les commandes discrètes $ouvr_{v_{13}}$ et $ferm_{v_{13}}$ aux instants où elles ont été simulées.

Dans ce scénario, le premier objectif est de présenter les résultats de diagnostic sous la forme de distributions de croyances sur le mode courant du système et sur sa trajectoire de modes passée. Ici, la trajectoire passée du système correspond à séquence des modes dans lesquels il a été. Le second objectif est de montrer comment le diagnostiqueur gère les situations ambiguës.

Pour faciliter la lecture des résultats, la Figure 3.17 montre le mode du système simulé à chaque instant durant le scénario 1.

La Figure 3.18 montre les observations continues $y_0 = l_1$ et $y_1 = m_{tot}$ générées par le simulateur à chaque instant. À partir de 18600s, on voit le niveau d'eau l_1 augmenter significativement de manière régulière pendant 1200s jusqu'à environ 5 mètres puis redescendre jusqu'à approximativement 2.5 mètre. Ces données sont représentatives des changements de modes réguliers

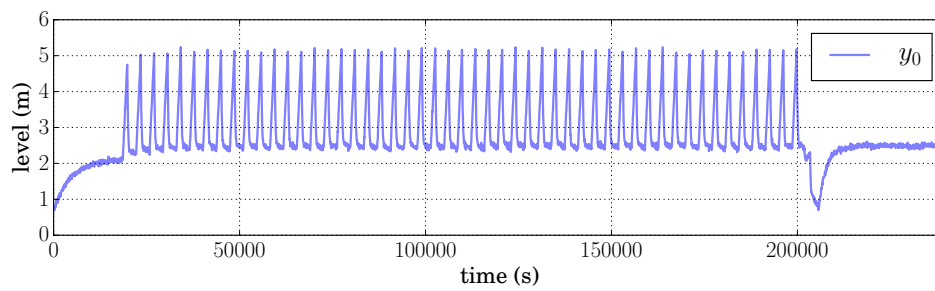
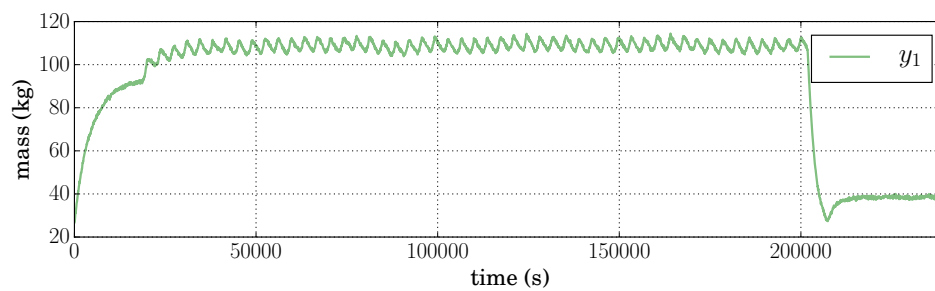
(a) Niveau d'eau l_1 (b) Masse totale d'eau m_{tot}

FIGURE 3.18 – Observations continues pour le scénario 1.

entre le mode Nom_1 et le mode Nom_2 . La faute f_1 survient à 201840s. Le système entre alors dans le mode Deg_2 puis oscille entre les modes Deg_2 et Deg_3 (ouvertures et fermetures de la vanne v_{13}) jusqu'à occurrence de f_0 à 206040s et l'entrée du système dans le mode Def_3 .

La suite de cette section présente et interprète les résultats de la méthodologie de diagnostic basée sur les HPPN pour le scénario 1.

La Figure 3.19 expose les distributions de croyances du diagnostiqueur sur le mode courant du système à chaque instant. Le degré de croyance d'un mode à chaque instant k est représenté par l'épaisseur du trait et le degré de croyance le plus élevé de tous les modes est affiché en bleu. Le degré de croyance d'un mode à l'instant k est le maximum des degrés de croyance de toutes les hypothèses du diagnostiqueur qui sont dans ce mode à l'instant k . Le degré de croyance d'une hypothèse est son score, calculé avec l'Équation 3.34.

Sur la Figure 3.19, on peut voir qu'à chaque instant, le diagnostiqueur considère plusieurs hypothèses concernant le mode courant du système. Par exemple, avant 18600s, le mode du système ayant le degré de croyance le plus élevé est le mode Nom_1 , qui est le mode du système simulé (voir Figure 3.17). D'autres modes ont des degrés de croyance élevés, comme les modes Nom_2 , Deg_1 ou Deg_2 . Cela est dû au comportement continu lent du système qui ne permet pas au diagnostiqueur d'identifier en une période d'échantillonnage la dynamique continue courante de manière certaine, et que plus d'observations lui sont nécessaires.

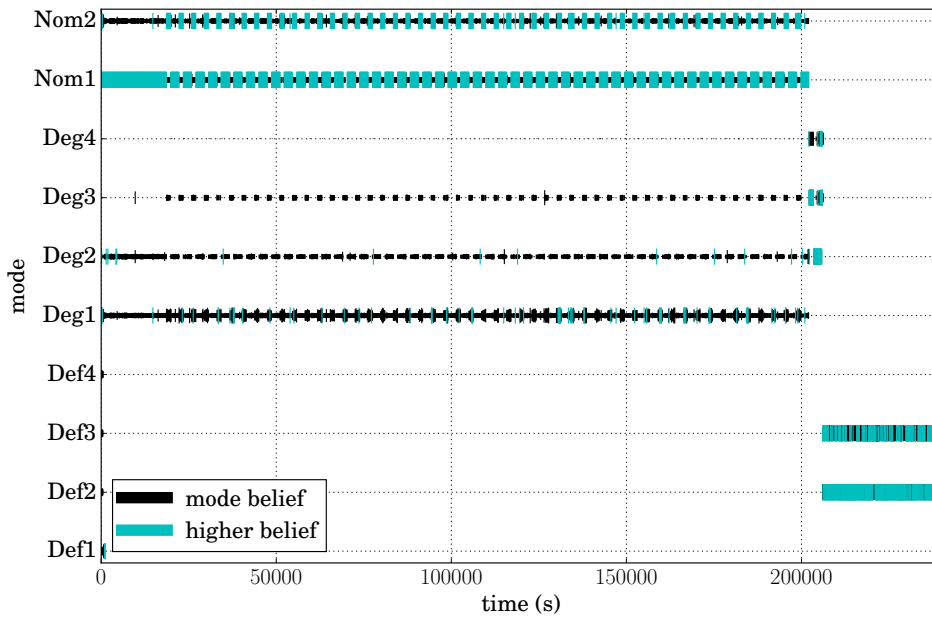


FIGURE 3.19 – Distributions de croyances sur le mode du système à chaque instant pour le scénario 1.

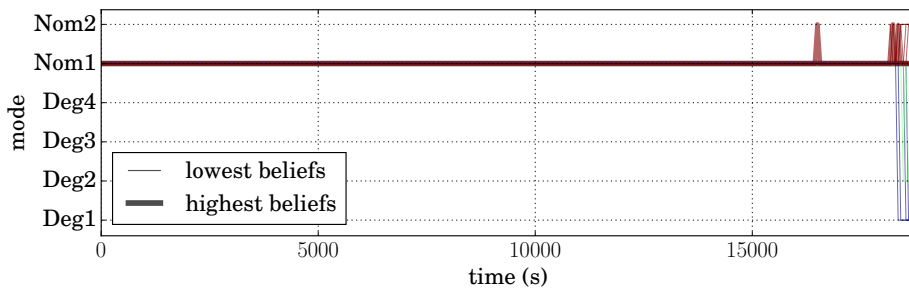
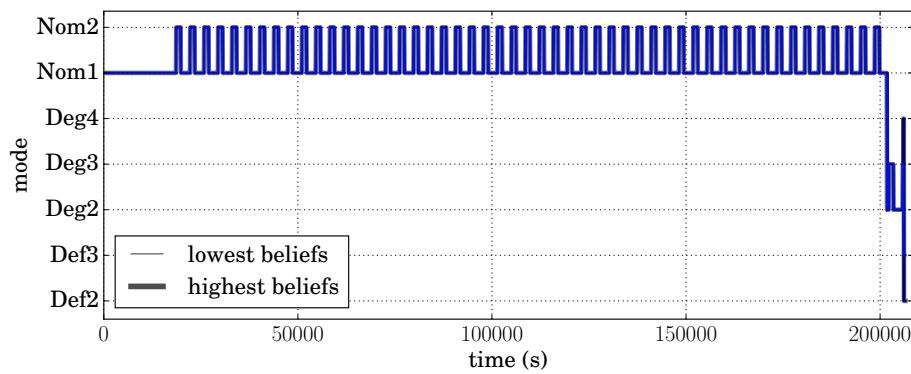


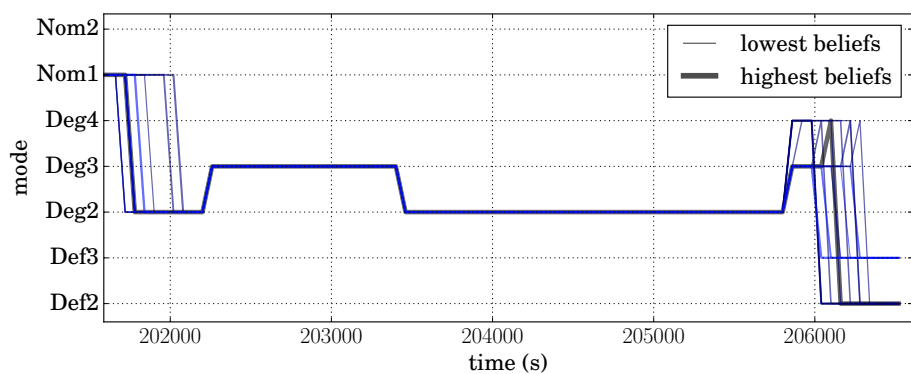
FIGURE 3.20 – Distribution de croyances sur la trajectoire passée du système à 18600s pour le scénario 1.

La Figure 3.20 montre les trajectoires passées des hypothèses contenues dans le diagnostiqueur à 18600s. L'épaisseur du trait est représentatif du degré de croyance de l'hypothèse. On peut voir que l'hypothèse ayant le degré de croyance le plus élevé considère que le système est resté dans le mode Nom_1 depuis le début du scénario. D'autres hypothèses considèrent que le système soit passé brièvement dans le mode Nom_2 . Ces hypothèses ont des degrés de croyance élevés car le niveau d'eau l_1 est en train de monter de manière significative (voir Figure 3.18(a)), ce qui est un comportement proche de la dynamique continue du mode Nom_2 . Leurs degrés de croyance sont néanmoins moins élevés que celui de l'hypothèse d'être resté dans le mode Nom_1 depuis le début car aucune commande discrète n'a encore été observée.

Dans la Figure 3.19, jusqu'à 200000s, on peut voir que le mode courant ayant le degré de croyance le plus élevé est quasiment toujours celui du système simulé (voir la Figure 3.17). Les



(a) Vue d'ensemble



(b) Zoom entre 201600s et 206600s

FIGURE 3.21 – Distribution de croyances sur la trajectoire passée du système à 206520s pour le scénario 1.

degrés de croyance plus ou moins élevés des autres modes s'expliquent soit par les incertitudes associées aux dynamiques et aux observations, soit par des situations ambiguës. Par exemple, à chaque instant, les degrés de croyance des modes Nom_2 et Deg_1 sont assez similaires car ces deux modes ont la même dynamique; la vanne v_{13} est en position fermée, qu'elle soit bloquée (faute f_4) ou non. Lorsque la vanne v_{13} est fermée, le mode Nom_2 a tout de même un degré de croyance supérieur à celui du mode Deg_1 car les commandes $ferm_{v_{13}}$ et $ouvr_{v_{13}}$ sont observées par le diagnostiqueur.

Bien qu'à chaque instant, le mode ayant le degré de croyance le plus élevé ne soit pas toujours celui du système simulé, la Figure 3.21(a), qui expose la distribution de croyances sur la trajectoire passée du système à 206520s (peu après l'entrée du système dans le mode Def_3), montre que les trajectoires des hypothèses du diagnostiqueur correspondent parfaitement à celle du système simulé jusqu'à 200000s.

La Figure 3.21(b) est un zoom de la Figure 3.21(a) entre 201600s et 206600s. Elle montre que les fautes f_1 et f_0 ont bien été diagnostiquées et illustre les incertitudes du diagnostiqueur concernant leurs dates d'occurrences. On voit sur la figure que le diagnostiqueur considère

que la faute f_1 est survenue entre 201660s et 202020s. Cet intervalle de temps contient bien 201840s, la date à laquelle a été simulée f_1 . Concernant la faute f_0 , le diagnostiqueur considère qu'elle est survenue entre 205980s et 206280s. La date 206040s à laquelle f_0 a été simulée est bien incluse dans ces bornes.

Dans la Figure 3.21(b), on peut néanmoins voir que l'hypothèse ayant le degré de croyance le plus élevé considère que le système est entré dans le mode Deg_4 avant l'occurrence de f_0 . Ce changement de mode correspond à l'occurrence de la faute f_4 (la vanne v_{13} est bloquée en position fermée). Il s'agit là encore d'une situation ambiguë car les modes Deg_3 et Deg_4 ont exactement la même dynamique continue. Ce phénomène met en évidence le fait que le diagnostiqueur considère l'occurrence de la faute f_4 à chaque instant et garde les hypothèses correspondantes tant qu'elles sont cohérentes avec les observations continues. Dans cette situation particulière, puisque le système est tombé en panne, toutes les hypothèses restent cohérentes avec les observations continues jusqu'à la fin et l'incertitude sur le mode courant du système est bien gardée. Cela est aussi visible sur la Figure 3.19 dans laquelle, à partir de 206280s, les modes Def_2 et Def_3 sont les uniques modes possibles du système (d'après le diagnostiqueur) et ont tous les deux des degrés de croyance très élevés.

3.8.2.2 Scénario 2

Le second scénario est le même que le scénario 1, à la différence près que des commandes discrètes $ouvr_{v_{13}}$ et $ferm_{v_{13}}$ ont aléatoirement été ajoutées (4.8%) et retirées (5.6%) des observations du diagnostiqueur. L'objectif est de montrer la robustesse du diagnostiqueur face à des observations discrètes fausses et manquantes.

La Figure 3.22 montre les distributions de croyances sur le mode du système à chaque instant du scénario 2. En comparant avec le scénario 1 (Figure 3.19), on peut remarquer deux phénomènes.

Premièrement, les croyances dominantes du mode Nom_1 lorsque la vanne v_{13} est ouverte et du mode Nom_2 lorsque qu'elle est fermée sont beaucoup moins régulières. Ce résultat était prévisible car les observations discrètes sont moins cohérentes que dans le scénario 1.

Deuxièmement, entre 18600s (début des ouvertures et fermetures répétées de la vanne v_{13}) et 200000s (juste avant l'occurrence de la faute f_1), les modes Deg_2 et Deg_3 ont un degré de croyance nul à quasiment tous les instants. Ce phénomène est la conséquence d'une incohérence trop importante entre les observations discrètes et la dynamique discrète du modèle. Techniquement, les ensembles d'événements portés par les configurations sont trop différents de l'ensemble des observations discrètes du diagnostiqueur. Ainsi, tous les jetons obtiennent des poids très faibles avec la fonction utilisée par défaut pour déterminer la pertinence d'une configuration (voir Section 3.4.2.1). De fait, les configurations ont toutes des poids similaires

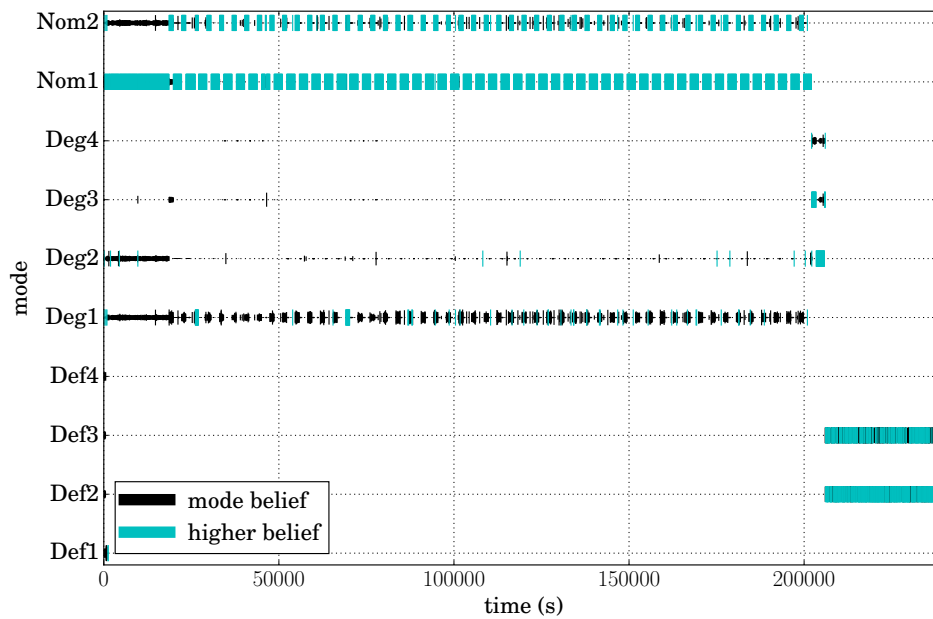


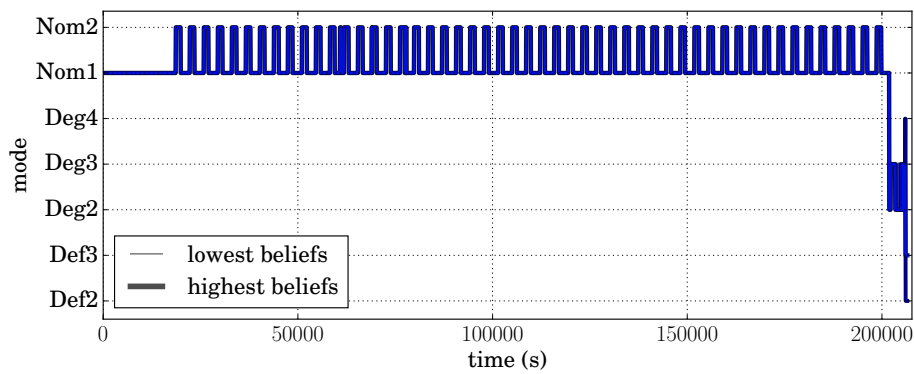
FIGURE 3.22 – Distributions de croyances sur le mode du système à chaque instant pour le scénario 2.

et cela signifie aussi qu'elles n'ont plus d'impact significatif sur le calcul des scores des hypothèses. Les hypothèses dont les particules ne sont pas suffisamment cohérentes avec les observations continues ne sont donc pas gardées.

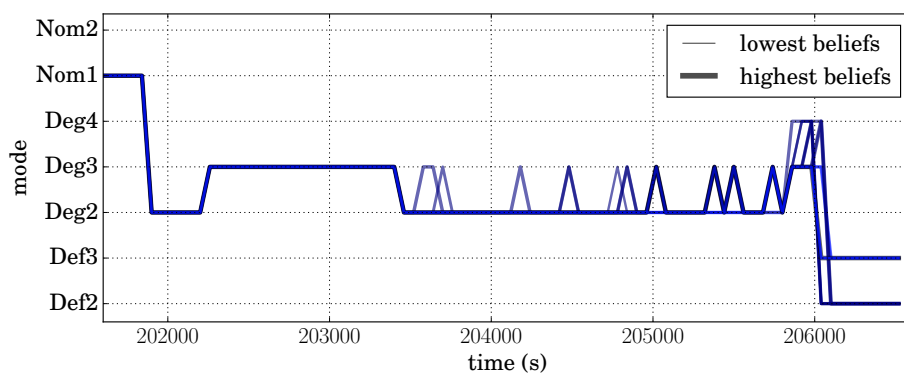
Dans le scénario 1, les hypothèses dont la configuration portait un ensemble d'événements proche des observations discrètes du diagnostiqueur étaient gardées plus longtemps, même si leurs particules étaient moins cohérentes avec les observations continues.

La Figure 3.23(a) montre la distribution de croyances sur la trajectoire passée du système à 206520s. On peut voir que la trajectoire ayant le degré de croyance le plus élevé jusqu'à 200000s est finalement très similaire à celle du système simulé (voir la Figure 3.17). Ceci illustre la robustesse du diagnostiqueur face aux observations fausses et manquantes. Le zoom entre 201600s et 206600s nous permet de voir que les fautes f_1 et f_0 sont correctement diagnostiquées et que moins d'incertitudes sont considérées vis-à-vis de leurs dates d'occurrences que dans le scénario 1. Par exemple, le diagnostiqueur considère que la faute f_1 est survenue à 201840s, qui est la date à laquelle elle a été simulée. Quant à la faute f_0 , le diagnostiqueur considère qu'elle est survenue entre 205980s et 206040s. Cet encadrement est plus petit que celui du scénario 1 mais il contient bien la date la simulation de la faute (206040s). Par ailleurs, le même phénomène concernant la situation d'ambiguïtés entre les modes Deg_3 et Deg_4 est observé.

La surveillance semble donc plus efficace dans ce scénario. On remarque cependant dans la



(a) Vue d'ensemble



(b) Zoom entre 201600s et 206600s

FIGURE 3.23 – Distribution de croyances sur la trajectoire passée du système à 206520s pour le scénario 2.

Figure 3.23(b) qu'entre 203480s et 205820, de nombreuses hypothèses considèrent que le système est passé brièvement dans le mode Deg_3 . Ceci est dû au fait que le calcul des scores accorde beaucoup plus d'importance à la partie continue, mais est aussi dû à la ressemblance entre les dynamiques continues des deux modes Deg_2 et Deg_3 , ainsi qu'aux incertitudes associées.

3.8.2.3 Scénario 3

Pour le troisième scénario, les observations continues sont générées avec quatre fois plus de bruits que les spécifications du modèle. Les observations discrètes contiennent toutes les commandes discrètes $ouvr_{v_{13}}$ et $ferm_{v_{13}}$ aux instants où elles ont été simulées. L'objectif est de montrer la robustesse du diagnostiqueur face à des imprécisions importantes de la partie continue du modèle.

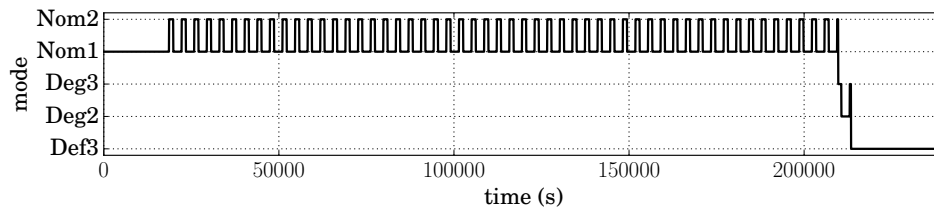


FIGURE 3.24 – Mode du système simulé à chaque instant pour le scénario 3.

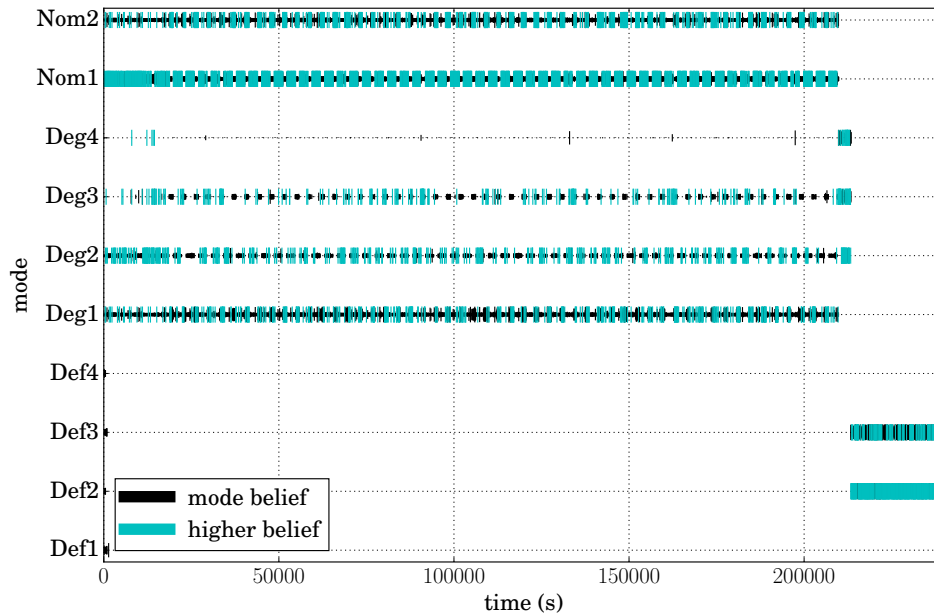
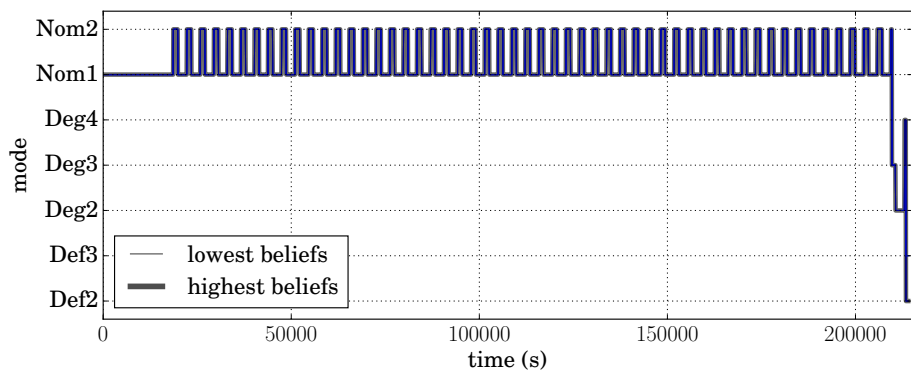


FIGURE 3.25 – Distributions de croyances sur le mode du système à chaque instant pour le scénario 3.

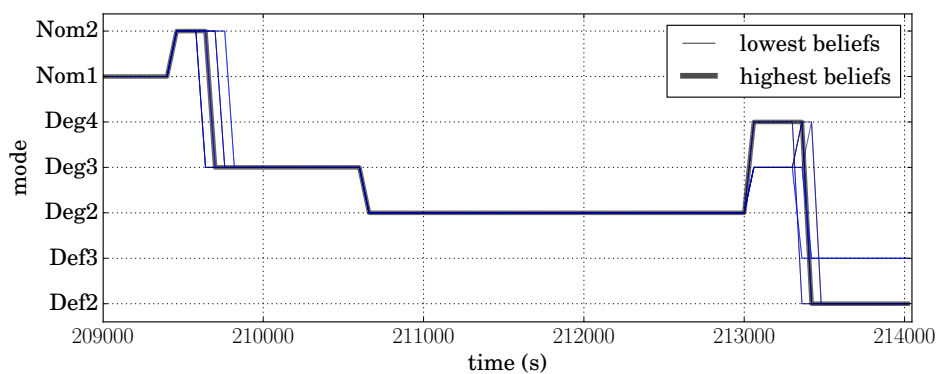
Le mode du système simulé à chaque instant est montré en Figure 3.24. Dans ce scénario, la faute f_1 est simulée à 209700s quand le système est dans le mode Nom_2 . La faute f_0 survient à 213360s.

La Figure 3.25 expose les distributions de croyances sur le mode du système à chaque instant du scénario 3. Par rapport au scénario 1 (Figure 3.19), on constate qu'aucun mode ne prédomine les autres à chaque instant. Ce résultat est cohérent car les observations continues ne permettent pas au diagnostiqueur d'identifier les dynamiques continues aussi précisément que dans le scénario 1.

Cependant, en regardant la distribution de croyances sur la trajectoires du système à 214020s en Figure 3.26(a), on peut voir que la trajectoire estimée du système jusqu'à l'occurrence de f_1 est exactement celle du système simulé. Ceci illustre la robustesse du diagnostiqueur face aux imprécisions des observations continues. Le zoom entre entre 209000s et 214000s nous permet de voir que les fautes f_1 et f_0 sont correctement diagnostiquées. Le diagnostiqueur considère



(a) Vue d'ensemble



(b) Zoom entre 209000s et 214000s

FIGURE 3.26 – Distribution de croyances sur la trajectoire passée du système à 214020s pour le scénario 3.

que la faute f_1 est survenue entre 209580s et 209760s, intervalle qui inclut bien 209700s, la date d'occurrence de la faute. Il considère aussi que la faute f_0 est survenue entre 213300s et 213420s, intervalle qui inclut 213360s, la date d'occurrence de la faute. Le phénomène concernant la situation d'ambiguïtés entre les modes Deg_3 et Deg_4 est également observé.

Le scénario 2 et le scénario 3 soulignent la robustesse du diagnostiqueur face aux incertitudes symboliques et numériques. Dans le scénario 2 nous avons vu que lorsqu'un manque de cohérence sur la partie discrète du modèle se présente, le diagnostiqueur réussit à garder les hypothèses qui expliquent le mieux la situation grâce à la partie continue. Le scénario 3 montre que la partie discrète joue un rôle similaire lorsque les incohérences concernent la partie continue.

3.9 Conclusion

Dans ce chapitre, nous avons proposé une méthode de diagnostic basée sur la génération d'un diagnostiqueur HPPN à partir du modèle HPPN du système. Le diagnostiqueur HPPN

surveillance sous incertitude à la fois le comportement du système et sa dégradation. Le processus de diagnostic utilise à chaque instant les observations sur le système pour faire évoluer son marquage (ses jetons) sur le principe de prédiction et de correction des filtres particulaires, combiné au principe du pseudo-tirage et à un algorithme stochastique de changement d'échelle (SSA).

Le diagnostic courant du système est le marquage courant du diagnostiqueur, qui porte une distribution de croyances sur l'état du système et sur les séquences d'événements (dont les fautes) menant à cet état. Les ambiguïtés sont évaluées selon les valeurs des jetons et leurs places.

Ce chapitre détaille également l'algorithme SSA qui instaure un compromis entre la précision des hypothèses de diagnostic et les performances calculatoires de la fonction de diagnostic.

Enfin, la méthodologie de diagnostic a été appliquée sur le système des réservoirs d'eau qui a été décrit et modélisé dans la Section 2.5. Le diagnostiqueur HPPN du système a été généré puis trois scénarios ont été joués pour illustrer les concepts abordés dans ce chapitre et la robustesse du diagnostiqueur face aux incertitudes symboliques et numériques. Cette application a été le sujet de [Gaudel *et al.* 15a] et [Gaudel *et al.* 15b].

Chapitre 4

Pronostic

Résumé

Ce chapitre présente la méthodologie de pronostic de système basée sur les HPPN et propose son intégration avec la méthodologie de diagnostic présentée au Chapitre 3.

La méthode de pronostic se base sur la génération d'un pronostiqueur HPPN à partir d'un modèle HPPN. Les transitions et les conditions du modèle sont transformées pour pouvoir simuler l'évolution de l'état du système dans le futur.

L'intégration consiste à utiliser le diagnostic courant pour calculer le pronostic courant à l'aide d'un ensemble de futures entrées sur le système. À chaque fois qu'un diagnostic est calculé, le pronostiqueur est réinitialisé et le processus de pronostic démarre.

L'initialisation du pronostiqueur utilise l'algorithme SSA afin de sélectionner les hypothèses de diagnostic dont les simulations des évolutions dans le futur sont les plus pertinentes. Le marquage du pronostiqueur évolue ensuite avec les futures entrées sur le système, planifiées à l'instant courant sur un certain horizon temporel.

Le pronostic courant du système est le marquage du pronostiqueur à la fin de la prédiction. Il porte une distribution de croyances sur les futurs états possibles et les séquences de fautes menant à ces états. On peut particulièrement obtenir une distribution de croyances sur la date à partir de laquelle le système sera dans un mode de défaillance.

Tous les concepts abordés dans la méthodologie de pronostic sont d'abord illustrés sur l'exemple du robot mobile puis appliqués au système des réservoirs d'eau.

La dernière section de chapitre propose une intégration avancée des modules de diagnostic et de pronostic ayant pour but d'améliorer les résultats des surveillance et de répondre à des objectifs relatifs à la gestion de santé.

4.1 Introduction

Dans un contexte de surveillance de santé, l'objectif du pronostic est de prédire les états futurs du système et son RUL/EOL, à partir du diagnostic courant et des entrées futures sur le système. La **fin de vie** (*End Of Life* ou **EOL**) est la date à laquelle le système n'est plus opérationnel, alors que la **durée de vie résiduelle** (*Remaining Useful Life* ou **RUL**) est la durée restante jusqu'à la fin de vie. Particulièrement, on s'intéresse à déterminer si et quand le système va entrer dans un mode de défaillance durant un horizon de prédiction τ_p , choisi par l'opérateur.

Avant de détailler la méthode de pronostic basée sur les HPPN, la Section 4.2 propose une méthodologie de surveillance de santé intégrant diagnostic et pronostic. L'intégration consiste à utiliser le diagnostic courant pour calculer le pronostic courant à l'aide d'un ensemble de futures entrées sur le système.

On définit l'ensemble, planifié au temps k , des entrées sur le système prévues durant l'horizon de prédiction τ_p , $U_k^+ = \{U_\kappa | \kappa \in \{k+1, \dots, k+\tau_p\}\}$, avec $U_\kappa = (U_\kappa^S, u_\kappa^N)$ la paire de l'ensemble des entrées discrètes U_κ^S et du vecteur d'entrée continu $u_\kappa^N \in \mathbb{R}^{n_u}$, à l'instant futur κ .

Dans ces travaux, nous proposons de générer un **pronostiqueur** HPPN d'un système à partir de son modèle HPPN. Ce pronostiqueur HPPN permet la simulation de l'évolution de l'état du système dans le futur et ainsi la détermination du EOL ou du RUL du système.

Les étapes de la génération du pronostiqueur sont détaillées en Section 4.3.

La Section 4.4 présente le processus du pronostic. Il consiste à initialiser le pronostiqueur avec le diagnostic courant Δ_k et l'algorithme SSA, puis à faire évoluer son marquage avec l'ensemble des futures entrées U_k^+ . Le pronostiqueur arrête la prédiction lorsque toutes les entrées ont été simulées ou avant si toutes les hypothèses ont atteint un mode de défaillance, dans le but de réduire les temps de calcul.

Dans la Section 4.5, le **pronostic** Π_k est défini comme le marquage du pronostiqueur à la date de fin de prédiction k_{EOP} (*End Of Prediction*). Le pronostic Π_k est une distribution de croyances sur les **trajectoires futures** du système, c'est-à-dire les états futurs du système et les séquences d'événements (et particulièrement de fautes) menant à ces états. Il permet aussi de retrouver une distribution de croyances sur le RUL/EOL du système.

La Section 4.6 fait un point sur les différentes sources d'incertitudes liées au pronostic des systèmes.

La Section 4.7 applique la méthode de pronostic au système des réservoirs d'eau présenté en Section 2.5.

Enfin, la Section 4.8 propose une intégration avancée des méthodes de diagnostic et de pronostic basées sur les HPPN ayant pour but d'améliorer les résultats de la surveillance de santé vis-à-vis d'objectifs relatifs à la gestion de santé.

4.2 Intégration des méthodes de diagnostic et de pronostic

Une vue d'ensemble d'une méthodologie de surveillance de santé intégrant diagnostic et pronostic est donnée dans l'Algorithme 2.

Algorithme 2 Intégration diagnostic/pronostic

```

1:  $HPPN_{\Phi} \leftarrow \text{CréerModèle}()$ 
2:  $HPPN_{\Delta} \leftarrow \text{GénéralerDiagnostic}(\text{HPPN}_{\Phi})$ 
3:  $HPPN_{\Pi} \leftarrow \text{GénéralerPronostic}(\text{HPPN}_{\Phi})$ 
4: pour chaque  $k$  faire
5:    $O_k \leftarrow (U_k^S, u_k^N, Y_k^S, y_k^N)$ 
6:    $\Delta_k \leftarrow \text{MettreÀJour}(\text{HPPN}_{\Delta}, k, O_k)$ 
7:    $\Pi_k \leftarrow \text{Pronostiquer}(\text{HPPN}_{\Pi}, \Delta_k, U_k^+)$ 
8: fin pour

```

La première étape (ligne 1) consiste à modéliser le système avec un HPPN (voir la Section 2.4). La seconde étape (ligne 2) consiste à générer le diagnostiqueur $HPPN_{\Delta}$ à partir du modèle $HPPN_{\Phi}$ (voir la Section 3.3). La troisième étape (ligne 3) consiste à générer le pronostiqueur $HPPN_{\Pi}$ à partir du modèle $HPPN_{\Phi}$ (présenté dans ce chapitre en Section 4.3). Ensuite, le processus du diagnostiqueur (lignes 4-6) utilise les observations (entrées et sorties du système) consécutives O_k pour mettre à jour le marquage du diagnostiqueur et calculer le diagnostic Δ_k . Enfin, le pronostic Π_k du système au temps k est calculé (ligne 7). Le processus de pronostic et le résultat Π_k sont décrits dans les Sections 4.4 et 4.5.

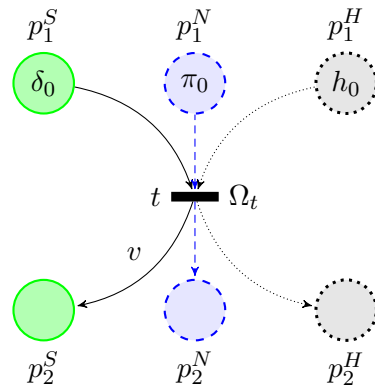
4.3 Génération du pronostiqueur

Cette section détaille les étapes de la génération d'un pronostiqueur $HPPN_{\Pi}$ à partir d'un modèle $HPPN_{\Phi}$.

Rappel : Structure d'un modèle $HPPN_{\Phi}$

$HPPN_{\Phi}$ est le modèle d'un système hybride. Chaque mode du système est représenté par une place symbolique, une place numérique et une place hybride, mais une place peut participer à la représentation de plusieurs modes. Les transitions modélisent les changements de modes, donc chaque transition t a trois places en entrée (une de chaque type) et

trois places en sortie (une de chaque type également). En conséquence, son ensemble de conditions associées est composé d'une condition symbolique, d'une condition numérique, et d'une condition hybride : $\Omega_t = \langle \omega_t^S, \omega_t^N, \omega_t^H \rangle$. Une interprétation généralisée est que le système change de mode si la condition logique $(\omega_t^S \wedge \omega_t^N) \vee \omega_t^H$ est satisfaite.



Définition 31 (Pronostiqueur HPPN). Soit un modèle HPPN défini par le 11-uplet $HPPN_\Phi = \langle P_\Phi, T_\Phi, A_\Phi, \mathcal{A}_\Phi, E_\Phi, X_\Phi, D_\Phi, \mathcal{C}_\Phi, \mathcal{H}_\Phi, \Omega_\Phi, \mathbb{M}_{0\Phi} \rangle$ comme présenté en Section 2.3. Le pronostiqueur $HPPN_\Pi$ de $HPPN_\Phi$ est un 11-uplet généré à partir de $HPPN_\Phi$:

$$HPPN_\Pi = \langle P_\Pi, T_\Pi, A_\Pi, \mathcal{A}_\Pi, E_\Pi, X_\Pi, D_\Pi, \mathcal{C}_\Pi, \mathcal{H}_\Pi, \Omega_\Pi, \mathbb{M}_{0\Pi} \rangle. \quad (4.1)$$

Les éléments de $HPPN_\Pi$ sont déterminés avec les étapes suivantes :

- Copie du modèle, détaillée en Section 4.3.1,
- Modification des conditions, détaillée en Section 4.3.2,
- Suppression des transitions non tirables, détaillée en Section 4.3.3.
- Suppression des bruits de l'évolution de l'état (optionnelle), détaillée en Section 4.3.4,

La suite de cette section détaille les différentes étapes de la génération du pronostiqueur.

4.3.1 Copie du modèle

La première étape est la copie du modèle :

$$HPPN_\Pi \leftarrow HPPN_\Phi. \quad (4.2)$$

Le rôle du pronostiqueur est de simuler l'évolution de l'état du système dans le futur pour déterminer le EOL ou le RUL du système. En conséquence, ses places, ses labels d'événements et ses espaces d'états restent les mêmes que ceux du modèle $HPPN_\Phi$:

- $P_{\Pi} = P_{\Phi}$,
- $E_{\Pi} = E_{\Phi}$,
- $X_{\Pi} = X_{\Phi}$,
- $D_{\Pi} = D_{\Phi}$.

Le marquage initial du pronostiqueur $\mathbb{M}_{0\Pi}$ est déterminé à partir du processus de diagnostic (voir la Section 4.4.1).

4.3.2 Modification des conditions

Les conditions $\Omega_t = \langle \omega_t^S, \omega_t^N, \omega_t^H \rangle$ d'une transition $t \in T_{\Pi}$ sont transformées en accord avec la gestion des incertitudes (voir Section 4.6).

Premièrement, si la condition symbolique ω_t^S teste l'occurrence d'un évènement qui n'est pas une entrée discrète (par exemple une faute ou une interaction avec l'environnement), elle devient \top , car le pronostiqueur travaille uniquement avec les évènements d'entrée et donc les autres évènements ne doivent pas obstruer la prédiction :

$$\forall t \in T_{\Pi}, (\omega_t^S = occ(b, v) \wedge v \notin U^S) \Rightarrow \omega_t^S \leftarrow \top, \quad (4.3)$$

où U^S est l'ensemble des labels de toutes les entrées discrètes.

Deuxièmement, si les deux conditions ω_t^S et ω_t^N valent \top , elles deviennent \perp , pour que le tirage de la transition t ne dépende que de ω_t^H ($\omega_t^S \wedge \omega_t^N$ vaut toujours \perp) :

$$\forall t \in T_{\Pi}, (\omega_t^S = \top \wedge \omega_t^N = \top) \Rightarrow (\omega_t^S \leftarrow \perp, \omega_t^N \leftarrow \perp). \quad (4.4)$$

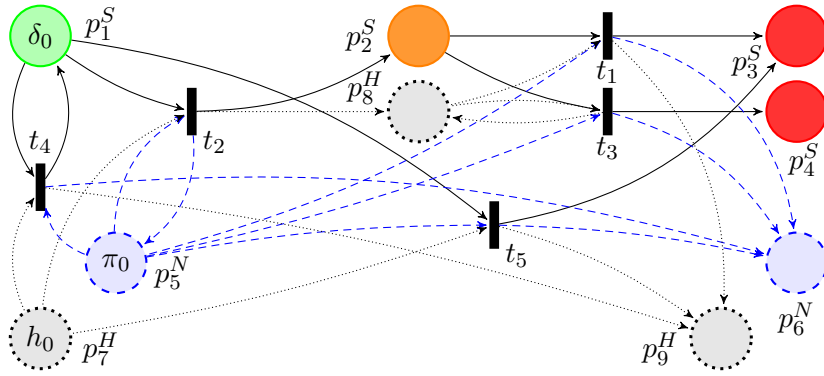
À ce stade de la génération, les annotations des arcs restent inchangées pour enregistrer l'ensemble des évènements simulés dans les ensembles d'évènements des configurations :

$$\mathcal{A}_{\Pi} = \mathcal{A}_{\Phi}. \quad (4.5)$$

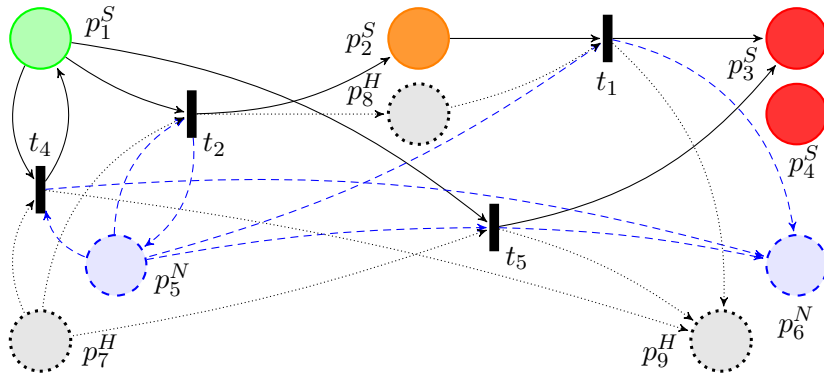
Exemple du robot mobile. Pour le robot mobile présenté en Section 2.2, $U^S = \{\text{éteindre}\}$.

Les conditions associées à t_4 restent les mêmes car elles représentent l'occurrence d'un évènement labellisé *éteindre*. Puisque que les fautes ne sont pas des évènements d'entrée, les conditions associées à t_1 , t_2 , t_3 et t_5 deviennent :

$$\begin{aligned} \Omega_{t_1} &= \langle \perp, \perp, d_k^1 > 0.9 \rangle, \\ \Omega_{t_2} &= \langle \perp, \perp, d_k^0 > 0.9 \rangle, \\ \Omega_{t_3} &= \langle \perp, \perp, \perp \rangle, \\ \Omega_{t_5} &= \langle \top, x_k^0 \leq 0, d_k^1 > 0.9 \rangle. \end{aligned}$$



(a) Modèle HPPN.



(b) Pronostiqueur HPPN après la fin de la génération.

FIGURE 4.1 – Illustration de la génération du pronostiqueur HPPN du robot mobile.

4.3.3 Suppression des transitions non tirables

Après ces modifications, si toutes les conditions de Ω_t sont \perp , c'est-à-dire qu'elles n'apportent aucune information intéressante pour la prédiction, la transition t est retirée du réseau :

$$T_{\Pi} \leftarrow (T \setminus \{t \in T_{\Pi} | \Omega_t = \langle \perp, \perp, \perp \rangle\}). \quad (4.6)$$

Les ensembles des arcs A_{Π} , des conditions Ω_{Π} , et des annotations \mathcal{A}_{Π} sont réduits en conséquence.

Exemple du robot mobile. La Figure 4.1 illustre la génération du pronostiqueur HPPN du robot mobile présenté en Section 2.2. Puisque les conditions Ω_{t_3} valent $\langle \perp, \perp, \perp \rangle$, t_3 est retirée du réseau, ce qui est cohérent car l'événement *wall*, provenant de l'environnement, est imprévisible et ne fait donc pas partie des entrées discrètes futures U_k^+ pour le pronostic.

4.3.4 Suppression des bruits de l'évolution de l'état

Cette étape est facultative suivant que l'on souhaite ou pas conserver les bruits liés aux dynamiques continues et hybrides.

Le bruit associé aux équations est supprimé des dynamiques continues, en accord avec la gestion des incertitudes (voir Section 4.6). Chaque ensemble $C_{p^N} \in \mathcal{C}_\Pi$ devient donc :

$$C_{p^N} \leftarrow \begin{cases} x_{k+1} = \mathbf{f}(k, x_k, u_k) \\ y_k = \mathbf{h}(k, x_k, u_k) \end{cases}, \quad (4.7)$$

où \mathbf{f} et \mathbf{g} sont dépendantes de la place numérique $p^N \in P_\Pi^N$ considérée.

Pour les mêmes raisons, chaque ensemble $H_{p^H} \in \mathcal{H}_\Pi$ devient :

$$H_{p^H} \leftarrow \left\{ d_{k+1} = \mathbf{g}(d_k, b_k, x_k, u_k) \right\}, \quad (4.8)$$

où \mathbf{g} est dépendante de la place hybride $p^H \in P_\Pi^H$ considérée.

4.4 Processus de pronostic

Le processus de pronostic peut-être exécuté à n'importe quel instant k et retourne un pronostic Π_k (Algorithme 2, ligne 7). Ses entrées sont le pronostiqueur $HPPN_\Pi$, le diagnostic courant Δ_k et l'ensemble U_k^+ des entrées futures du système au temps k .

Les étapes du processus de pronostic sont données dans l'Algorithme 3.

Algorithme 3 Pronostiquer

Entrée : $HPPN_\Pi, \Delta_k, U_k^+$

Sortie : $\hat{\mathbb{M}}_{k_{\text{EOP}}|k}$

- 1: *Initialiser*($HPPN_\Pi, \Delta_k$)
 - 2: **pour chaque** $U_\kappa \in U_k^+$ **faire**
 - 3: **si** $TousDéfaillants(HPPN_\Pi) = \perp$ **alors**
 - 4: $\hat{\mathbb{M}}_{\kappa|k} \leftarrow MettreÀJour(HPPN_\Pi, \kappa, U_\kappa)$
 - 5: **sinon**
 - 6: **sortie de la boucle pour**
 - 7: **fin si**
 - 8: **fin pour**
 - 9: $k_{\text{EOP}} \leftarrow \kappa$
-

Les sous-sections suivantes décrivent en détail les étapes du processus de pronostic.

4.4.1 Initialisation du pronostiqueur

Pour garder l'incertitude liée au diagnostic, le marquage initial du pronostiqueur $\mathbb{M}_{0\Pi} = \mathbb{M}_{k\Pi}$ est déterminé à partir du diagnostic Δ_k (donné par l'Équation 3.37) : la distribution des configurations, particules et jetons hybrides représentant toutes les hypothèses sur le système au temps k .

Les jetons de Δ_k sont dupliqués de telle sorte que chaque hypothèse de précision n est représentée dans le pronostiqueur par m hypothèses de précision 1, où $m \in \mathbb{N}_+$ est déterminé avec l'**algorithme stochastique de changement d'échelle (SSA)** (détaillé en Section 3.6).

Plus précisément, chaque hypothèse $\{\delta_k, \pi_k^1, \dots, \pi_k^n, h_k^1, \dots, h_k^n\}$ de Δ_k , où les n jetons hybrides lient les n particules à la configuration δ_k , est reproduite dans le marquage initial $\mathbb{M}_{k\Pi}$ du pronostiqueur et prend la forme d'une distribution équivalente $\{\delta_k^1, \pi_k^1, h_k^1, \dots, \delta_k^m, \pi_k^m, h_k^m\}$, dans laquelle, quel que soit $i \in \{1, \dots, m\}$, h_k^i lie δ_k^i et π_k^i , et chaque δ_k^i a la même valeur que δ_k , soit la valeur b_k .

Propriété 4.1. *Une fois initialisé avec Δ_k , le pronostiqueur possède autant de configurations que de particules et de jetons hybrides :*

$$|M_k^S| = |M_k^N| = |M_k^H|. \quad (4.9)$$

Lorsqu'une hypothèse est complètement reproduite, on a $m = n$: chaque particule π_k ou jeton hybride h_k de Δ_k est dupliqué dans le pronostiqueur. Cependant, le plus souvent, et dans le but d'améliorer les performances de calcul, les hypothèses dans Δ_k ne sont que partiellement reproduites : $m \leq n$.

L'algorithme SSA détermine les précisions m à associer à chaque hypothèse, en se basant sur leur degré de croyance, et trois paramètres d'échelle ρ_{Π}^{\min} , ρ_{Π}^{\max} et ρ_{Π}^{tot} . Ensuite, pour chaque hypothèse de Δ_k , les m particules et m jetons hybrides à dupliquer dans le pronostiqueur sont sélectionnés aléatoirement.

Les paramètres ρ_{Π}^{\min} et ρ_{Π}^{\max} représentent respectivement le nombre minimum et le nombre maximum de jetons de chaque type pour représenter une hypothèse dans le pronostiqueur. Cela signifie que la précision m d'une hypothèse est choisie pour satisfaire le prédicat $\rho_{\Pi}^{\min} \leq m \leq \rho_{\Pi}^{\max}$. Le paramètre ρ_{Π}^{tot} est le nombre total de jetons (de chaque type) disponibles pour représenter toutes les hypothèses. Après l'initialisation du pronostiqueur, le nombre total de jetons $|M_k|$ est donc toujours inférieur ou égal à $3 \times \rho_{\Pi}^{\text{tot}}$, car il y a trois types de jetons.

Les paramètres ρ_{Π}^{\min} et ρ_{Π}^{\max} sont respectivement les précisions minimale et suffisante accordées au pronostiqueur pour simuler l'évolution d'une hypothèse dans le futur. Les paramètres d'échelle offrent donc un compromis entre le nombre d'hypothèses à projeter dans le futur et la précision accordée à chacune, par rapport au nombre maximum de jetons à utiliser pour la simulation. Comme pour le processus de diagnostic, ils peuvent être choisis pour répondre à des contraintes de performance.

Exemple du robot mobile. L'initialisation du pronostiqueur du robot mobile présenté en Section 2.2 est illustré en utilisant le diagnostic à l'instant k suivant :

$$\Delta_k = [[\delta^1] \quad [\delta^2] \quad \emptyset \quad \emptyset \quad [\pi^1, \pi^2, \pi^3] \quad \emptyset \quad [h^1, h^2] \quad [h^3] \quad \emptyset],$$

contenant deux hypothèses. L'hypothèse $\{\delta^1, \pi^1, \pi^2, h^1, h^2\}$ considère que le système est dans le mode nominal $Nominal_1$, car les jetons sont dans les places p_1^S, p_5^N et p_7^H . L'hypothèse $\{\delta^2, \pi^3, h^3\}$ considère que la faute f_1 est survenue et que le système est dans le mode $Dégradé_1$ car les jetons sont dans les places p_2^S, p_5^N et p_8^H .

On choisit les paramètres d'échelle du pronostiqueur $(\rho^{min}, \rho^{max}, \rho^{tot}) = (2, 2, 2)$, ce qui permet de reproduire tous les jetons de toutes les hypothèses ($m = n$) dans le pronostiqueur. De Δ_k est donc généré $\mathbb{M}_{k\Pi}$:

$$\mathbb{M}_{k\Pi} = [[\delta^1, \delta^3] \quad [\delta^2] \quad \emptyset \quad \emptyset \quad [\pi^1, \pi^2, \pi^3] \quad \emptyset \quad [h^1, h^2] \quad [h^3] \quad \emptyset],$$

où h^1 lie δ^1 et π^1 , h^2 lie δ^3 et π^2 , et h^3 lie δ^2 et π^3 . La configuration δ^3 a la même valeur que δ^1 :

$$b_k^3 = b_k^1.$$

4.4.2 Processus du pronostiqueur

Une fois le pronostiqueur $HPPN_\Pi$ initialisé, son marquage évolue en accord avec les entrées futures $U_k^+ = \{U_\kappa | \kappa \in \{k+1, \dots, k+\tau_p\}\}$, où τ_p est l'horizon de prédiction.

Pour chaque paire d'entrées futures $U_\kappa = (U_\kappa^S, u_\kappa^N)$, le marquage $\hat{\mathbb{M}}_{\kappa|k}$ est déterminé en trois étapes à partir du marquage $\hat{\mathbb{M}}_{\kappa-1|k}$:

- Mise à jour des valeurs des configurations avec l'entrée discrète, détaillée en Section 4.4.2.1,
- Tirage des transitions, détaillé en Section 4.4.2.2,
- Mise à jour des valeurs des jetons, détaillée en Section 4.4.2.3,

Le processus est illustré dans la Figure 4.2.

La suite de cette section présente le détail des étapes du processus du pronostiqueur à partir du marquage $\mathbb{M}_{\kappa|k}$.

4.4.2.1 Mise à jour des valeurs des configurations avec l'entrée discrète

Si l'ensemble U_κ^S contient un label d'événement d'entrée $v \in E_\Pi$, les ensembles d'événements b_κ des configurations sont mis à jour avec l'événement (v, κ) :

$$\forall \delta_\kappa \in M^S, \quad b_\kappa \leftarrow b_\kappa \cup \{(v, \kappa)\}. \quad (4.10)$$

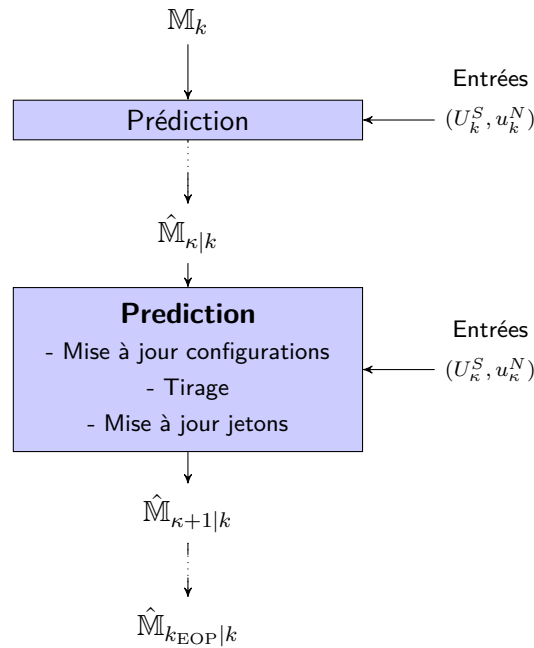


FIGURE 4.2 – Processus du pronostiqueur HPPN.

4.4.2.2 Tirage des transitions

Toutes les transitions tirables sont tirées simultanément selon des règles similaires à celles présentées dans la Section 2.3, à la différence près que l'ensemble \mathcal{S}_κ^t des jetons acceptés par une transition $t \in T$ au temps κ est déterminé de manière à respecter la condition logique $(\omega_t^S \wedge \omega_t^N) \vee \omega_t^H$.

Notons $M_\kappa(p)$ l'ensemble des jetons dans la place $p \in P$ au temps κ . Soient $(p^S, p^N, p^H) \in (P^S \cap {}^o t) \times (P^N \cap {}^o t) \times (P^H \cap {}^o t)$ et $\omega_t^S \in \Omega_t$, $\omega_t^N \in \Omega_t$, $\omega_t^H \in \Omega_t$. L'ensemble \mathcal{S}_κ^t est formellement défini par :

$$\mathcal{S}_\kappa^t = \left\{ (\delta_\kappa, \pi_\kappa, h_\kappa) \in (M_\kappa(p^S) \times M_\kappa(p^N) \times M_\kappa(p^H)) \mid \text{lien}(\delta_\kappa, \pi_\kappa, h_\kappa) \wedge ((\omega_t^S(\delta_\kappa) \wedge \omega_t^N(\pi_\kappa)) \vee \omega_t^H(h_\kappa)) \right\}, \quad (4.11)$$

où $\text{lien} : M(p^S) \times M(p^N) \times M(p^H) \rightarrow \mathbb{B}$ est une fonction booléenne qui renvoie VRAI si h_κ lie δ_κ et π_κ , et FAUX sinon.

Règle 4.2 (Transition tirable). *Une transition $t \in T$ est tirable (ou franchissable) au temps κ s'il existe au moins un jeton accepté par les conditions Ω_t :*

$$|\mathcal{S}_\kappa^t| > 0. \quad (4.12)$$

Règle 4.3 (Tirage d'une transition). *Soit $t \in T$ une transition tirable. Pour chaque type de place en entrée et en sortie de t , le tirage de $t \in T$ à l'instant κ est défini par :*

$\forall P^o \in \{P^S, P^N, P^H\}, p \in P^o \cap {}^a t, p' \in P^o \cap t^o,$

$$\begin{aligned} M_{\kappa+1}(p) &= M_{\kappa}(p) \setminus \mathcal{S}_{\kappa}^t(p), \\ M_{\kappa+1}(p') &= M_{\kappa}(p') \cup \mathcal{S}_{\kappa}^t(p), \end{aligned} \quad (4.13)$$

où $\mathcal{S}_{\kappa}^t(p)$ est l'ensemble des jetons de \mathcal{S}_{κ}^t qui sont dans la place p .

4.4.2.3 Mise à jour des valeurs des jetons

Pendant le tirage des transitions, les valeurs des configurations sont mises à jour avec les annotations des arcs \mathcal{A}_{Π} suivant la règle 4.4.

Règle 4.4 (Mise à jour de la valeur d'une configuration). *L'ensemble des événements b_{κ} d'une configuration déplacée à travers un arc a pendant le tirage d'une transition $t \in T$ au temps κ , est mis à jour avec le label d'événement $v \in E$ annotant a ; l'événement (v, κ) lui est ajouté :*

$$b_{\kappa+1} \leftarrow b_{\kappa} \cup \{(v, \kappa)\}. \quad (4.14)$$

Après le tirage des transitions, les valeurs des jetons hybrides puis celles des particules sont mises à jour en fonction des places auxquelles appartiennent les jetons et des entrées continues à l'instant κ . On rappelle que les bruits associés aux équations d'évolution des places numériques et hybrides ont été supprimés durant la génération du pronostiqueur.

Règle 4.5 (Mise à jour de la valeur d'un jeton hybride). *Après le tirage des transitions, le vecteur d'état hybride d_{κ} d'un jeton hybride est mis à jour avec l'équation d'évolution associée à la place hybride à laquelle le jeton hybride appartient (voir Équation 2.1), et les valeurs de la configuration et de la particule qu'il lie :*

$$d_{\kappa+1} = \mathbf{g}(d_{\kappa}, b_{\kappa+1}, x_{\kappa}, u_{\kappa}^N), \quad (4.15)$$

où u_{κ}^N est le vecteur des entrées continues à l'instant κ .

Utiliser $b_{\kappa+1}$ n'est pas un problème car l'ensemble $b_{\kappa+1}$ ne contient que des événements survenant strictement avant $\kappa + 1$ (voir Définition 4.4).

Règle 4.6 (Mise à jour de la valeur d'une particule). *Après le tirage des transitions, le vecteur d'état numérique x_{κ} d'une particule est mis à jour avec l'équation d'évolution associée à la place numérique à laquelle la particule appartient (voir Équation 2.2) :*

$$x_{\kappa+1} = \mathbf{f}(x_{\kappa}, u_{\kappa}^N), \quad (4.16)$$

où u_{κ}^N est le vecteur des entrées continues à l'instant κ .

Exemple du robot mobile. La Figure 4.3 illustre le processus du pronostiqueur du robot mobile à partir du marquage $\hat{M}_{\kappa|k}$ suivant :

$$\hat{M}_{\kappa|k} = [[\delta^1] \quad [\delta^2] \quad \emptyset \quad \emptyset \quad [\pi^1, \pi^2] \quad \emptyset \quad [h^1] \quad [h^2] \quad \emptyset],$$

où h^1 lie δ^1 et π^1 , et h^2 lie δ^2 et π^2 . Supposons qu'il n'y ait pas d'entrée discrète au temps κ , que la valeur de π^1 représente une vitesse non nulle, et que les valeurs de h^1 et h^2 représentent des probabilités d'occurrence de la faute f_2 respectivement de 0.8 et 0.95. Ainsi, seule la transition t_1 est tirable car les jetons δ^2 , π^2 , et h^2 sont acceptés par les conditions $\Omega_{t_1} = \langle \perp, \perp, d_{\kappa}^1 > 0.9 \rangle$.

Après le tirage des transitions, le marquage du pronostiqueur à l'instant $\kappa + 1$ est $\hat{M}_{\kappa+1|k}$:

$$\hat{M}_{\kappa+1|k} = [[\delta^1] \quad \emptyset \quad [\delta^2] \quad \emptyset \quad [\pi^1] \quad [\pi^2] \quad [h^1] \quad \emptyset \quad [h^2]],$$

qui contient deux hypothèses.

L'hypothèse $\{\delta^1, \pi^1, h^1\}$, considère qu'aucune faute ne surviendra jusqu'au temps κ . Les jetons sont dans les places p_1^S , p_5^N et p_7^H , ce qui permet de déduire que le système sera dans le mode *Nominal*₁ au temps $\kappa + 1$.

Dans l'hypothèse, $\{\delta^2, \pi^2, h^2\}$, la faute f_2 surviendra au temps κ . Les jetons sont dans les places p_3^S , p_6^N et p_9^H , ce qui permet de déduire que le système sera dans le mode *Défaillant*₁ au temps $\kappa + 1$.

Propriété 4.7 (Caractère borné du pronostiqueur). *Le pronostiqueur est initialisé avec au plus $3 \times \rho_{\Pi}^{tot}$ jetons (voir Section 4.4.1) et le processus du pronostiqueur ne crée aucun nouveau jeton lors de l'évolution du marquage.*

En conséquence, le pronostiqueur $HPPN_{\Pi}$ est $3 \times \rho_{\Pi}^{tot}$ -borné.

4.5 Pronostic

Le pronostic Π_k est le marquage du pronostiqueur HPPN à la fin du processus de pronostic :

$$\Pi_k \triangleq \hat{M}_{k_{EOP}|k}. \quad (4.17)$$

Propriété 4.8 (Fin de prédiction). *Quel que soit l'horizon de prédiction τ_p choisi, on a toujours :*

$$k_{EOP} \leq k + \tau_p. \quad (4.18)$$

Le pronostic Π_k est une distribution de croyances sur les futures modes du système jusqu'à l'instant k_{EOP} .

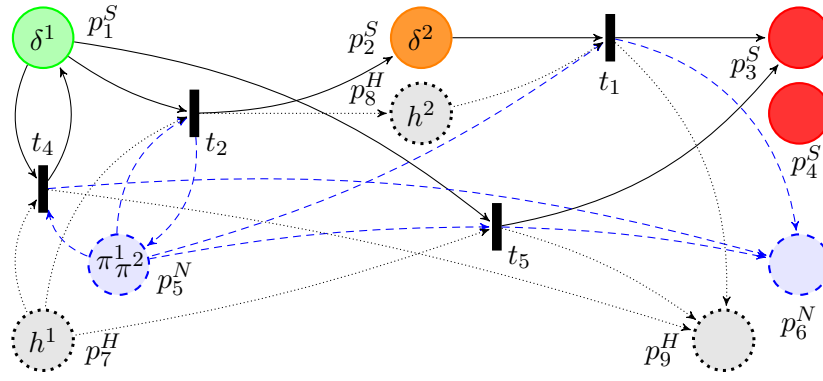
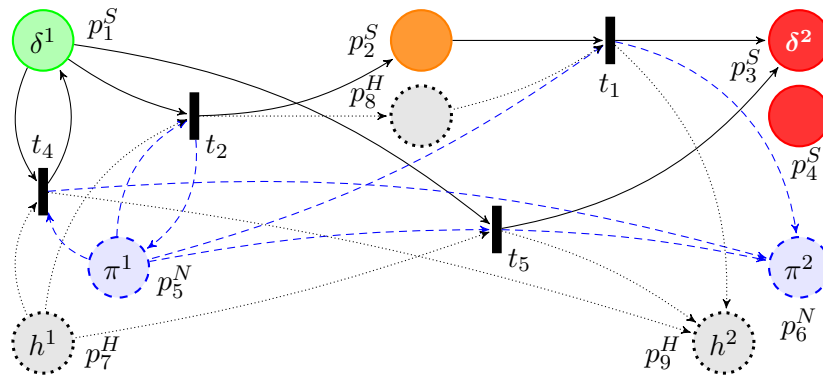

 (a) Pronostiqueur HPPN avec le marquage $\hat{M}_{\kappa|k}$.

 (b) Pronostiqueur HPPN avec le marquage $\hat{M}_{\kappa+1|k}$.

FIGURE 4.3 – Illustration du processus du pronostiqueur HPPN du robot mobile.

On peut notamment prédire, à partir de la connaissance de Π_k , les événements menant aux futurs modes de défaillance, en particulier les fautes et leurs dates d'occurrence.

Le pronostic Π_k contient toutes les informations nécessaires pour l'obtention d'une distribution de croyances sur le EOL et le RUL, puisqu'il contient les dates à partir desquelles chaque hypothèse sera dans un mode de défaillance.

Soit une hypothèse $\{\delta, \pi, h\}$ contenue dans le pronostic Π_k et soient p^S la place symbolique de la configuration δ , p^N la place numérique de la particule π et p^H la place hybride du jeton hybride h . Si les places p^S , p^N et p^H représente un mode de défaillance, alors le EOL de l'hypothèse est la date d'occurrence du dernier événement survenu pour cette hypothèse, i.e. la date la plus grande de tous les événements contenus dans l'ensemble d'événement b de la configuration δ :

$$EOL(\{\delta, \pi, h\}) \triangleq \max\{\kappa | (v, \kappa) \in b\}. \quad (4.19)$$

Avec le EOL d'une hypothèse, on peut retrouver son RUL en fonction du temps k à partir duquel à été lancé le calcul du pronostic :

$$RUL(\{\delta, \pi, h\}) \triangleq EOL(\{\delta, \pi, h\}) - k. \quad (4.20)$$

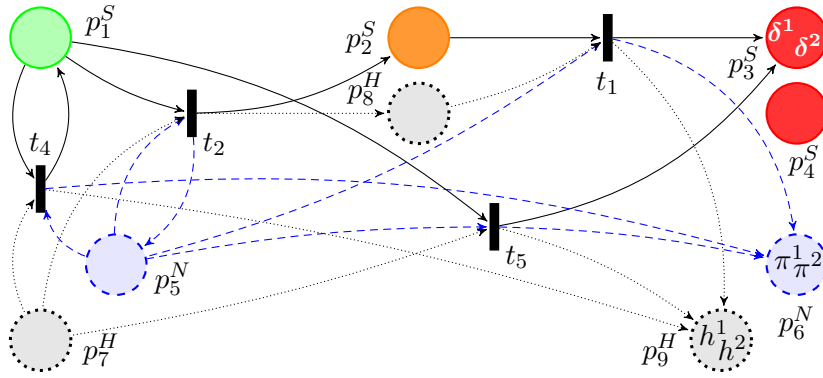


FIGURE 4.4 – Pronostiqueur HPPN du robot mobile à la fin du processus de pronostic avec le marquage $\hat{M}_{k_{EOP}|k}$.

Ainsi, les RUL/EOL des hypothèses couplés aux scores de chacune d'entre elles calculés avec l'Équation 3.34 forme la distribution de croyances sur le RUL/EOL du système.

Exemple du robot mobile. La Figure 4.4 présente le pronostic Π_k du robot mobile à la fin du processus qui contient deux hypothèses :

$$\Pi_k = [\emptyset \ \emptyset \ [\delta^1, \delta^2] \ \emptyset \ \emptyset \ [\pi^1, \pi^2] \ \emptyset \ \emptyset \ [h^1, h^2]],$$

avec $k = 100$. Les deux hypothèses $\{\delta^1, \pi^1, h^1\}$ (ayant un score de 0.1) et $\{\delta^2, \pi^2, h^2\}$ (ayant un score de 0.8) considèrent que le système entrera dans le mode *Défaillant*₁ (places p_3^S , p_6^N , et p_9^H) durant l'horizon de prédiction $\tau_p = 300$. L'ensemble d'événement de δ^1 vaut $b^1 = \{(f_1, 120), (f_2, 200)\}$ et celui de δ^2 vaut $b^2 = \{(f_2, 230)\}$. On peut donc en déduire $\{(200, 0.1), (230, 0.8)\}$, la distribution de croyances sur le EOL du système au temps k et $\{(100, 0.1), (130, 0.8)\}$, la distribution de croyances sur son RUL au temps k .

4.6 Gestion des incertitudes

L'incertitude dans une méthode de pronostic provient principalement de trois sources :

- la connaissance de l'état courant du système,
- les bruits liés à l'évolution de l'état du système,
- la connaissance des futures entrées sur le système.

Considérer toutes ces sources d'incertitude permet de prédire un maximum de trajectoires possibles pour le système.

4.6.1 Connaissance de l'état courant

Beaucoup d'études considèrent l'estimation de l'état continu courant comme le point de départ d'un processus de pronostic [Daigle *et al.* 14a, Jouin *et al.* 16] et éventuellement du mode

courant dans le cadre des systèmes hybrides [Daigle *et al.* 15a]. Dans notre méthodologie, le processus de pronostic se base sur le diagnostic HPPN, qui contient une distribution de croyances sur les états discret, continu et hybride, mais aussi sur l'ensemble des événements survenus sur le système, nécessaire pour simuler l'évolution de l'état hybride.

4.6.2 Évolution de l'état dans le futur

Intuitivement, ignorer le bruit sur l'évolution de l'état du système empêche l'exploration de toutes les trajectoires possibles. Cependant, la question est discutable, tant pour la dynamique discrète du modèle, que pour les dynamiques continue et hybride.

Concernant la partie discrète du modèle, considérer à chaque instant que n'importe quel événement peut survenir provoque une explosion combinatoire. Lors du processus de pronostic, par manque d'observations, cette explosion combinatoire ne peut être corrigée comme cela est fait durant le processus de diagnostic avec les observations.

Concernant la partie continue du modèle, la gestion des bruits liés à l'évolution de l'état du système est inspirée des approches de diagnostic de systèmes continus avec des estimateurs (filtres de Kalman ou filtres particulaires, etc.). Cependant, ce type d'approche peut retourner des résultats de pronostic non concluants si les bruits liés aux dynamiques des paramètres inconnus sont trop importants. Il peut donc être intéressant de retirer ces bruits lors du processus de pronostic.

Concernant la partie hybride du modèle, elle ne peut par construction pas être corrigée grâce à des observations : elle est donc intrinsèquement incertaine.

Pour éviter une explosion combinatoire, l'occurrence d'un événement à n'importe quel instant n'est pas envisagé durant le processus du pronostiqueur. En contraste avec le processus du diagnostiqueur, les transitions ne sont donc pas pseudo-tirées. En revanche, les entrées discrètes sont simulées et les conditions sur l'état hybride sont utilisées pour simuler les futurs changements de mode. Pour les cas où les bruits liés aux évolutions des états continu et hybride sont trop important, nous proposons également une étape optionnelle durant la génération du pronostiqueur qui permet de les supprimer des dynamiques continue et hybride.

4.6.3 Connaissance des entrées futures

Dans un cas réel, déterminer l'ensemble des futures entrées U_k^+ est une autre source d'incertitudes. Les commandes de bas niveau sont difficiles à prédire précisément, même si le plan de mission de haut niveau est connu. Considérer du bruit dans le plan de mission et dans les futures commandes est une solution à ce problème [Daigle *et al.* 15b].

Il est également possible que les futures entrées soit entièrement inconnues. Ceci est un problème uniquement si le temps n'influence ni l'évolution de l'état continu, ni celle de l'état hybride du système. Une solution consiste à propager les entrées courantes en première approximation. Une autre solution consiste à générer un ensemble d'entrées à partir des entrées précédentes.

4.7 Cas d'étude : les réservoirs d'eau

Dans cette partie, le pronostiqueur HPPN du système des réservoirs d'eau présenté en Section 2.5 est généré.

Différents scénarios sont ensuite testés pour illustrer comment le pronostiqueur HPPN exploite le diagnostic courant, le modèle de comportement et le modèle de dégradation du système pour simuler l'évolution de l'état du système dans le futur.

4.7.1 Pronostiqueur HPPN

À partir du modèle HPPN spécifié en Section 2.5.2, le pronostiqueur HPPN du système a été généré. Il est présenté en Figure 4.5. On peut voir qu'il possède les mêmes 19 places que le modèle HPPN.

Le pronostiqueur possède également le même nombre de transitions que le modèle pour les raisons suivantes :

- les événements $ouvr_{v_{13}}$, $ferm_{v_{13}}$, $ouvr_{v_{32}}$ et $ferm_{v_{32}}$ sont des commandes discrètes qui peuvent apparaître dans l'ensemble des entrées futures,
- le passage en dessous du seuil l_{2min} est un indicateur de la faute f_0 ,
- les probabilités d'occurrences de toutes les autres fautes sont estimées dans l'état hybride.

Ainsi tous les changements de mode sont prédictibles et aucune transition n'est enlevée durant la génération du pronostiqueur.

Lors de la génération du pronostiqueur, les bruits liés aux évolutions continues et hybrides sont supprimés du modèle, i.e. l'étape présentée en Section 4.3.4 est exécutée.

Rappel

Le pronostiqueur utilise trois paramètres d'échelle ρ_{Π}^{min} , ρ_{Π}^{max} et ρ_{Π}^{tot} . Les paramètres ρ_{Π}^{min} et ρ_{Π}^{max} sont respectivement les précisions minimale et suffisante accordées au pronostiqueur pour simuler l'évolution d'une hypothèse dans le futur. Le paramètre ρ_{Π}^{tot} est quant à lui représentatif du nombre maximum de jetons à utiliser pour simuler l'évolution de

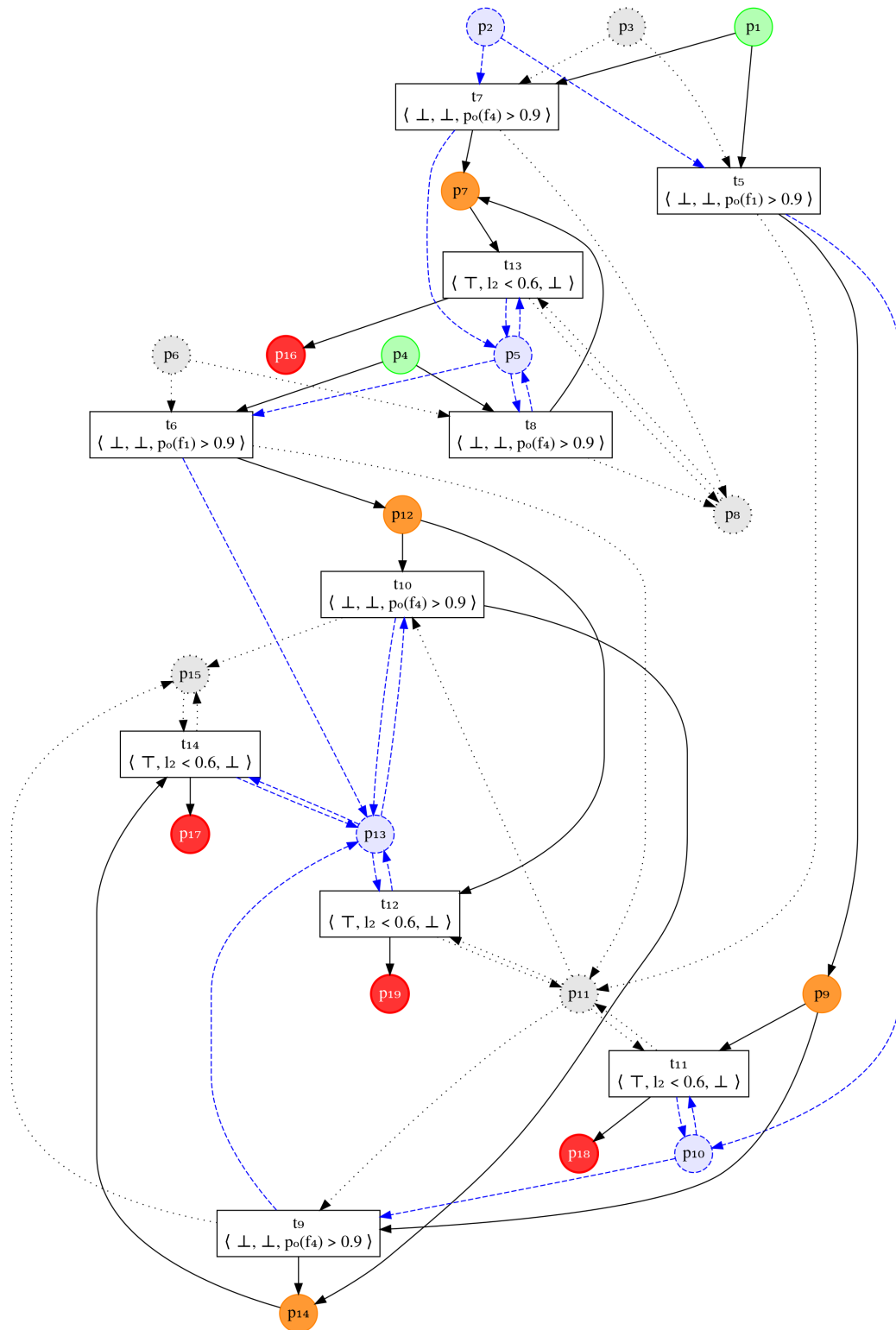


FIGURE 4.5 – Pronostiqueur HPPN du système des réservoirs d'eau.

toutes les hypothèses réunies.

Le processus de pronostic simule l'évolution des hypothèses pendant un horizon de prédiction τ_p .

On souhaite simuler les évolutions d'un grand nombre d'hypothèses avec peu de précision. On fixe donc les paramètres d'échelle du pronostiqueur à :

$$(\rho_{\Pi}^{min}, \rho_{\Pi}^{max}, \rho_{\Pi}^{tot}) = (1, 3, 50)_{\Pi}.$$

Ainsi, les évolutions de 16 à 50 hypothèses seront simulées avec des précisions variants entre 1 et 3.

4.7.2 Résultats

Nous définissons trois scénarios d'étude. Les conditions initiales du système sont les mêmes que dans la Section 3.8 et sa mission est la même. Le débit de l'eau versée dans le réservoir $Tank_1$, $q_{1\kappa} = 0.0000023 \text{ m}^3 \cdot \text{s}^{-1}$, reste constant dans le temps et après 310 minutes de fonctionnement, la vanne v_{13} est fermée toutes les heures pendant 20 minutes pour effectuer un traitement de l'eau du réservoir $Tank_1$.

Pour chaque scénario, au bout d'un certain temps, le réservoir $Tank_1$ se met à fuir (faute f_1) et le niveau d'eau du réservoir $Tank_2$ passe sous le seuil l_{2min} (faute f_0) quelques instants après. L'occurrence de la faute f_1 est simulée avec les lois de vieillissement associées aux dynamiques hybrides du modèle HPPN.

Enfin, le pronostiqueur calcule un pronostic toutes les 30 minutes et l'horizon de prédiction est fixé à $\tau_p = 600000\text{s}$, soit 10000 périodes d'échantillonnage.

4.7.2.1 Scénario 1

Dans ce scénario, on se replace dans le contexte du scénario 1 décrit en Section 3.8.2.1. L'objectif est de présenter les résultats de pronostic sous la forme de distributions de croyances sur la durée de vie résiduelle (RUL) du système et sur sa future trajectoire de modes.

Par ailleurs, on suppose que les futures entrées sont connues précisément : le débit q_1 restera constant et la vanne v_{13} sera fermée toutes les heures pendant 20 minutes à partir de 310 minutes après le début de la mission. On s'attend donc à ce que les estimations du EOL retournées par le pronostiqueur soient proches mais antérieures au EOL du système simulé.

Dans ce scénario, la faute f_1 survient à 201840s et f_0 survient à 206040s. La fin de vie du système simulé est donc 206040s.

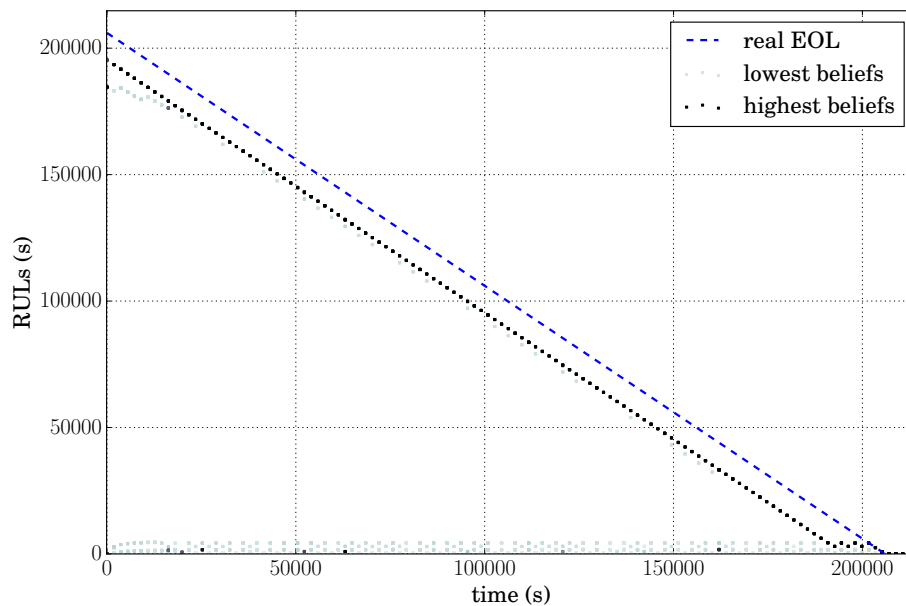


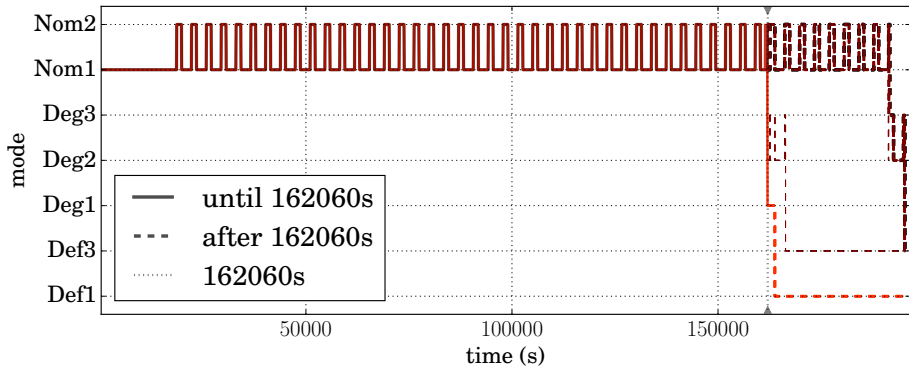
FIGURE 4.6 – Estimations du RUL du système pour le scénario 1.

La Figure 4.6 montre les estimations du RUL du système tout au long du scénario. Les RUL des hypothèses sont dessinés avec des nuances de gris ; celles avec les degrés de croyance les plus élevés sont noires, tandis que celles avec des degrés de croyance faibles sont en gris clair. La fin de vie de du système simulé (EOL) est représentée par une ligne discontinue.

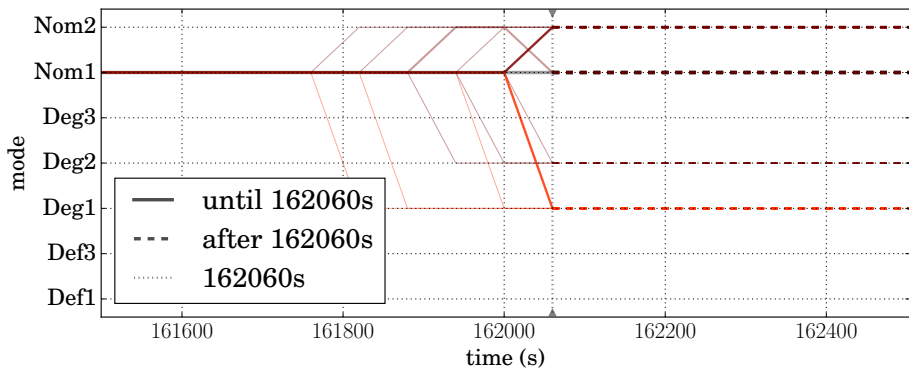
Dans la figure, on peut voir que le EOL ayant les degrés de croyance les plus élevés survient approximativement 10800 secondes (soit presque 3 heures) avant le EOL du système simulé, et que cet écart est conservé jusqu'à 192840s. Ces résultats correspondent donc à nos attentes.

On remarque aussi que le pronostiqueur estime un RUL plus court (entre 460s et 4300s) pour certaines hypothèses ayant des degrés de croyance faibles. Ces hypothèses sont celles qui considèrent que le mode courant du système est un mode dégradé (voir Section 3.8.2.1). À certains instants particuliers, le degré de croyance de ces hypothèses est plus élevé, ce qui engendre la présence de RUL courts ayant des degrés de croyance élevé. C'est le cas par exemple à 16380s, 20060s, 25420s, 52420s, 63160s ou encore à 162060s.

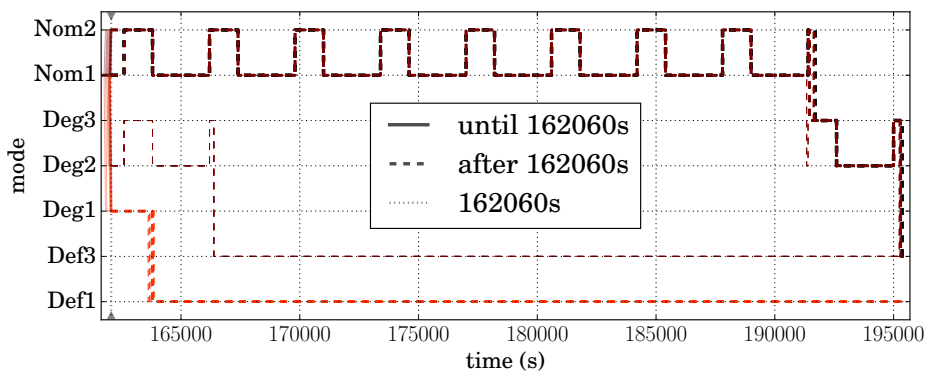
Dans un contexte de gestion de santé, une estimation en temps réel des trajectoires passée et future du système peut être plus intéressante que l'estimation du RUL/EOL. La Figure 4.7(a) expose la distribution de croyances sur les trajectoires passée et future du système à 162060s en combinant les hypothèses du diagnostic avec celles du pronostic. Elle détaille les séquences de modes par lesquelles le système va atteindre un mode de défaillance et à quel instant cela va se produire.



(a) Vue d'ensemble



(b) Zoom entre 161500s et 162500s



(c) Zoom entre 161500s et 195120s

FIGURE 4.7 – Distribution de croyances sur les trajectoires passée et future du système à 162060s pour le scénario 1.

La Figure 4.7(b) est un zoom de la Figure 4.7(a) entre 161500s et 162500s. Il est possible de grouper les hypothèses en fonction du mode qu'elles considèrent être le mode du système à 161500s. On distingue donc quatre groupes, un pour chaque mode parmi Nom_2 , Nom_1 , Deg_2 et Deg_1 .

Dans la Figure 4.7(c), qui est un zoom de la Figure 4.7(a) entre 161500s et 195120s, on peut voir que les trajectoires futures des hypothèses dans les modes Nom_1 et Nom_2 se rejoignent à 162600s. Le pronostiqueur considère donc que si le système est dans le mode Nom_1 ou Nom_2 à 162600s, il va continuer à osciller entre les deux modes. La faute f_1 surviendra alors entre 191340s et 191700s et le système entrera dans un mode défaillance entre 195240s et 195360s. Si le système est dans le mode Deg_2 (i.e. que la faute f_1 est déjà survenue), le pronostiqueur prédit que le système entrera dans un mode de défaillance à 166380s. De la même manière, si le système est dans le mode Deg_1 (la faute f_4 est survenue), il entrera dans un mode de défaillance entre 163620s et 163860s.

4.7.2.2 Scénario 2

Ce scénario est le même que le scénario 1 à l'exception près que les futures entrées sont supposées ne pas être connues. L'objectif est de montrer qu'il est possible de lancer un pronostic dans un tel cas bien que les résultats ne soient pas aussi satisfaisants que lorsque les entrées futures sont connues.

Puisque le pronostiqueur ne connaît pas les futures entrées, il ne peut pas simuler les changements de mode du système qui sont conditionnés uniquement par l'occurrence d'un événement discret. Ainsi il ne pourra pas simuler les passages dans le mode Nom_2 et tenir compte du niveau de stress plus accentué lié à ce mode. On s'attend donc à obtenir des EOL plus tardifs que dans le scénario 1.

La Figure 4.8 montre les estimations du RUL du système tout au long du scénario 2. On peut voir que jusqu'à ce que la vanne v_{13} commence à être régulièrement fermée, le EOL ayant le degré de croyance le plus élevé survient approximativement à 235000s, soit 28960 secondes (environ 8 heures) après le EOL du système simulé.

À partir de 18600s (lorsque la vanne commence à être régulièrement fermée), le EOL qui a le degré de croyance le plus élevé survient de plus en plus tôt car le diagnostiqueur estime que le système passe effectivement de plus en plus de temps dans le mode Nom_2 dans lequel le système est plus stressé que dans le mode Nom_1 .

Ces résultats correspondent à nos attentes vis-à-vis de l'ignorance des futures entrées du système. Ils illustrent aussi que l'état hybride est correctement estimé par le diagnostiqueur et que le pronostiqueur se sert de cette estimation pour la propager dans le futur.

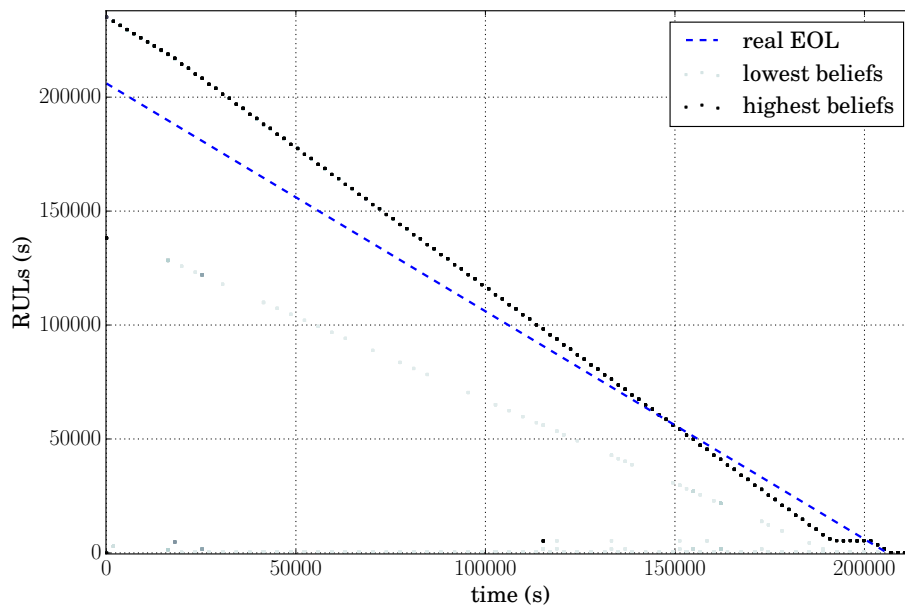


FIGURE 4.8 – Estimations du RUL du système pour le scénario 2.

4.7.2.3 Scénario 3

Dans ce scénario, on se replace dans le contexte du scénario 3 décrit en Section 3.8.2.3. L'objectif est d'illustrer l'influence des résultats de diagnostic sur la fonction de pronostic.

La faute f_1 survient à 209700s et f_0 survient à 213360s. La fin de vie du système simulé est donc 213360s.

Dans la Figure 4.9, qui montre les estimations du RUL du système tout au long du scénario 3, on retrouve des résultats similaires à ceux du scénario 1 (voir Figure 4.6). On remarque néanmoins que les hypothèses qui considèrent que le RUL du système est court ont des degrés de croyance plus élevés que dans le scénario 1. Ces résultats sont prévisibles car les diagnostics pour ce scénario sont plus incertains que ceux du scénario 1. Ceci met en évidence l'influence du diagnostic courant du systèmes sur les résultats du pronostiqueur.

Ces trois scénarios illustrent la manière dont le pronostiqueur utilise les informations contenues dans le diagnostic courant et la connaissance sur les entrées futures du système, ainsi que la façon dont il propage dans le futur les incertitudes qui leurs sont associées.

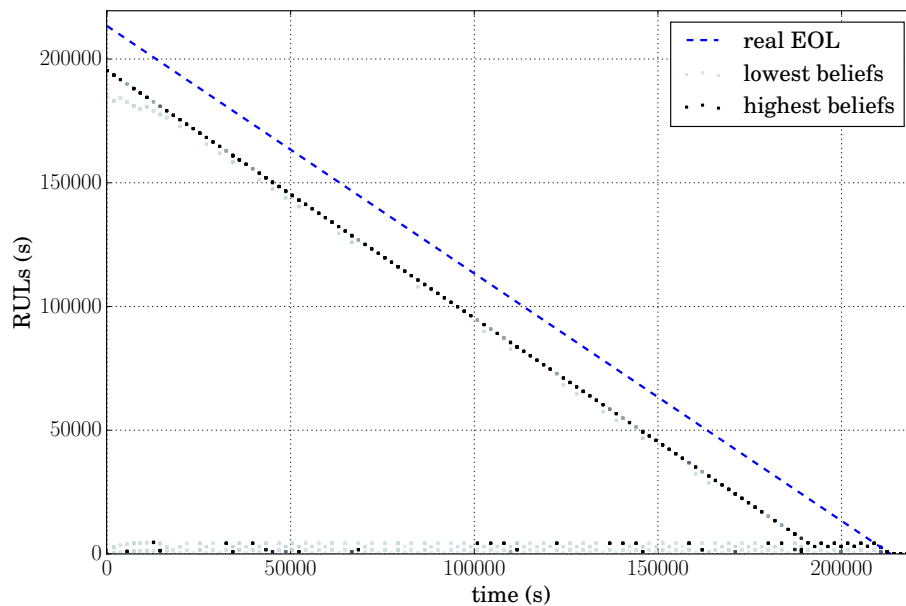


FIGURE 4.9 – Estimations du RUL du système pour le scénario 3.

4.8 Vers une intégration avancée des méthodes de diagnostic et de pronostic

Dans ces travaux, le formalisme des HPPN est utilisé pour interfacier les modules de diagnostic et de pronostic, via notamment la représentation de la connaissance sur l'état du système sous incertitude. Dans la Section 4.2, nous proposons une intégration qui consiste à utiliser le diagnostic courant comme point de départ du processus de pronostic.

Dans cette section, on se propose d'intégrer les deux modules d'une façon plus approfondie, de manière à ce que les résultats d'un module puissent influencer le processus de l'autre.

En ce qui concerne l'influence du diagnostic courant du système sur le processus de pronostic, il peut être pertinent, par exemple, de n'effectuer seulement la prédiction des hypothèses qui estiment que le système est dans un mode dégradé ou bien qui considèrent qu'il a récemment changé de mode. Le pronostiqueur projeterait alors moins d'hypothèses dans le futur mais avec des précisions plus importantes.

Concernant l'influence du pronostic courant du système sur le processus de diagnostic, il peut être intéressant que le diagnostiqueur accorde plus de précision aux hypothèses ayant les RUL les plus courts ou encore des RUL qui remettent en cause la réalisation de la mission. Le diagnostiqueur aurait ainsi une meilleure estimation de ces hypothèses et pourrait potentiellement les incriminer ou les discriminer plus rapidement.

Dans le cadre de la gestion de santé, un tel rapprochement des deux méthodes est propice à l'obtention de meilleurs résultats de surveillance de santé, et donc à de meilleures prises de décision concernant le système. Les fonctions d'influences dépendent cependant davantage des attentes de la méthode de surveillance que de la qualité des techniques employées pour réaliser cette surveillance. La définition de ces fonctions d'influences relève donc des objectifs relatifs à la gestion de santé du système étudié.

L'idée ici n'est pas de définir des fonctions d'influence mais de montrer qu'elles peuvent être intégrées de manière cohérente dans le cadre de la surveillance de santé basée sur les HPPN. Nous proposons de rajouter une étape dans chacun des processus durant laquelle la fonction d'influence transforme la distribution des scores juste avant l'utilisation de l'algorithme SSA. L'étape ajoutée au processus de diagnostic influence le rééchantillonnage des jetons dans le diagnostiqueur et donc la précision de chacune des hypothèses. Celle ajoutée au processus de pronostic influence la reproduction des hypothèses dans le pronostiqueur et donc leurs précisions lors de leurs projection dans le futur.

On distingue par la suite la fonction d'influence du diagnostic courant du système sur le processus de pronostic ($\Delta \rightarrow \Pi$) de celle de l'influence du pronostic courant du système sur le processus de diagnostic ($\Pi \rightarrow \Delta$). Ces deux fonctions d'influence sont mises en évidence dans la Figure 4.10, dans laquelle est présentée l'intégration avancée des méthodes de diagnostic et de pronostic basées sur les HPPN.

On peut voir dans la figure que le processus de diagnostic utilise les observations consécutives sur le système pour réaliser les étapes de prédiction et de correction et déterminer le diagnostic courant. En fonction des scores associés à chaque hypothèses et de la fonction d'influence $\Delta \rightarrow \Pi$, le pronostiqueur peut ainsi être initialisé de manière pertinente vis-à-vis des objectifs des objectifs fixés dans le cadre de la gestion de santé du système. Le pronostiqueur utilise ensuite les entrées futures sur le système pour réaliser les étapes de prédiction et déterminer le pronostic courant. En fonction des scores des hypothèses calculés avec les nouvelles observations et de la fonction d'influence $\Pi \rightarrow \Delta$, le diagnostiqueur peut alors déterminé les nouvelles précisions des hypothèses de manière pertinente vis-à-vis des objectifs relatifs à la gestion de santé du système. Le nouveau diagnostic calculé va pouvoir influencer le calcul du prochain pronostic, et ainsi de suite.

Dans le cadre de la surveillance de santé basée sur les HPPN, cette intégration est avantageuse pour plusieurs raisons. Premièrement, les résultats de l'un des deux modules ne sont pas utilisés pour interpréter directement les résultats de l'autre mais pour influencer la précision avec laquelle ils vont être déterminés. Deuxièmement, les fonctions d'influence $\Delta \rightarrow \Pi$ et $\Pi \rightarrow \Delta$ sont intégrées de manière cohérente vis-à-vis du déroulement des processus de diagnostic et pronostic et de leurs gestions des incertitudes. Enfin, la mise en places des fonctions d'influence $\Delta \rightarrow \Pi$ et $\Pi \rightarrow \Delta$ est générique vis-à-vis du type d'objectif à réaliser, ce qui pourrait mener

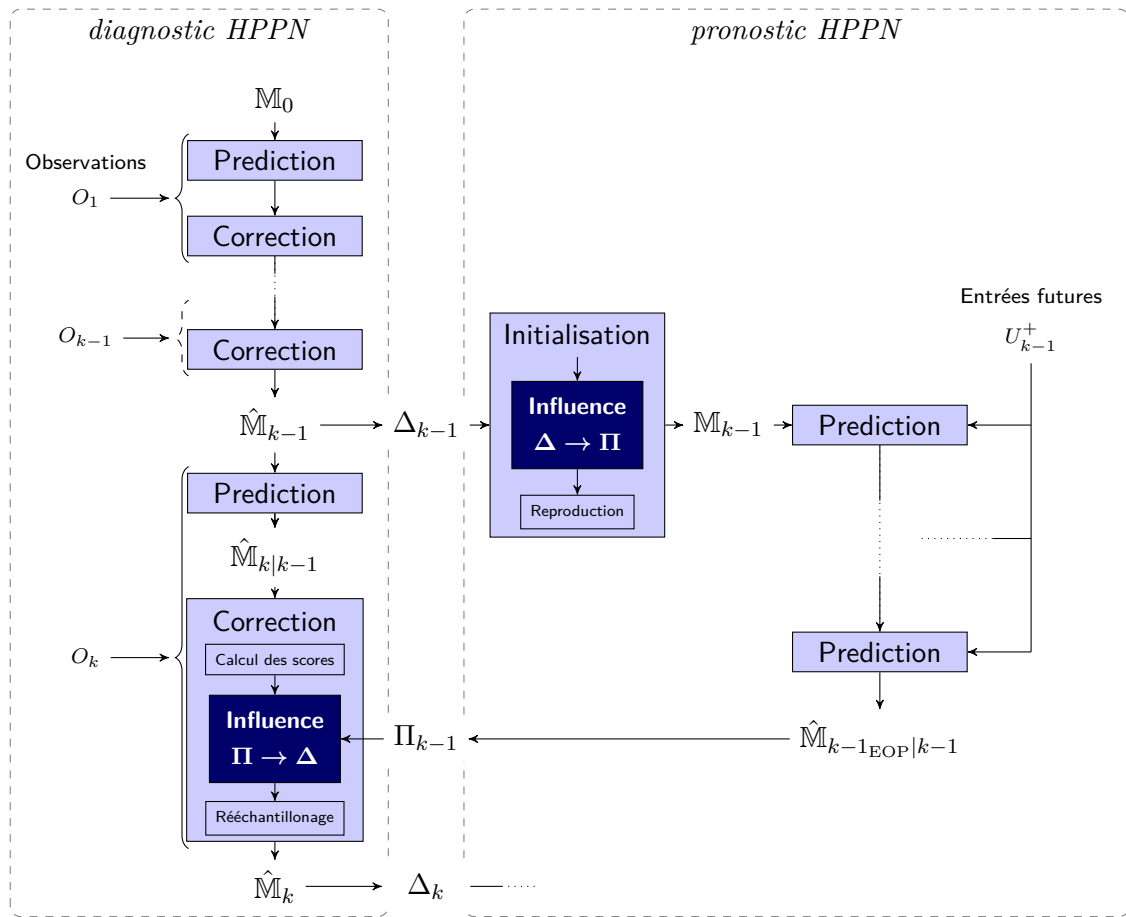


FIGURE 4.10 – Intégration avancée des méthodes de diagnostic et de pronostic basée sur les HPPN.

à une réflexion quant à l'exploitation des résultats de surveillance basée sur les HPPN dans le cadre de la gestion de santé des systèmes.

4.9 Conclusion

Dans ce chapitre, nous proposons dans un premier temps une méthodologie de surveillance qui intègre les méthodes de diagnostic et de pronostic basées sur les HPPN. L'intégration consiste à utiliser le diagnostic courant (déterminé par le diagnostiqueur HPPN) pour calculer le pronostic courant à l'aide d'un ensemble de futures entrées qui correspondent au plan de mission du système.

La méthode de pronostic repose sur la génération d'un pronostiqueur HPPN à partir du modèle HPPN du système. Le pronostiqueur HPPN prédit l'évolution du système dans le futur sur un certain horizon de prédiction. Le pronostiqueur est initialisé avec le diagnostic courant du système et l'algorithme SSA, puis son marquage évolue en fonction des futures entrées sur le système. Le pronostic courant du système est le marquage du pronostiqueur à la fin de la

prédiction. Il porte une distribution de croyances sur les futurs états possibles et les séquences de fautes menant à ces états.

La méthodologie de pronostic a également été appliquée sur le système des réservoirs d'eau qui a été décrit et modélisé dans la Section 2.5. Le pronostiqueur HPPN a été généré puis des résultats de la méthode sur différents scénarios ont illustrés la manière dont le pronostiqueur exploite le diagnostic courant et la connaissance sur les futures entrées du système, mais aussi la manière dont les incertitudes sont propagées dans le futur.

Enfin, la dernière section de ce chapitre propose une intégration avancée des méthodes de diagnostic et de pronostic basées sur les HPPN. Cette intégration propose d'utiliser le diagnostic pour influencer le processus de pronostic, ainsi que le pronostic pour influencer le processus de diagnostic, dans le but d'améliorer les résultats de la surveillance et de répondre à des objectifs relatifs à la gestion de santé du système.

Chapitre 5

Cas d'étude : le rover K11

Résumé

Dans ce chapitre, la méthodologie de surveillance de santé basée sur les HPPN présentée dans les chapitres précédents est appliquée au K11, un prototype de rover planétaire.

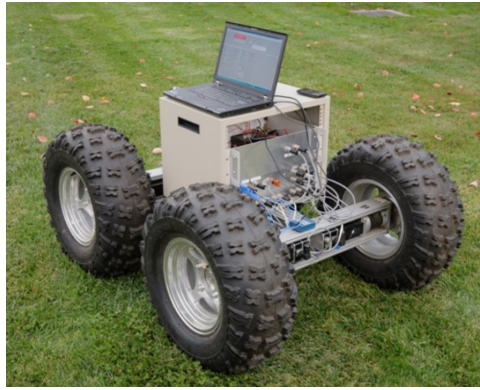
Le K11 est un rover alimenté par une batterie possédant de nombreux capteurs ainsi que quatre roues motorisées et contrôlables. C'est actuellement un banc d'essai de méthodes de diagnostic et de pronostic mis en place par le NASA Ames Research Center sur lequel il est possible de provoquer ou de simuler des fautes et des défaillances.

Dans ces travaux, le K11 est étudié comme un système en opération réalisant des missions d'exploration et exposé à des défaillances techniques, comme des pannes de capteurs, l'apparition de charges parasites sur sa batterie ou des surchauffes de ses moteurs.

Dans un premier temps, le K11 est présenté en détail et l'évolution de sa santé est décrite à l'aide d'un système multimode. Puis, le système est modélisé avec un HPPN et ses diagnostiqueur et pronostiqueur sont générés.

La méthodologie de surveillance de santé est ensuite testée sur trois scénarios, mettant en avant la robustesse de l'approche face aux données réelles et l'utilisabilité des résultats.

Enfin, les performances calculatoires des fonctions de diagnostic et de pronostic sont données et discutées. L'utilité de l'algorithme SSA utilisé par les deux fonctions et de ses paramètres d'échelle ajustables est alors mise en évidence.

FIGURE 5.1 – Le rover K11 [Balaban *et al.* 13].

5.1 Introduction

Dans ce chapitre, on se propose d'appliquer la méthodologie de surveillance au K11, un prototype de rover planétaire. Les objectifs sont de montrer la robustesse des méthodes de diagnostic et de pronostic aux données et aux contraintes réelles, ainsi que l'utilisabilité des résultats.

Le K11 est un rover à quatre roues conçu pour tester les architectures économes en énergie des rovers, dans les conditions de l'Antarctique [Lachat *et al.* 06]. Sa conception a ensuite été repensée par le NASA Ames Research Center [Balaban *et al.* 13] à des fins de diagnostic et de pronostic [Daigle *et al.* 14b], et aussi de replanification à partir du pronostic [Sweet *et al.* 14]. Il s'agit maintenant d'un banc d'essai sur lequel il est possible de provoquer (ou simuler) des fautes et des défaillances.

Dans ces travaux, il est étudié comme un système en opération exposé à des défaillances et réalisant des missions. Les entrées du rover sont ses commandes continues et ses sorties sont les mesures continues des capteurs. Le rover n'a pas d'entrée ou de sortie discrète.

Cette étude a été réalisée en collaboration avec la NASA dans le cadre d'une mobilité de trois mois au NASA Ames Research Center.

5.2 Description

Le rover K11 (voir la Figure 5.1), est alimenté par vingt-quatre batteries unicellulaires au lithium-ion de 2.2 Ah. Une mission type du rover consiste à visiter un ensemble de positions (points d'intérêts) pour y effectuer des expériences scientifiques, avant de revenir à sa station de charge. Un module de prise de décision (DM) détermine les positions à visiter et l'ordre de visite, en fonction du terrain, des récompenses associées aux expériences à réaliser, et des conditions du rover. Les quatre roues du K11 sont libellées par leurs localisations : la roue avant gauche (FL pour *front-left*), la roue avant droite (FR pour *front-right*), la roue arrière

gauche (BL pour *back-left*) et la roue arrière droite (BR pour *back-right*). Chaque roue est entraînée par un moteur à balai indépendant commandé par un contrôleur de mouvement numérique mono-axe. Un ordinateur de bord exécute le logiciel de contrôle et d'acquisition des données. Le rover est un véhicule à direction différentielle, c'est à dire que les roues ne sont pas orientables et qu'il pivote lorsque les roues des côtés droit et gauche sont commandées par des vitesses différentes. Le module de gestion d'énergie fournit la recharge de la batterie et des fonctions d'équilibrage de charge. Il envoie aussi à l'ordinateur de bord les mesures de tension et de température de chacune des cellules de la batterie. Le module d'acquisition des données collecte les mesures du courant et des températures des moteurs et les envoie à l'ordinateur de bord. Les contrôleurs des moteurs revoient des données de mouvement, telles que les vitesses commandées et les vitesses réelles. Plus de détails concernant la description du rover sont fournis dans [Balaban *et al.* 13].

Toutes les observations continues sur le rover, ainsi que la liste des fautes considérées dans cette étude, sont présentées dans la Table 5.1. Quatre signaux commandent les roues avec un contrôleur PID (proportionnel, intégral, dérivée) et l'ensemble des capteurs renvoie 61 signaux de mesure. Plusieurs types de faute liés au système d'énergie (batterie), au système électromécanique (moteurs, contrôleurs), et aux capteurs (dérives, biais, défaillances), ont été implémentées sur ce banc d'essai.

Le K11 n'a pas d'actionneur ni de capteur discret et a donc principalement été étudié comme un système continu, pour lequel les fautes étaient définies par des contraintes sur l'état continu (dépassements de seuil). On se propose d'abstraire les fautes anticipées en événements non observables. Une description simplifiée de l'évolution de la santé du K11 est présentée dans la Figure 5.2. Les modes de fautes multiples n'y sont pas inclus et seulement le moteur avant gauche FL est considéré.

Tant qu'aucune faute ne survient, le rover est dans le mode $Nominal_1$ avec la dynamique continue C_1 .

L'occurrence de la faute f_1 représente la **fin de la décharge** (*End Of Discharge* ou EOD) de la batterie, i.e. la date à laquelle la batterie est trop déchargée pour alimenter le système. Cette date est supposée atteinte lorsque la tension de la batterie est inférieure à 3 V et mène à l'échec de la mission (mode $Failed_1$ avec la dynamique continue C_5).

La faute f_2 représente l'émergence d'une charge parasite sur la batterie, découlant d'un sous module électrique activé en permanence, par exemple. La charge parasite augmente le courant total et donc le drainage de la batterie (mode $Degraded_1$ avec la dynamique continue C_2), ce qui provoque l'atteinte prématurée de la EOD.

La faute f_3 (resp. f_4 , f_5 et f_6) représente une friction accrue du moteur FL (resp. FR, BL et BR). Une friction accrue induit le besoin d'une quantité de courant plus importante pour

TABLE 5.1 – Commandes continues, mesures continues, et types des fautes du K11.

Type de commande	Commentaires	Unités
Vitesse des roues	Les vitesses commandées sont les mêmes pour les roues du même côté	rad/s
Type de mesure	Commentaires	Unités
Vitesse des roues	Une pour chaque roue	rad/s
Courant total	Un capteur de courant sur le bus d'alimentation	A
Courant des moteurs	Un pour chaque moteur	A
Température des moteurs	Une pour chaque moteur	°C
Température de la batterie	Une pour chaque cellule de batterie	°C
Tension de la batterie	Une pour chaque cellule de batterie	V
Labels d'événements des fautes	Descriptions des fautes	Effets
f_1	Épuisement de la batterie	Mène à la défaillance
f_2	Charge électrique parasite	Augmente le drainage de la batterie
f_3, f_4, f_5, f_6	Frictions accrues des moteurs	Augmentent le drainage de la batterie et la températures des moteurs
f_7, f_8, f_9, f_{10}	Surchauffes des moteurs	Mènent à la défaillance
$f_{11}, f_{12}, f_{13}, f_{14}$	Capteurs de température des moteurs défectueux	Impossibilité d'estimer les températures des moteurs

satisfaire la même vitesse (mode *Degraded₂* avec la dynamique continue *C₃*). En outre, les demandes de charge sont plus importantes ce qui augmentent la température du moteur.

Le pire scénario pour un moteur est une surchauffe. La chaleur finit par détruire l'isolation des roulements, provoquant des courts-circuits et menant à la défaillance du moteur. La surchauffe du moteur FL (resp. FR, BL et BR) est représenté par la faute f_7 (resp. f_8 , f_9 et f_{10}). L'occurrence de n'importe laquelle de ces fautes mène à la défaillance du rover (mode *Failed₂* avec la dynamique continue *C₅*) et représente donc la **fin de vie** (*End Of Life* ou EOL) du rover. On considère un moteur en surchauffe si sa température excède 70 °C.

Le températures des moteurs sont mesurées par quatre capteurs. Néanmoins, ces capteurs sont connus pour tomber en panne de manière inattendue en renvoyant des valeurs incohérentes. Ces défaillances sont représentées par les fautes f_{11} , f_{12} , f_{13} et f_{14} . Nous considérons que le modèle de température n'est pas suffisamment exact sans une correction avec les observations.

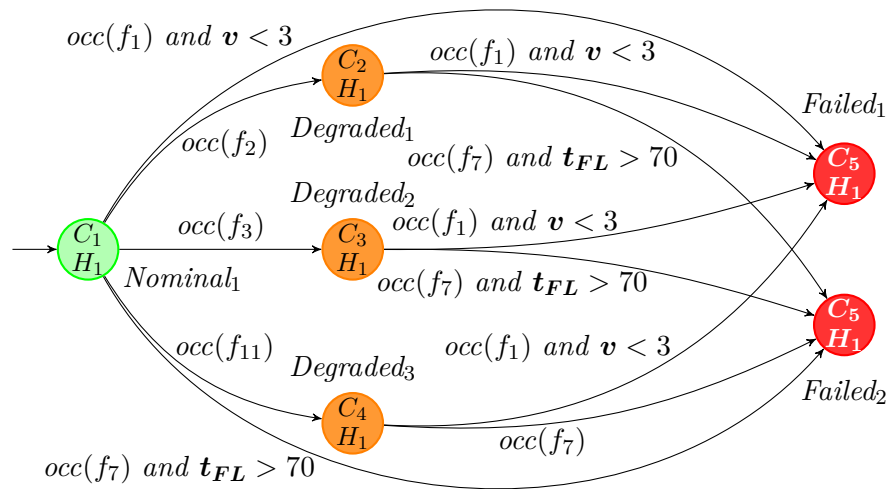


FIGURE 5.2 – Description simplifiée de l'évolution de la santé du rover K11.

En conséquence, une fois que f_{11} (resp. f_{12} , f_{13} et f_{14}) est survenue, l'occurrence de la faute f_7 (resp. f_8 , f_9 et f_{10}) ne correspond à aucune condition sur la température du moteur FL (resp. FR, BL et BR); voir l'arc entre *Degraded₃* et *Failed₂*). Dans la Figure 5.2, le mode *Degraded₃* avec la dynamique continue C_4 , représente le mode dans lequel le capteur de température du moteur FL est défectueux.

Le rover n'a aucun état hybride à surveiller, donc tous les modes ont la même dynamique hybride H_1 , qui correspond à une dynamique stationnaire.

5.3 Modélisation du rover

En considérant les quatre moteurs et les fautes multiples, nous avons identifié 192 modes et 240 changements de mode.

Le modèle HPPN du rover possède 241 places (192 places symboliques, 48 places numériques, 1 place hybride) et 240 transitions.

Le diagnostiqueur HPPN possède le même nombre de places et de transitions que le modèle. L'étape de fusion de la génération du diagnostiqueur ne réduit pas le nombre de transitions car toutes les dynamiques continues sont différentes (voir la Section 3.3). Puisqu'il n'y a qu'une seule place hybride, il n'y a pas de transition dans le niveau hybride.

Le pronostiqueur HPPN a le même nombre de places que le modèle, mais seulement 160 transitions (un gain de 33 %). En effet, lors de l'étape de transformation des conditions de la

génération du pronostiqueur (voir la Section 4.3), seules les transitions étant associées à une contrainte sur l'état continu sont gardées car tous les évènements sont des fautes (et non des commandes discrètes) et qu'aucune contrainte sur l'état hybride n'est associée aux transitions.

Les DES sous-jacent de la description du K11, son modèle HPPN, son diagnostiqueur HPPN et son pronostiqueur HPPN, sont disponibles au lien suivant : <https://homepages.laas.fr/echanthe/K11>.

Le modèle continu nominal est représenté par un ensemble d'équations différentielles qui unifie le modèle de la batterie et ceux des températures [Balaban *et al.* 13]. Il peut être écrit à temps discret en considérant un temps d'échantillonnage de 1/20s, alors que le temps d'échantillonnage des observations est d'environ 1s. On considère 30 variables d'états, incluant la position en 3 dimensions du rover, sa position angulaire relative, les erreurs du contrôle des roues, les températures des moteurs et les températures des roulements des moteurs. Les 24 batteries sont regroupées en une seule dans le but de ne considérer que 3 variables d'état pour la batterie (3 charges, la température et la tension) au lieu de 120. Le modèle de la batterie a été validé avec des données expérimentales dans [Daigle *et al.* 14b, Sweet *et al.* 14]. Cependant, l'unification des modèles de la batterie, du mouvement, et des températures, augmente fortement la complexité et les incertitudes.

L'occurrence et l'effet de la faute f_2 sont modélisés par un paramètre variant dans le temps. La charge parasite est modélisée par l'apparition d'un courant additionnel atteignant une valeur entre 1.5 A et 4.5 A, en quelques secondes après l'occurrence de la faute. Premièrement, deux paramètres sont ajoutés au vecteur d'état continu pour suivre à la fois la durée depuis l'occurrence de la faute et la valeur du courant additionnel. Ensuite, l'augmentation incertaine du courant additionnel est modélisée par l'ajout d'un bruit gaussien, avec une espérance et un écart type commençant respectivement à 3 et 0.3 et décroissant jusqu'à 0 et 0.01 en même temps qu'augmente la durée depuis l'occurrence de la faute.

Enfin, le modèle de la température est assez incertain et donc les mesures des températures sont supposées fiables lorsque les capteurs ne sont pas défectueux. Nous modélisons les fautes f_{11} , f_{12} , f_{13} , f_{14} en augmentant considérablement le bruit associé aux capteurs de température des moteurs car les capteurs défectueux ne renvoient que de grandes valeurs incohérentes. Les fautes f_3 , f_4 , f_5 et f_6 et les frictions accrues des moteurs peuvent être modélisées par des paramètres variant avec le temps (par des résistances additionnelles des moteurs) comme la faute f_2 , mais elles ne font pas partie de cette étude.

5.4 Implémentation

La méthodologie de surveillance basée sur les HPPN (modélisation, diagnostiqueur, pronostiqueur, etc.) est implémentée en Python 3.4. Les tests ont été effectués sur un processeur

Intel(R) 4 Core(TM) i5 – 4590 à 3.30 GHz avec 16 GB de RAM, tournant sous GNU/Linux (Linux 3.13, x86_64). Dans le but de réduire le temps de calcul, les mises à jours des valeurs des jetons sont réparties sur les 4 cœurs physiques. Le reste de l'implémentation utilise un seul cœur.

5.5 Résultats

Trois scénarios étudiés dans [Sweet et al. 14] sont considérés dans ce chapitre. Dans un premier temps, on s'intéresse aux résultats de la méthodologie avec des jeux de paramètres d'échelle fixes pour le diagnostiqueur et le pronostiqueur. Dans un second temps, la performance de la méthodologie est étudiée en comparant les résultats, les temps de calcul et l'utilisation de la mémoire pour différents jeux de paramètres d'échelle.

La mission du rover est de visiter un maximum de 12 points d'intérêts (montrés dans la Figure 5.3(a)) et de retourner à sa station de charge. Des récompenses sont associées à chacun des points d'intérêts. Dans des conditions nominales, le module de décision (DM) du rover retourne un chemin de 5 points d'intérêts, commençant et terminant à la même position.

Pour chaque scénario, le K11 commence sa mission à 0s avec une batterie chargée complètement et avec tous ses composants à la température ambiante.

Les trois scénarios sont les suivants :

- un scénario nominal durant lequel aucune faute ne survient (Figure 5.3(b)),
- un scénario durant lequel une charge parasite apparaît sur la batterie et cette dernière se décharge complètement avant la fin de la mission car le DM est désactivé (Figure 5.3(c)).
- un scénario durant lequel une charge parasite apparaît sur la batterie (i.e. la faute f_2 survient) et le DM replanifie la mission pour que le rover revienne à sa station de charge avant la EOD (Figure 5.3(d)),

Deux capteurs de température (FL et BL) sont actuellement défectueux sur le rover. Ces fautes ont un impact sur la surveillance mais pas sur le système physique, donc le DM retourne le même chemin que dans des conditions nominales. Ces défaillances des capteurs sont diagnostiquées en une période d'échantillonnage par le diagnostiqueur si le mode initial est supposé inconnu. Pour la suite, on considère que le mode initial est un mode dégradé.

Par souci de clarté, les modes sont dorénavant désignés avec des mots-clés représentatifs de l'état physique du rover. Par exemple, le mode initial est désigné comme le mode *Sensor BL FL fault*. On considère aussi la fin de décharge (EOD) comme une fin de vie (EOL) du rover.

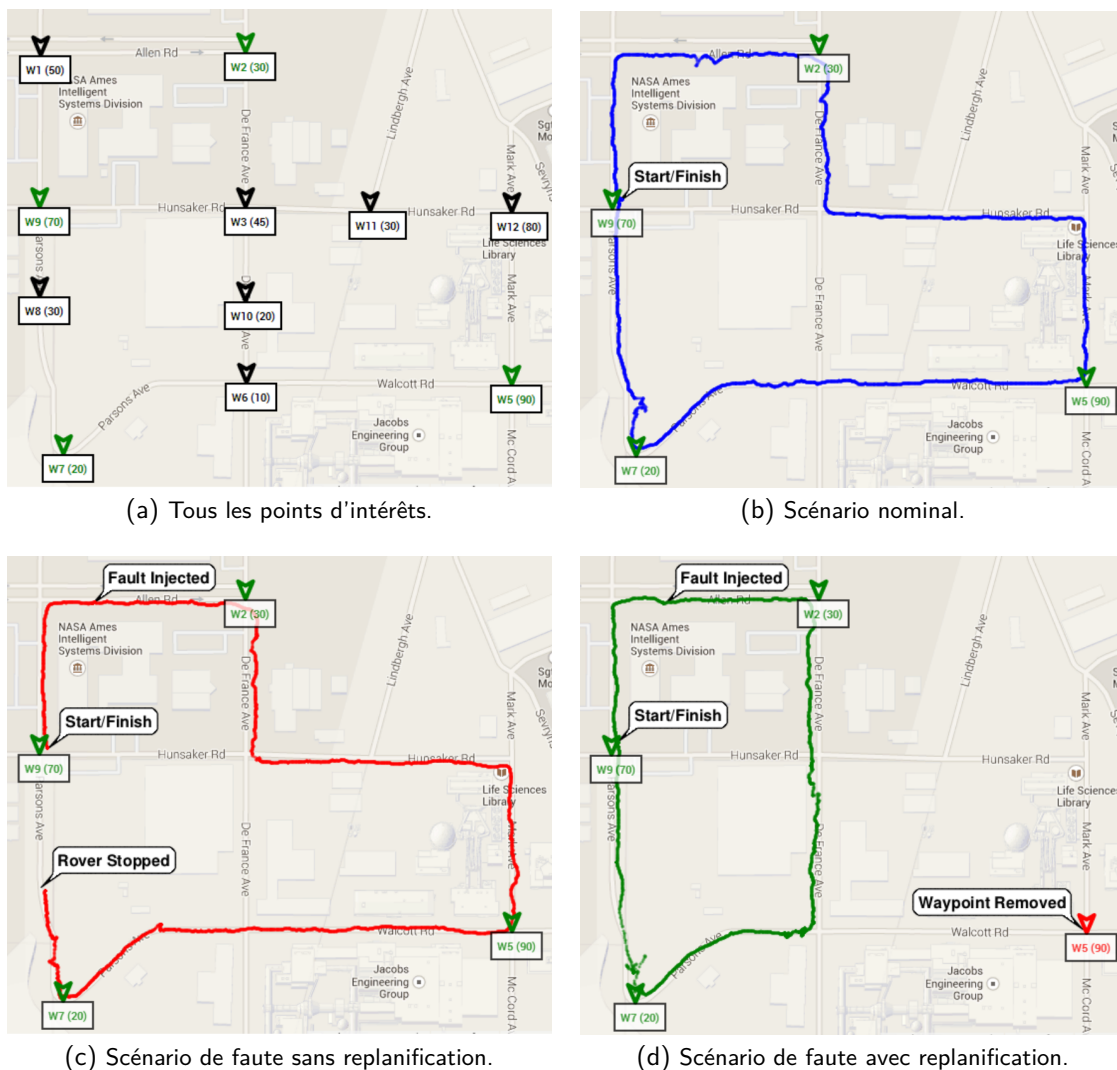


FIGURE 5.3 – Routes empruntées par le rover dans le scénario nominal et avec l'apparition d'une charge parasite sur la batterie, avec et sans replanification [Sweet et al. 14].

Rappel

Le diagnostiqueur utilise trois paramètres d'échelle ρ_{Δ}^{min} , ρ_{Δ}^{max} et ρ_{Δ}^{tot} . Les paramètres ρ_{Δ}^{min} et ρ_{Δ}^{max} sont représentatifs respectivement des précisions minimale et suffisante accordées au diagnostiqueur pour surveiller une hypothèse. Le paramètre ρ_{Δ}^{tot} est quant à lui représentatif du nombre maximum de jetons à utiliser pour surveiller toutes les hypothèses réunies.

Le score d'une hypothèse est calculé avec l'Équation 3.34, qui dépend d'un paramètre α représentatif de la confiance relative entre la partie discrète et la partie continue du modèle.

Le pronostiqueur utilise trois paramètres d'échelle ρ_{II}^{min} , ρ_{II}^{max} et ρ_{II}^{tot} . Les paramètres ρ_{II}^{min}

et ρ_{Π}^{max} sont respectivement les précisions minimale et suffisante accordées au pronostiqueur pour simuler l'évolution d'une hypothèse dans le futur. Le paramètre ρ_{Π}^{tot} est quant à lui représentatif du nombre maximum de jetons à utiliser pour simuler l'évolution de toutes les hypothèses réunies.

Le processus de pronostic simule l'évolution des hypothèses pendant un horizon de prédiction τ_p .

Concernant les paramètres d'échelle du diagnostiqueur, on considère que les équations d'évolutions continues sont assez précises mais on souhaite que le diagnostiqueur puisse surveiller de nombreuses hypothèses en même temps. En considérant les 32 variables d'état continues, nous fixons dans un premier temps les paramètres d'échelle du diagnostiqueur à :

$$(\rho_{\Delta}^{min}, \rho_{\Delta}^{max}, \rho_{\Delta}^{tot}) = (40, 80, 1500)_{\Delta},$$

pour pouvoir surveiller entre 18 et 37 clusters de particules avec des précisions variants entre 40 et 80. La précision de l'hypothèse initiale est $n_0 = 80$.

On accorde une confiance égale entre les parties discrète et continue du modèle : $\alpha = 0.5$. Sachant qu'il n'y a pas d'observation discrète, cela signifie que le calcul des scores des hypothèses favorise les hypothèses qui considère le moins de fautes.

Pour pouvoir simuler uniquement l'évolution dans le futur des hypothèses les plus pertinentes, mais aussi limiter le temps de calcul du pronostic, on fixe dans un premier temps les paramètres d'échelle du pronostiqueur à :

$$(\rho_{\Pi}^{min}, \rho_{\Pi}^{max}, \rho_{\Pi}^{tot}) = (1, 5, 100)_{\Pi}.$$

Ainsi, les évolutions de 20 à 100 hypothèses seront simulées avec des précisions variants entre 1 et 5.

L'horizon de prédiction est fixé à $\tau_p = 10000s$ car les missions durent au maximum 150 minutes (temps approximatif de la décharge complète de la batterie en fonctionnement nominal). Dans les trois scénarios, les futures commandes se basent sur les données enregistrées, puis la dernière commande est répétée jusqu'à ce que l'horizon de prédiction soit atteint. Le pronostiqueur calcule un pronostic toutes les 30 secondes.

5.5.1 Scénario nominal

Dans le scénario nominal, aucune faute ne survient et le rover exécute avec succès sa mission. Il rejoint sa station de charge à 5064s.

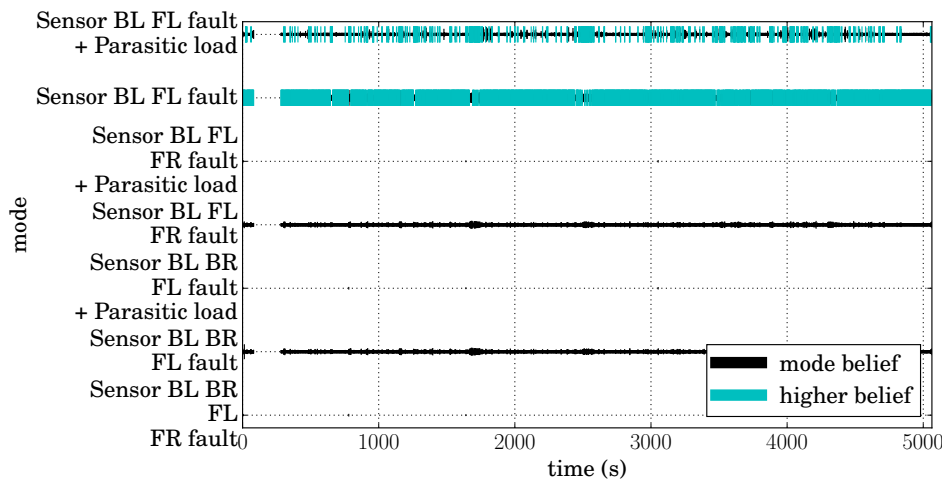


FIGURE 5.4 – Distributions de croyances sur le mode du système à chaque instant avec pour le scénario nominal - $(40, 80, 1500)_{\Delta}$.

La Figure 5.4 montre les distributions de croyances sur le mode du système à chaque instant. Le degré de croyance d'un mode à chaque instant k est représenté par l'épaisseur du trait et le degré de croyance le plus élevé de tous les modes est affiché en bleu. Le degré de croyance d'un mode à l'instant k est le maximum des degrés de croyance de toutes les hypothèses du diagnostiqueur étant dans ce mode à l'instant k . Le degré de croyance d'une hypothèse est son score, calculé avec l'Équation 3.34.

Dans la Figure 5.4, le vide entre 81s et 281s correspond à une pause durant l'expérimentation. La figure montre que le diagnostiqueur accorde quasiment tout le temps le degré de croyance le plus élevé au mode réel *Sensor BL FL fault*.

D'autres modes possèdent également un degré de croyance élevé à cause des incertitudes liées au modèle. Par exemple, la Figure 5.5 montre la tension mesurée aux bornes de la batterie comparée aux estimations des deux hypothèses ayant les degrés de croyance les plus élevés à 650s. L'hypothèse 1 considère qu'aucune faute n'est survenue sur le système avec un degré de croyance de 0.1, alors que l'hypothèse 2 considère qu'une charge parasite est apparue sur la batterie à 641s avec un degré de croyance de 0.278. À 650s, la charge parasite est estimée à 2.56 A et l'on peut voir dans la Figure 5.5 que l'hypothèse 2 est plus proche des données que l'hypothèse 1. Dans la suite du scénario, le diagnostiqueur élimine néanmoins l'hypothèse 2 à partir de 665s car une telle charge n'est pas cohérente avec les données suivantes. À fin du scénario, l'hypothèse avec le degré de croyance le plus élevé est bien celle qui considère qu'aucune faute n'est survenue.

Concernant les résultats de pronostic, on s'attend à ce que les estimations du EOL ayant les degrés de croyance les plus élevés soient ultérieures à 5064s, date à laquelle la mission s'est

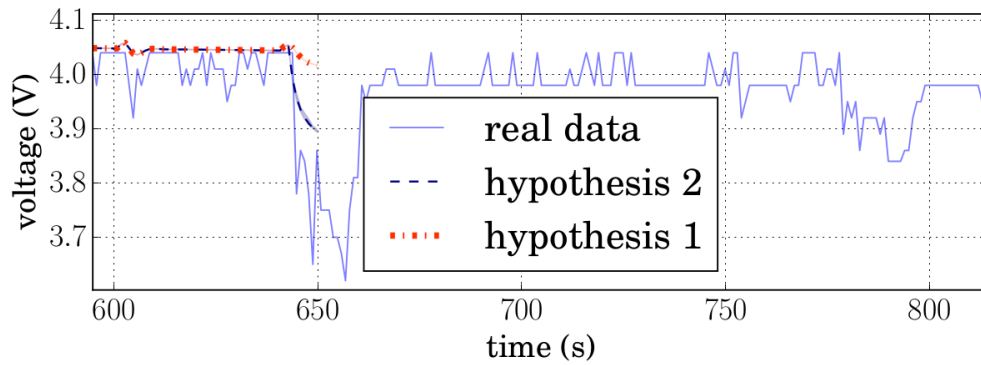


FIGURE 5.5 – Tension mesurée aux bornes de la batterie du rover et les estimations de deux hypothèses du diagnostiqueur à 650s du scénario nominal.

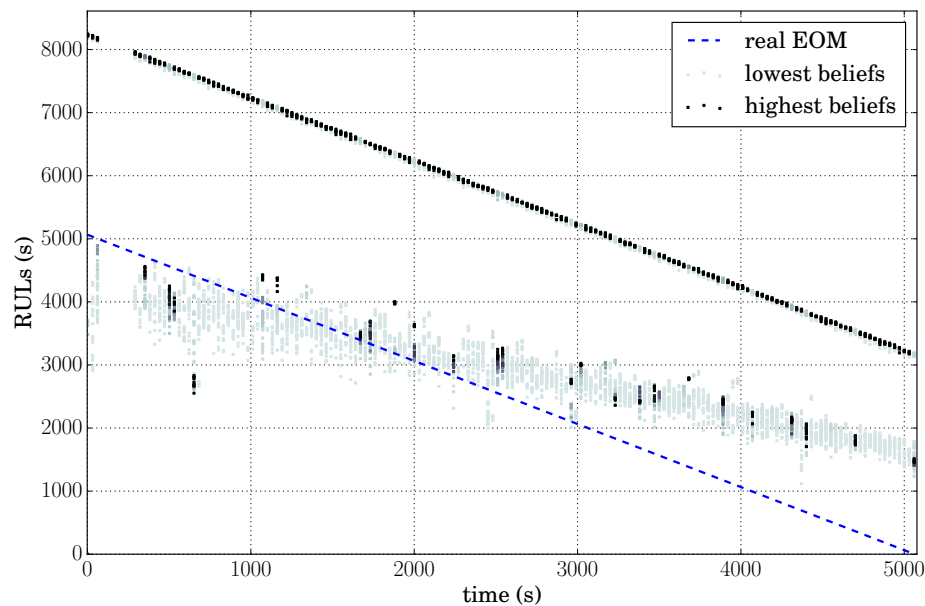


FIGURE 5.6 – Estimations du RUL du système pour le scénario nominal - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.

terminée. La Figure 5.6 montre les estimations du RUL du système tout au long du scénario. Le RUL des hypothèses sont dessinés avec des nuances de gris ; celles avec les degrés de croyance les plus élevés sont noires, tandis que celles avec les degrés de croyance faibles sont en gris clair. La fin de mission réelle (EOM pour *End Of Mission*) est représentée par une ligne discontinue.

La Figure 5.6 montre les estimations du RUL en comparaison au temps qu'il reste réellement avant la fin de la mission. On peut voir sur la figure que les EOL ayant les degrés de croyance les plus importants sont estimés à plus de 8000s quasiment à chaque instant. Ces résultats sont cohérents avec nos attentes car il y a une différence constante d'environ 3000 secondes entre le EOM et ces EOL. La figure montre également la présence de RUL plus courts et qui

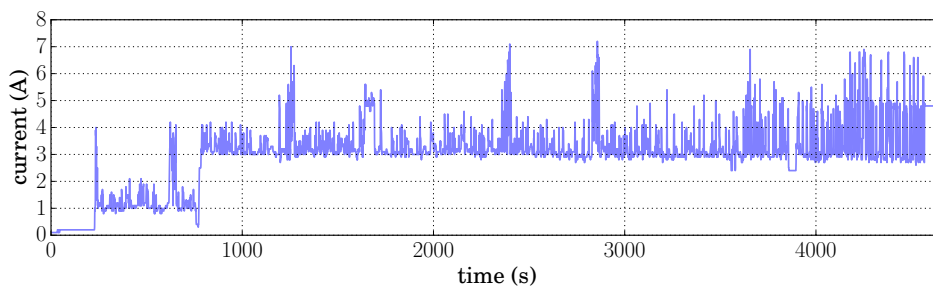


FIGURE 5.7 – Intensité du courant mesurée dans le bus d'alimentation du rover durant le scénario de faute sans replanification.

ont majoritairement des degrés de croyance plus faibles. Ces RUL sont ceux des hypothèses du diagnostiqueur qui considèrent qu'une charge parasite est survenue sur la batterie.

5.5.2 Scénario de faute sans replanification

Dans ce scénario, une charge parasite de 2A apparaît sur la batterie, ce qui correspond à l'occurrence de la faute f_2 . La batterie du rover se décharge complètement avant le retour à la station de charge.

On peut voir l'apparition de la charge parasite dans la Figure 5.7, qui montre l'intensité du courant mesurée dans le bus d'alimentation du rover à chaque instant de la mission. Entre 0s et 228s, les commandes sont nulles donc le rover n'est pas en déplacement et le courant total est faible. À partir de 228s, le rover commence à se déplacer et le courant total vaut en moyenne 1A, ce qui correspond à une consommation en électricité des quatre moteurs en action. Entre 772s et 786s, on peut observer le courant augmenter brutalement à environ 3A avec l'apparition de la charge parasite. Dans la Figure 5.8 qui montre la tension mesurée aux bornes de la batterie à chaque instant de la mission, on peut également repérer l'apparition de la charge parasite entre 772s et 786s et l'arrêt du rover à cause d'une tension trop faible à 4571s.

La Figure 5.9 expose les distributions de croyances sur le mode du système à chaque instant. À partir de 774s, le degré de croyance du mode *Sensor BL FL fault + Parasitic load* est le plus élevé jusqu'à 4571s, lorsque le K11 atteint la décharge complète (EOL) et s'arrête. Après 4571s, seul le mode *Sensor BL FL fault + Parasitic load + Discharged* a le degré de croyance le plus élevé mais le diagnostiqueur considère toujours l'hypothèse selon laquelle le rover n'a pas atteint le EOL.

La Figure 5.10 montre les estimations du RUL du système tout au long du scénario. La fin de vie réelle (EOL) est représentée par une ligne discontinue. Avant l'occurrence de la faute, on peut reconnaître le même schéma que dans le scénario nominal. Après l'occurrence de la

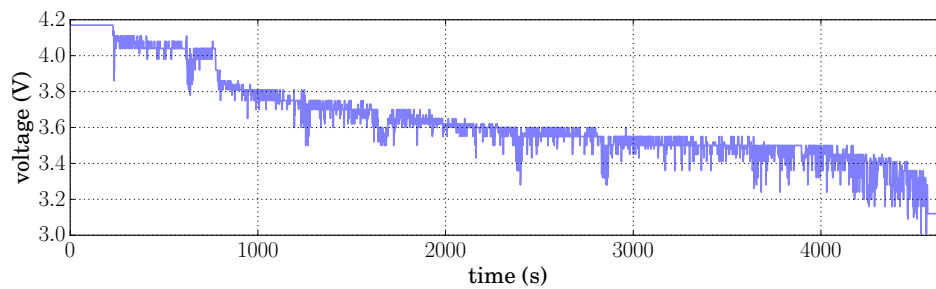


FIGURE 5.8 – Tension mesurée aux bornes de la batterie du rover durant le scénario de faute sans replanification.

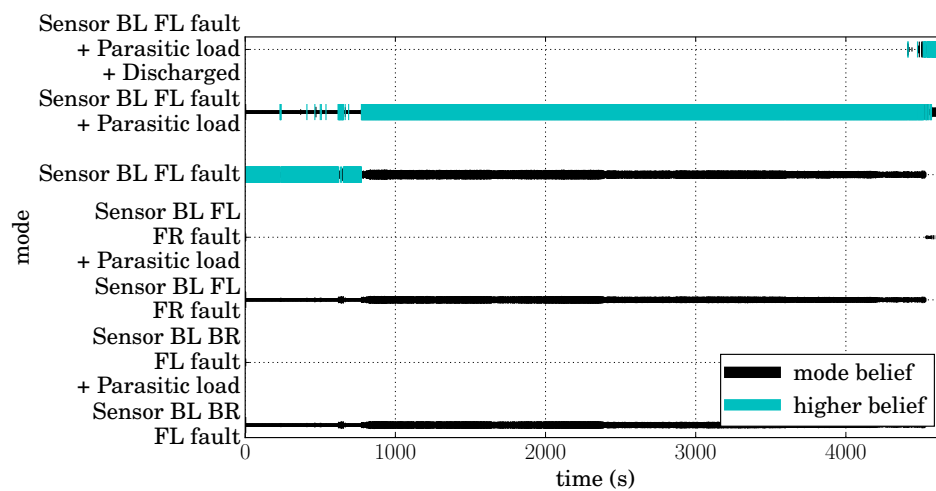


FIGURE 5.9 – Distributions de croyances sur le mode du système à chaque instant avec pour le scénario de faute sans replanification - $(40, 80, 1500)_{\Delta}$.

faute, les estimations des EOL des hypothèses ayant les degrés de croyance les plus élevés sont antérieures au véritable EOL, et, après 3000s, elles sont très proches de lui.

Entre 750s et 2800s, on distingue deux groupes de RUL ; celui de l'hypothèse initiale (l'hypothèse qui considère que le système est toujours dans le mode initial) et celui des hypothèses qui considèrent que la faute f_2 est survenue. Le second groupe fournit des résultats acceptables, mais il n'encadre pas le vrai EOL de façon permanente. Cela est dû à une estimation inexacte de la charge parasite qui est liée au modèle de faute de la batterie. Deux modèles de décharge de la batterie en présence d'une charge parasite (2A et 2.5A) sont comparés à la tension mesurée aux bornes de la batterie dans la Figure 5.11. On peut voir que le modèle avec une charge de 2.5A correspond mieux aux données réelles entre 800s et 2000s, alors que le modèle avec une charge de 2A correspond mieux aux données à partir de 2500s. Le diagnostiqueur estime dans un premier temps la charge parasite à environ 2.5A, et donc le pronostiqueur prédit une entrée dans un mode de défaillance plus tôt que ce qui se passe réellement. Après 1700s, la

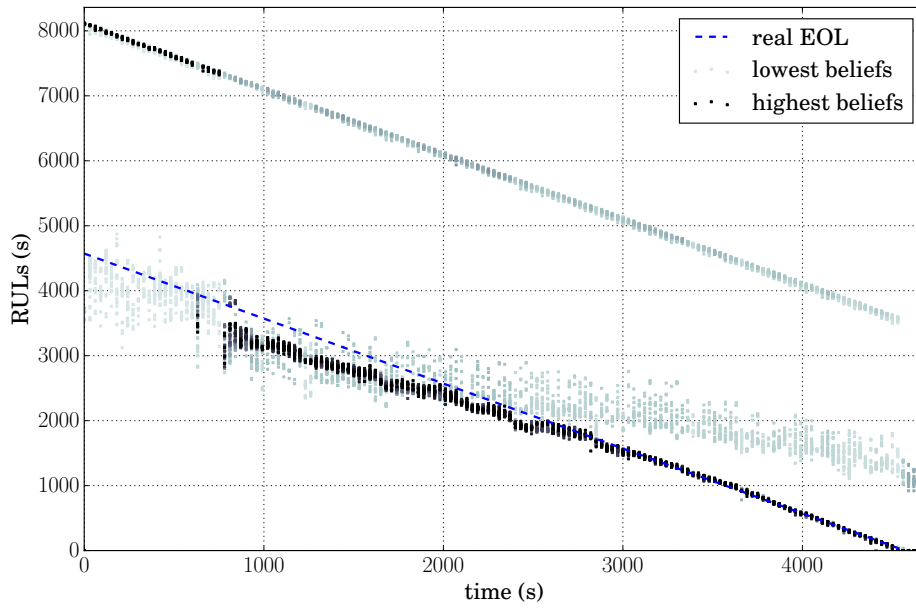


FIGURE 5.10 – Estimations du RUL du système pour le scénario de faute sans replanification - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.

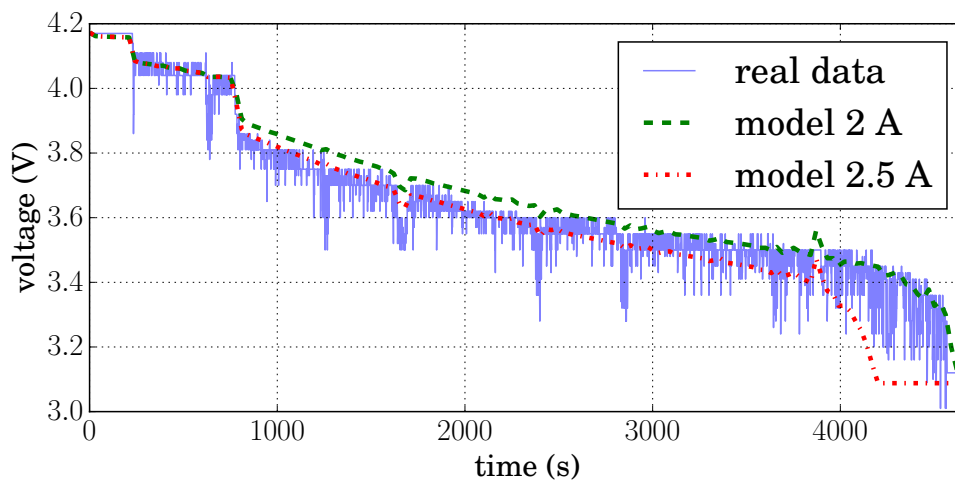


FIGURE 5.11 – Comparaison de la tension mesurée aux bornes de la batterie durant le scénario de faute sans replanification et de simulations de modèles avec des charges parasite de 2A et 2.5A.

charge parasite est estimée à environ 2A, et le pronostiqueur retourne des RUL qui sont plus cohérents avec la réalité.

Dans la seconde partie du scénario, un troisième groupe apparaît dans les estimations du RUL de la Figure 5.10. Il correspond aux hypothèses découlant de l'hypothèse initiale et qui considèrent que la charge parasite est apparue récemment. Les degrés de croyance de ces hypothèses sont cependant plus faibles que les autres.

Dans le cas du rover qui utilise le module DM pour replanifier sa mission si cette dernière n'est plus réalisable, il est intéressant de pouvoir avoir une estimation de ses trajectoires passée et future. La Figure 5.12(a) montre la distribution sur la trajectoire du système à 781s. Elle combine les hypothèses du diagnostic avec celles du pronostic.

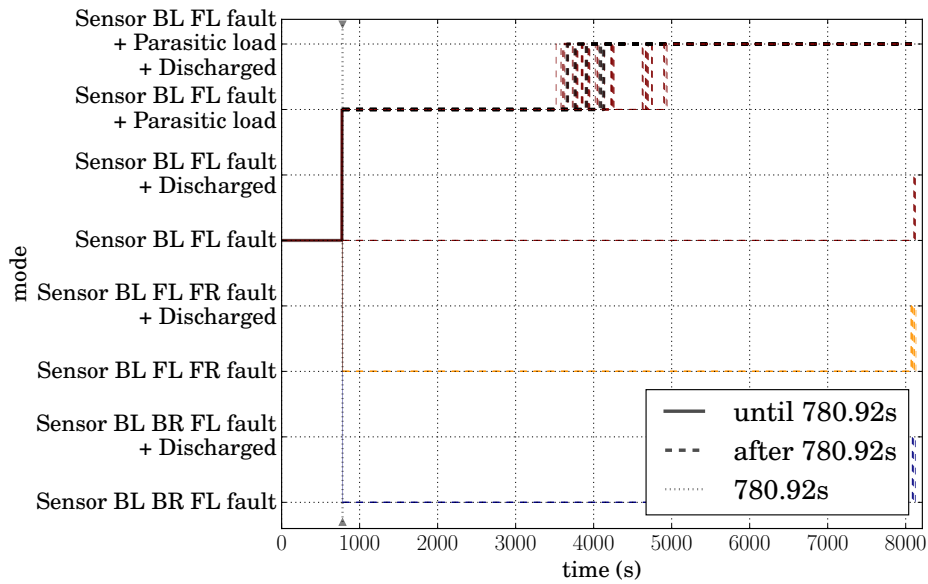
Concernant le futur, la figure présente clairement les deux trajectoires possibles du système : soit il se déchargera entre 3500s et 5000s, soit il se décharge vers 8050s. Concernant le passé, elle montre que l'occurrence de la faute f_2 est estimée entre 760s et 781s, et plus vraisemblablement entre 771s et 776s (voir le zoom de la Figure 5.12(b)), ce qui est en accord avec notre analyse des données montrées dans les Figures 5.7 et 5.8.

5.5.3 Scénario de faute avec replanification

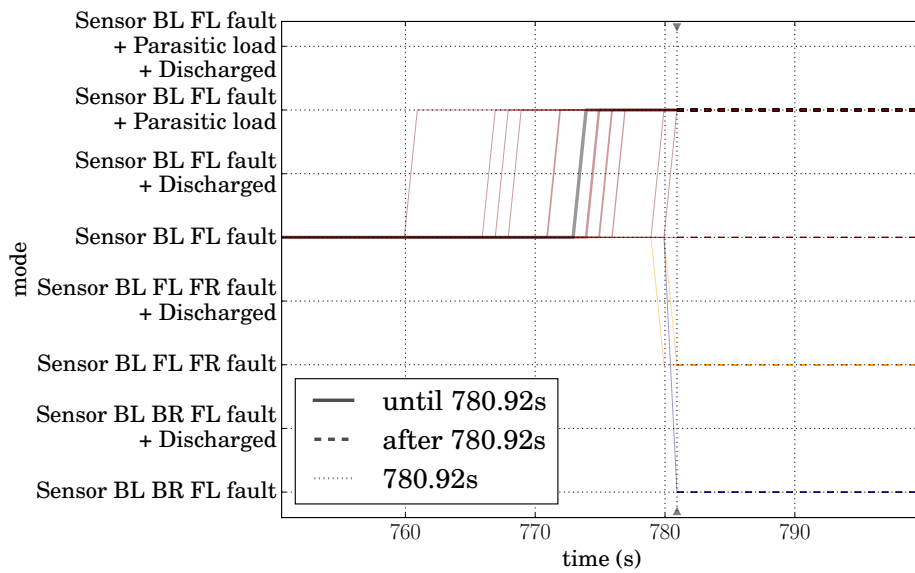
Dans ce scénario, la charge parasite de 2A apparaît sur la batterie entre 677s et 681s mais le DM annule la visite du dernier point d'intérêt pour que le rover revienne à sa station de charge à 3874s, avant la décharge complète de sa batterie.

La Figure 5.13 montre que la faute f_2 est immédiatement détectée par le diagnostiqueur. À partir de 679s, le degré de croyance du mode *Sensor BL FL fault + Parasitic load* est le plus élevé jusqu'à la fin de la mission.

En ce qui concerne les résultats de pronostic, on s'attend à ce que les estimations du EOL ayant les degrés de croyance les plus élevés soient ultérieures à 3874s (le EOM réel), excepté pendant la courte période durant laquelle le DM interne du rover détecte, isole et replanifie la mission. La Figure 5.14 montre les estimations du RUL du système tout au long du scénario. Dans cette figure, on retrouve les mêmes phénomènes déjà discutés pour le scénario de faute sans replanification. De plus, les résultats sont cohérents avec nos attentes car à partir de 750s, quasiment toutes les estimations des RUL sont plus longues que la durée restante jusqu'à la fin réelle de la mission.



(a) Vue d'ensemble.



(b) Zoom entre 750s et 800s.

FIGURE 5.12 – Distribution de croyances sur les trajectoires passée et future du système à 781s pour le scénario avec faute sans replanification - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.

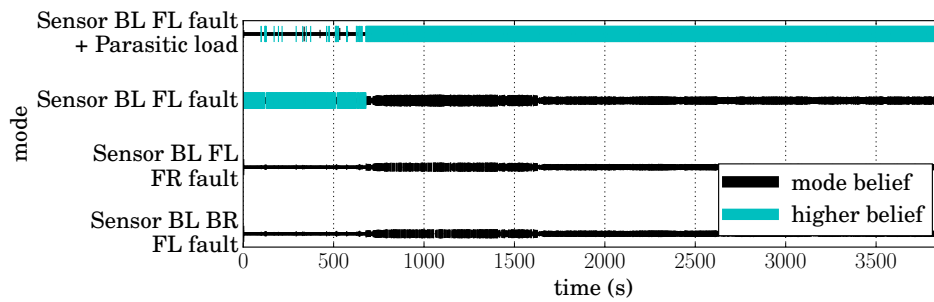


FIGURE 5.13 – Distributions de croyances sur le mode du système à chaque instant avec pour le scénario de faute avec replanification - $(40, 80, 1500)_{\Delta}$.

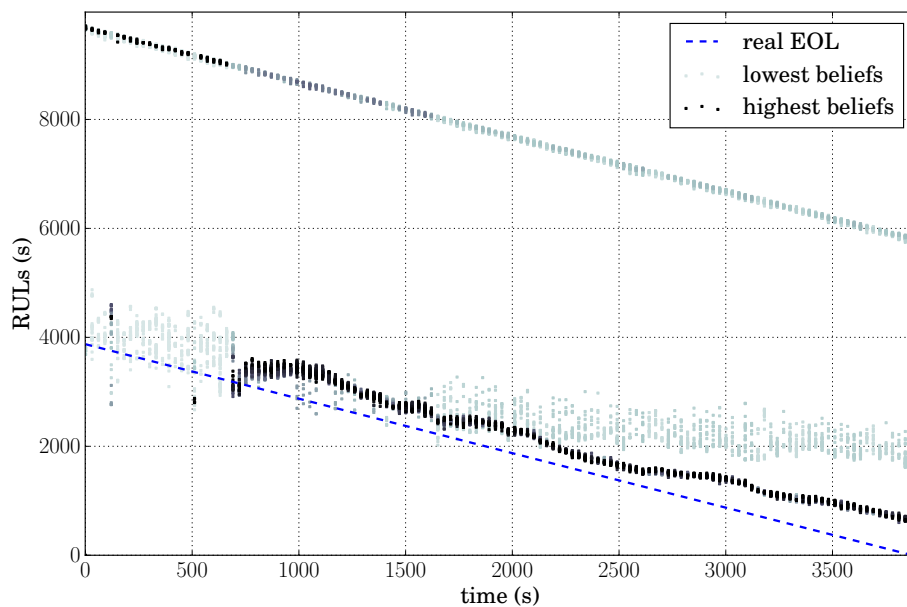


FIGURE 5.14 – Estimations du RUL du système pour le scénario de faute avec replanification - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.

5.5.4 Analyse des performances

Les fautes sont toujours détectées en une période d'échantillonnage car le diagnostiqueur HPPN considère toutes les possibilités durant l'étape de prédiction et garde le marquage cohérent durant l'étape de prédiction.

Les temps de calcul d'un diagnostic et d'un pronostic, et la mémoire RAM maximum utilisée pour différents jeux de paramètres d'échelle, sont donnés dans la Table 5.2.

Ces métriques soulignent le fait que les temps de calcul avec le jeu de paramètres d'échelle initial restent acceptables mais ne respectent pas les contraintes de temps réel. En effet, de

TABLE 5.2 – Performances calculatoires de la surveillance de santé basée sur les HPPN du rover K11 pour différents jeux de paramètres d'échelle.

Paramètres d'échelle		Temps Δ_k (s)	Temps Π_k (s)	RAM max. (MB)
(40, 80, 1500) $_{\Delta}$ et (1, 5, 100) $_{\Pi}$	min.	0.28	52.12	
	max.	4.54	1188.48	
	moy.	3.35	605.32	126.73
(40, 80, 400) $_{\Delta}$ et (1, 5, 100) $_{\Pi}$	min.	0.28	0.03	
	max.	1.00	444.73	
	moy.	0.56	122.15	122.15
(20, 60, 400) $_{\Delta}$ et (1, 1, 1) $_{\Pi}$	min.	0.22	0.03	
	max.	0.98	33.50	
	moy.	0.74	12.35	112.18

TABLE 5.3 – Performances calculatoires de la surveillance de santé basée sur les HPPN du système des réservoirs d'eau.

Paramètres d'échelle		Temps Δ_k (s)	Temps Π_k (s)	RAM max. (MB)
(40, 80, 1500) $_{\Delta}$ et (1, 5, 100) $_{\Pi}$	min.	0.12	0.08	
	max.	3.96	201.73	
	moy.	1.22	44.29	102.56

nouvelles observations arrivent approximativement toutes les secondes, mais le temps moyen du calcul du diagnostic est 4.54s. De plus, on lance un processus de pronostic toutes les 30s, mais le temps du calcul du pronostic minimum est 52.12s.

Ces performances s'expliquent principalement par le fait que les méthodes de diagnostic et de pronostic se basent sur la simulation pas à pas de l'évolution des nombreux jetons des HPPN. Bien que les générations et les processus du diagnostiqueur et du pronostiqueur HPPN aient été pensés pour optimiser ces simulations, ce type de méthodes dites séquentielles restent par principe coûteuses en terme de calcul.

Une comparaison avec les performances calculatoires de la méthode de surveillance sur le système des réservoirs d'eau avec les mêmes paramètres d'échelle (voir la Table 5.3) permet de déduire que les performances dépendent également du modèle HPPN du système. La complexité du modèle HPPN est par ailleurs difficile à évaluer, car les nombres de places et de transitions, la structure du réseau, les complexités calculatoires des dynamiques continues et hybrides, le nombres de variables d'états, sont des paramètres qui influent sur les performances générales de la méthode.

S'il est difficile de déterminer la complexité d'un modèle HPPN et d'évaluer son impact sur les performances calculatoires de la méthode de surveillance, on propose dans ces travaux de contrôler ces performances à l'aide des paramètres d'échelle.

On peut en effet voir dans la Table 5.2 que réduire les paramètres d'échelle diminue les temps de calcul et la mémoire RAM maximum utilisée. En choisissant $\rho_{\Delta}^{tot} = 400$, le temps maximum obtenu pour le calcul du diagnostic est 0.98s et est inférieur à 1s. La qualité de la surveillance en est néanmoins impactée car le diagnostiqueur n'a à sa disposition qu'entre 5 et 10 hypothèses. Une conséquence directe que l'on peut voir sur les Figures 5.15 (a), (c) et (e), et sur les Figures 5.16 (a), (c) et (e), est que l'apparition d'une charge parasite est considérée comme sûre assez rapidement à cause des données entachées d'imprécisions du rover. Toutefois, le diagnostiqueur estime que la valeur de la charge parasite est faible, ce qui a pour conséquence d'obtenir des résultats de pronostic satisfaisants comme le montrent les Figures 5.15 (b), (d) et (f).

Réduire les paramètres d'échelle du pronostiqueur à $(\rho_{\Pi}^{min}, \rho_{\Pi}^{max}, \rho_{\Pi}^{tot}) = (1, 1, 1)_{\Pi}$ permet d'obtenir des temps de calcul du pronostic bornés à 33.50s. Les Figures 5.16 (b), (d) et (f) exposent les résultats de pronostic avec ces nouveaux paramètres d'échelle sur les trois scénarios. Comme on peut s'y attendre, les résultats comprennent beaucoup moins d'incertitudes mais la seule hypothèse du pronostiqueur permet d'obtenir un RUL du système exploitable par des techniques de gestion de santé comme la replanification de la mission ou dans un cadre de maintenance.

5.6 Conclusion

Ce chapitre propose une expérimentation de la méthodologie de surveillance de santé sur le rover planétaire K11, un banc d'essai développé par le NASA Ames Research Center à des fins de diagnostic et de pronostic.

Cette application met en avant l'applicabilité de la méthode sur des systèmes réels et sa robustesse face à des données entachées d'incertitudes. Le K11 n'a en effet ni commande discrète, ni sortie de capteur discrète et aucune connaissance sur les dates d'occurrences des fautes n'est disponible. En outre, la correspondance entre les données réelles et le modèle continu est assez incertaine.

L'expérimentation met en évidence les limites de notre approche vis-à-vis particulièrement des performances des fonctions de diagnostic et de pronostic qui sont basées sur la simulation pas à pas de l'évolution du système. Le compromis, instauré par l'algorithme SSA (Section 3.6), entre la précision des résultats et les performances calculatoires de la méthode est finalement proposé comme une solution au problème et son utilité est illustrée sur les différents scénarios du rover K11.

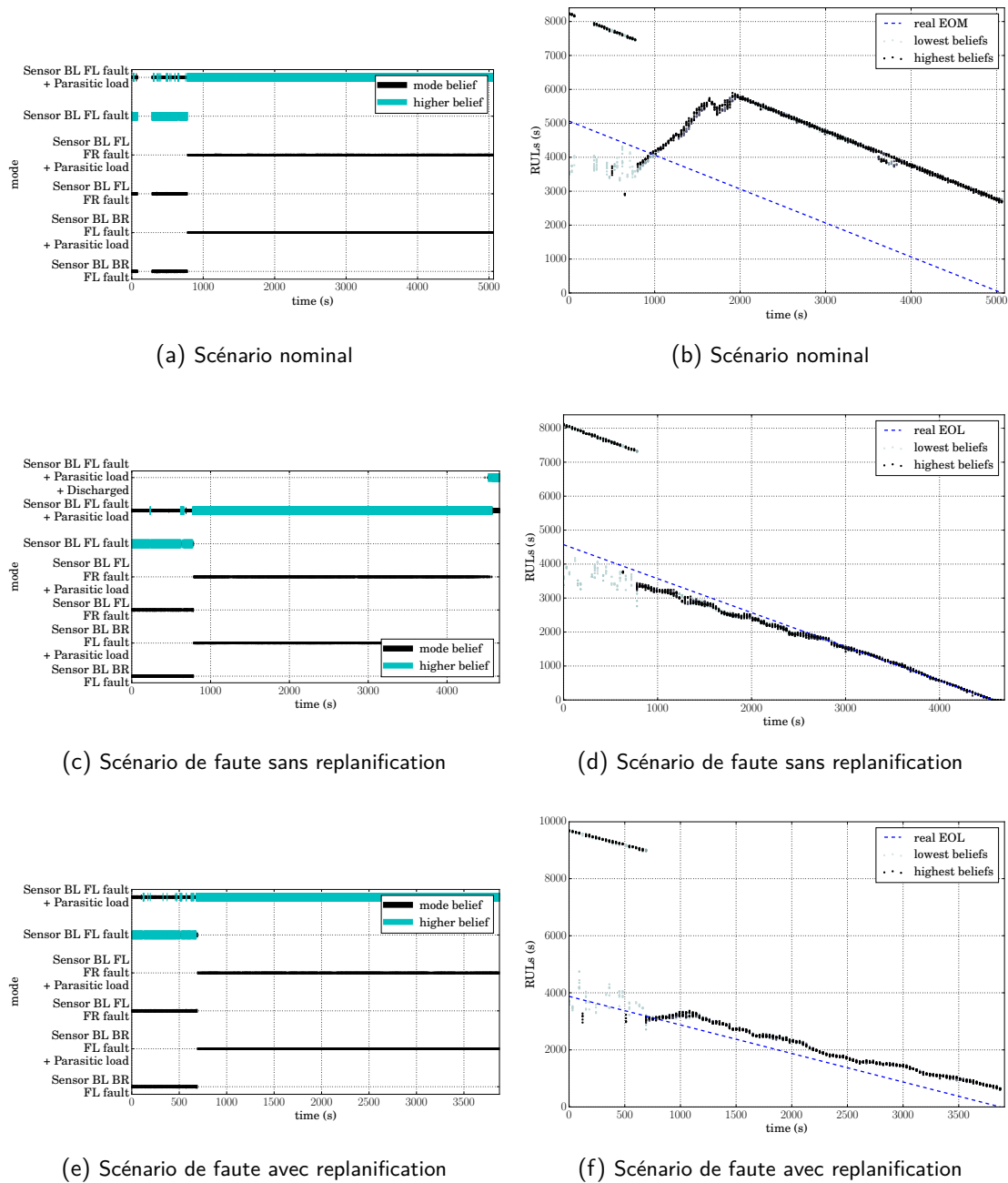


FIGURE 5.15 – Degrés de croyance du mode courant et estimations du RUL - $(40, 80, 400)_\Delta$ et $(1, 5, 100)_\Pi$.

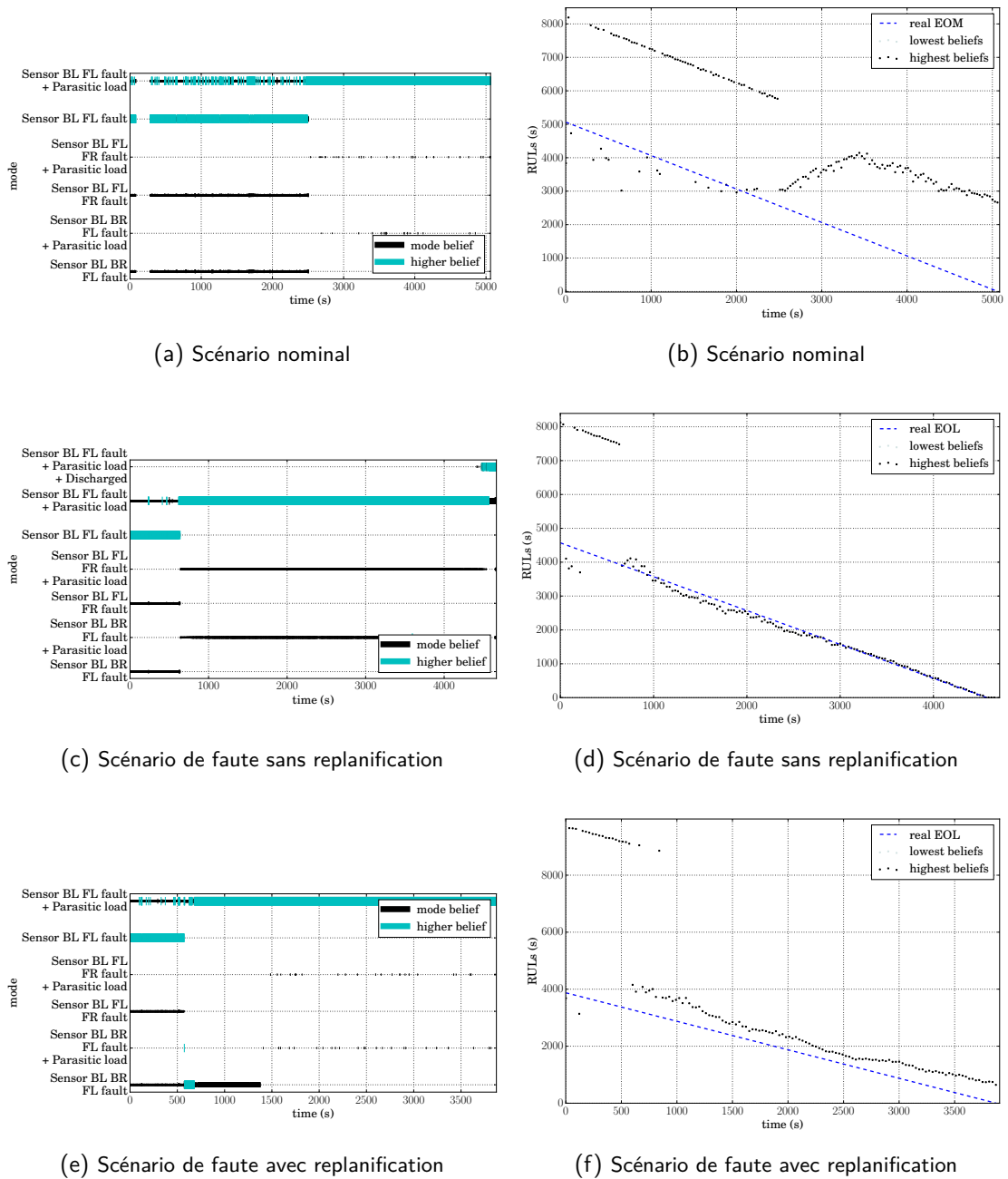


FIGURE 5.16 – Degrés de croyance du mode courant et estimations du RUL - $(20, 60, 400)_{\Delta}$ et $(1, 1, 1)_{\Pi}$.

L'application de la méthodologie de diagnostic basée sur les HPPN a été le sujet des travaux de [Gaudel *et al.* 16].

Conclusions et Perspectives

Cette étude s'inscrit dans le domaine de la gestion de santé des systèmes, qui vise à développer des outils d'aide à la maintenance, mais également à améliorer les prises de décision en autonomie des systèmes en cas de pannes. Dans ce cadre, des techniques de diagnostic déterminent si et pourquoi le système est en panne, alors que des techniques de pronostic déterminent quand les pannes vont survenir et leurs conséquences. Si ces techniques semblent être corrélées, elles sont généralement étudiées séparément, car les échelles de temps manipulées par les deux processus sont très différentes.

L'objectif de ces travaux est d'étudier l'intérêt d'une approche intégrée des méthodes de diagnostic et de pronostic pour la surveillance des systèmes hybrides sous incertitude.

Cette problématique soulève plusieurs verrous, dont la nécessité de définir une architecture intégrant diagnostic et pronostic dans le cadre du PHM. La première partie du Chapitre 1 se concentre sur l'étude des architectures de systèmes existantes et montre qu'il devient indispensable de considérer les fonctions de diagnostic et de pronostic dès la conception du système. Ensuite, l'étude des formalismes utilisés pour la surveillance des systèmes montre que les systèmes hybrides semblent les plus appropriés pour la modélisation de l'évolution de la santé des systèmes complexes, de part leur combinaison explicite de dynamiques discrètes et continues et par la spécification des incertitudes liées aux modèles et aux observations. De nombreux travaux ont été réalisés sur le diagnostic de systèmes hybrides, mais peu sur le pronostic de système hybride. La gestion des incertitudes au niveau du diagnostic est parfois le sujet de travaux mais peu d'entre eux concernent le couplage interactif du diagnostic et du pronostic pour les systèmes hybrides.

Le premier apport de cette thèse est la proposition dans le Chapitre 2 d'un cadre de modélisation utilisant le formalisme des réseaux de Petri hybrides particuliers (HPPN) qui enrichit les réseaux de Petri avec des dynamiques continues et hybrides. En aval du développement de ce formalisme, l'idée est de proposer une méthodologie de modélisation des systèmes hybrides. Les HPPN ont plusieurs avantages :

- ils bénéficient de la capacité d'expression et de la compacité des réseaux de Petri,

- ils permettent d'exprimer à la fois les contraintes des systèmes à événements discrets et des systèmes continus,
- ils permettent de spécifier les incertitudes sur la connaissance du système et sur les observations,
- ils permettent d'exprimer des caractéristiques purement hybrides.

La méthodologie de modélisation se base sur une description de l'évolution de la santé avec un **système multimode**, pour lequel à chaque mode sont associées une dynamique continue représentant le comportement du système et une dynamique hybride décrivant l'évolution de sa dégradation.

Le Chapitre 3 propose une méthode de diagnostic basée sur la génération d'un diagnostiqueur HPPN à partir du modèle HPPN. Le diagnostiqueur HPPN surveille sous incertitude à la fois le comportement du système et sa dégradation. Le processus de diagnostic utilise à chaque instant les observations sur le système pour faire évoluer son marquage (ses jetons) sur le principe de prédiction et de correction des **filtres particuliers**, combiné à un **algorithme stochastique de changement d'échelle** (SSA). Le diagnostic courant du système est le marquage courant du diagnostiqueur, qui porte une distribution de croyances sur l'état du système et sur les séquences d'événements (dont les fautes) menant à cet état. Les ambiguïtés sont évaluées selon les valeurs des jetons et leurs places. Ce chapitre détaille également l'algorithme SSA qui instaure un **compromis entre la précision des hypothèses de diagnostic et les performances calculatoires de la fonction de diagnostic**.

Le Chapitre 4 débute par une proposition d'intégration des méthodes de diagnostic et de pronostic basées sur les HPPN. La méthode de pronostic repose sur la génération d'un pronostiqueur HPPN à partir du modèle HPPN du système. Le pronostiqueur HPPN prédit l'évolution du système dans le futur en fonction des entrées qui correspondent au plan de mission sur un certain horizon de prédiction. Le processus de pronostic se base sur le diagnostic courant du système et l'algorithme SSA pour retourner une distribution de croyances sur les futurs états possibles et les séquences de fautes menant à ces états. Le chapitre propose également une **intégration avancée** des modules de diagnostic et de pronostic ayant pour but d'améliorer les résultats de surveillance vis-à-vis d'objectifs relatifs à la gestion de santé du système.

Tout au long des Chapitres 2, 3 et 4, les concepts théoriques ont été illustrés sur un exemple simple de robot mobile et la méthodologie a été appliquée sur un système de réservoirs d'eau, un cas d'étude simulé. Les résultats de la fonction de diagnostic illustrent comment le diagnostiqueur HPPN est robuste aux observations incertaines, manquantes et fausses. Les résultats de la fonction de pronostic illustrent comment le pronostiqueur HPPN exploite le diagnostic courant, le modèle de comportement et le modèle de dégradation du système pour retourner une distribution de croyances sur la durée de vie résiduelle (RUL).

Le Chapitre 5 propose une expérimentation de la méthodologie sur le rover planétaire K11, un banc d'essai développé par le NASA Ames Research Center à des fins de diagnostic et de pronostic. Cette application met en avant l'applicabilité de la méthode sur des systèmes réels et sa robustesse face à des données entachées d'incertitudes. Elle illustre également l'utilité du compromis entre la précision des résultats et les performances calculatoires de la méthode instauré par l'algorithme SSA. Enfin, l'expérimentation met en évidence les limites de la méthode vis-à-vis particulièrement des performances de la fonction de pronostic qui est basée sur la simulation pas à pas du système dans le futur.

Ce travail propose donc un cadre formel pour réaliser la modélisation, le diagnostic et le pronostic de systèmes sous incertitude en utilisant les réseaux de Petri hybrides particuliers et démontre son applicabilité et son utilisabilité sur deux cas d'étude.

Discussions et Perspectives

Tout au long de ce manuscrit, des hypothèses sur le problème ou sa résolution sont posées. Il serait intéressant de les remettre en cause pour des développements à venir.

Dynamiques hybrides observables

L'hypothèse la plus importante de ces travaux est la non observabilité de la dynamique hybride. Dans nos cas d'application, les dynamiques hybrides servent à estimer la dégradation du système comme des probabilités d'occurrences de fautes dont les évolutions dépendent d'informations continues et discrètes. Cependant, le comportement du système pourrait aussi dépendre de telles informations. Dans notre cadre de modélisation (Section 2.3), le comportement du système est représenté par une dynamique discrète et des dynamiques continues en interaction. L'évolution de l'état continu est donc différente pour chaque mode mais ne dépend pas de la séquence d'événements survenue sur le système. Or, ce genre de dépendance peut exister dans le cadre d'étude de stabilité par exemple. Dans une perspective holistique de la modélisation du système, deux dynamiques hybrides pourrait être associées à chaque mode (au lieu d'une dynamique continue et d'une dynamique hybride). La première serait corrigable à l'aide des observations (comme l'est la dynamique continue dans notre approche) et la seconde serait non corrigable (comme l'est la dynamique hybride dans notre approche). Ainsi, la réduction de la complexité lors de la génération de diagnostiqueur (voir Chapitre 3) garderait son intérêt vis-à-vis de la séparation des informations observables et non observables pour améliorer les performances calculatoires.

Résolution analytique des modèles de dégradation

Dans le Chapitre 4, nous avons choisi de simuler l'évolution de la dégradation en parallèle de la simulation de l'évolution du comportement avec les entrées futures sur le système. Le choix d'utiliser des techniques de simulation permet d'harmoniser l'exploitation des dynamiques de comportement et de dégradation dans la fonction de pronostic car un modèle peut toujours être simulé.

Certains modèles de dégradation ont l'avantage de pouvoir être exploités analytiquement. Dans un tel cas, il pourrait être intéressant de calculer rapidement un pronostic basé sur ces raisonnements analytiques, et de confronter ce résultat avec un pronostic plus précis issu de la simulation. Le résultat de pronostic analytique pourrait par exemple influencer la sélection des hypothèses durant l'étape d'initialisation du processus de pronostic présenté dans la Section 4.4.

Extension de la méthodologie aux systèmes hétérogènes

Dans la Section 2.4, nous formulons une hypothèse sur la méthodologie de modélisation du système hybride qui impose à une transition d'avoir une seule place de chaque type en entrée et en sortie. Cette hypothèse nous permet de simplifier la génération du diagnostiqueur et du pronostiqueur car la structure du modèle hybride est maîtrisée et toutes les combinaisons de relations possibles entre places et transitions ne sont pas considérées.

Cette hypothèse réduit cependant très largement les possibilités de modélisation avec les HPPN. On peut imaginer par exemple des sous parties du réseau utilisant uniquement les places symboliques, sans interaction avec des places continues ou hybrides. Le cadre de modélisation s'étendrait alors à des systèmes hétérogènes intégrant des composants ayant des dynamiques de types différents. Certains composants seraient modélisés avec seulement des dynamiques continues, d'autres parties du modèle exploiteraient uniquement les places discrètes, etc.

Dans ce cas, une réflexion approfondie sur les générations du diagnostiqueur et du pronostiqueur, ainsi que sur leurs processus doit être réalisée. Une telle approche pourrait par ailleurs mener à la distribution et/ou la décentralisation des fonctions de diagnostic et de pronostic. Dans la même direction, il pourrait être intéressant de réfléchir à l'utilisation d'autres mécanismes des réseaux de Petri, tels que le parallélisme et la synchronisation.

Extension de la méthodologie aux systèmes réparables

Dans ces travaux, on se place dans un contexte où le système étudié n'est pas réparable. La réparation d'un système est complexe à modéliser car l'état du système et sa dégradation sont

difficile à estimer après des périodes de maintenance plus ou moins longues durant lesquelles il a évolué dans des conditions potentiellement inconnues.

Des actions de maintenance, telles que la réparation ou le remplacement d'un composant du système, peuvent également modifier les modes opérationnels. Dans de tels cas, le cadre de modélisation basée sur les HPPN devrait être étendu pour que les modes opérationnels puissent évoluer au cours du temps.

Exploitation des résultats de surveillance

Ces travaux, bien que définis dans un contexte de gestion de santé, se limitent à la surveillance de la santé des systèmes. Une des perspectives de ces travaux est donc l'exploitation des résultats des méthodes de diagnostic et de pronostic ayant des formes de distributions de croyances. Cette exploitation peut s'orienter vers plusieurs objectifs :

- l'utilisation des résultats du diagnostic et du pronostic pour améliorer la surveillance de santé vis-à-vis d'objectifs relatifs à la gestion de santé du système (abordée dans la Section 4.8),
- l'intégration d'une fonction de diagnostic actif (qui permet d'effectuer des actions dans le but de raffiner le résultat de diagnostic) qui pourrait être également influencée par les résultats de pronostic,
- la replanification et la reconfiguration du système en autonomie, dans l'objectif d'assurer la mission du système, de minimiser son usure et/ou son indisponibilité.

Les perspectives de ces travaux de thèse s'orientent donc dans cinq directions :

- la considération dans le modèle d'une dynamique hybride observable,
- la proposition d'une méthode de pronostic couplant la résolution analytique et la simulation de modèle de dégradation pour un résultat de pronostic plus précis,
- l'extension de la méthodologie aux systèmes hétérogènes et l'intégration de mécanismes des réseaux de Petri tels que le parallélisme et la synchronisation,
- l'extension de la méthodologie aux systèmes réparables,
- l'exploitation des résultats des fonctions de diagnostic et de pronostic prenant des formes de distributions de croyances.

Liste des figures

1	Architecture de gestion de santé de système intégrant diagnostic et pronostic.	13
2	Évolution unidirectionnelle d'un système sans maintenance ou action de réparation.	15
1.1	Une architecture hiérarchique générique à trois niveaux : fonctionnel, exécutif et décisionnel.	25
1.2	Une architecture orientée-santé générique.	27
1.3	Classification des méthodes de diagnostic.	28
1.4	Principe du diagnostic basé sur les modèles.	30
1.5	Évolution au cours du temps d'une distribution de probabilité représentant l'état continu du système.	31
1.6	Exemple d'un modèle automate et de son diagnostiqueur [Soldani 08]	34
1.7	Exemple d'un réseau de Petri.	36
2.1	Description du robot mobile.	47
2.2	Exemple illustratif d'un HPPN au temps $k = 0$	49
2.3	Illustration des clusters de particules.	52
2.4	Illustration du tirage d'une transition au temps k	54
2.5	Proposition de modélisation avec HPPN de l'évolution de la santé du robot mobile présenté en Section 2.2.	56
2.6	Système des réservoirs d'eau.	59
2.7	Description de l'évolution de la santé du système des réservoirs d'eau.	61
2.8	Modèle HPPN de l'évolution de la santé du système des réservoirs d'eau.	63
2.9	Correspondance entre la description du systèmes des réservoirs d'eau en Figure 2.7 et les éléments de son modèle HPPN en Figure 2.8.	64
2.10	Fonction de répartition de la loi de Weibull pour $\beta = 1$ et différentes valeurs de η	66
3.1	Illustration du pseudo-tirage d'une transition au temps k	72
3.2	Comparaison entre les données mesurées de la tension aux bornes d'une batterie et son modèle de déchargement non bruité.	72
3.3	Illustration de l'étape de séparation en deux niveaux du diagnostiqueur sur un HPPN simple.	76
3.4	Illustration de l'étape de séparation en deux niveaux du diagnostiqueur HPPN du robot mobile.	77
3.5	Illustration de l'étape de fusion des transitions du diagnostiqueur sur un HPPN simple.	79
3.6	Illustration de l'étape de fusion des transitions du diagnostiqueur HPPN du robot mobile.	80
3.7	Diagnostiqueur HPPN du robot présenté en Section 2.2.	81

3.8	Processus du diagnostiqueur HPPN.	82
3.9	Illustration de l'étape de prédiction du processus du diagnostiqueur HPPN du robot mobile.	85
3.10	En foncé : différence symétrique de deux ensembles A et B , i.e. $(A \setminus B) \cup (B \setminus A)$	87
3.11	Fonction gaussienne pour $\mu = 0$ et $\sigma = 1$	87
3.12	Fonction gaussienne pour $\mu = 0$ et différentes valeurs de σ	88
3.13	Illustration de deux hypothèses partageant le même cluster de particules.	91
3.14	Illustration de l'étape de correction du processus du diagnostiqueur HPPN du robot mobile.	93
3.15	Illustration du calcul des précisions associées à 200 hypothèses avec l'algorithme SSA avec différents paramètres d'échelle ($\rho^{min}, \rho^{max}, \rho^{tot}$).	97
3.16	Diagnostiqueur HPPN du système des réservoirs d'eau.	101
3.17	Mode du système simulé à chaque instant pour le scénario 1.	102
3.18	Observations continues pour le scénario 1.	103
3.19	Distributions de croyances sur le mode du système à chaque instant pour le scénario 1.	104
3.20	Distribution de croyances sur la trajectoire passée du système à 18600s pour le scénario 1.	104
3.21	Distribution de croyances sur la trajectoire passée du système à 206520s pour le scénario 1.	105
3.22	Distributions de croyances sur le mode du système à chaque instant pour le scénario 2.	107
3.23	Distribution de croyances sur la trajectoire passée du système à 206520s pour le scénario 2.	108
3.24	Mode du système simulé à chaque instant pour le scénario 3.	109
3.25	Distributions de croyances sur le mode du système à chaque instant pour le scénario 3.	109
3.26	Distribution de croyances sur la trajectoire passée du système à 214020s pour le scénario 3.	110
4.1	Illustration de la génération du pronostiqueur HPPN du robot mobile.	118
4.2	Processus du pronostiqueur HPPN.	122
4.3	Illustration du processus du pronostiqueur HPPN du robot mobile.	125
4.4	Pronostiqueur HPPN du robot mobile à la fin du processus de pronostic avec le marquage $\hat{M}_{k_{EOP} k}$	126
4.5	Pronostiqueur HPPN du système des réservoirs d'eau.	129
4.6	Estimations du RUL du système pour le scénario 1.	131
4.7	Distribution de croyances sur les trajectoires passée et future du système à 162060s pour le scénario 1.	132
4.8	Estimations du RUL du système pour le scénario 2.	134
4.9	Estimations du RUL du système pour le scénario 3.	135
4.10	Intégration avancée des méthodes de diagnostic et de pronostic basée sur les HPPN.	137
5.1	Le rover K11 [Balaban <i>et al.</i> 13].	140
5.2	Description simplifiée de l'évolution de la santé du rover K11.	143
5.3	Routes empruntées par le rover dans le scénario nominal et avec l'apparition d'une charge parasite sur la batterie, avec et sans replanification [Sweet <i>et al.</i> 14].	146

5.4	Distributions de croyances sur le mode du système à chaque instant avec pour le scénario nominal - $(40, 80, 1500)_{\Delta}$.	148
5.5	Tension mesurée aux bornes de la batterie du rover et les estimations de deux hypothèses du diagnostiqueur à 650s du scénario nominal.	149
5.6	Estimations du RUL du système pour le scénario nominal - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.	149
5.7	Intensité du courant mesurée dans le bus d'alimentation du rover durant le scénario de faute sans replanification.	150
5.8	Tension mesurée aux bornes de la batterie du rover durant le scénario de faute sans replanification.	151
5.9	Distributions de croyances sur le mode du système à chaque instant avec pour le scénario de faute sans replanification - $(40, 80, 1500)_{\Delta}$.	151
5.10	Estimations du RUL du système pour le scénario de faute sans replanification - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.	152
5.11	Comparaison de la tension mesurée aux bornes de la batterie durant le scénario de faute sans replanification et de simulations de modèles avec des charges parasite de 2A et 2.5A.	152
5.12	Distribution de croyances sur les trajectoires passée et future du système à 781s pour le scénario avec faute sans replanification - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.	154
5.13	Distributions de croyances sur le mode du système à chaque instant avec pour le scénario de faute avec replanification - $(40, 80, 1500)_{\Delta}$.	155
5.14	Estimations du RUL du système pour le scénario de faute avec replanification - $(40, 80, 1500)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.	155
5.15	Degrés de croyance du mode courant et estimations du RUL - $(40, 80, 400)_{\Delta}$ et $(1, 5, 100)_{\Pi}$.	158
5.16	Degrés de croyance du mode courant et estimations du RUL - $(20, 60, 400)_{\Delta}$ et $(1, 1, 1)_{\Pi}$.	159

Liste des tables

1.1	Position des principales architectures concernant l'intégration de diagnostic ou de pronostic à la supervision de la mission.	24
1.2	Description des niveaux de criticité de la stratégie FDIR.	26
2.1	Commandes, mesures et types des fautes du système des réservoirs d'eau. . . .	59
2.2	Équations d'évolutions continues non bruitées du système des réservoirs d'eau.	65
2.3	Paramètres des dynamiques hybrides du système des réservoirs d'eau.	67
5.1	Commandes continues, mesures continues, et types des fautes du K11.	142
5.2	Performances calculatoires de la surveillance de santé basée sur les HPPN du rover K11 pour différents jeux de paramètres d'échelle.	156
5.3	Performances calculatoires de la surveillance de santé basée sur les HPPN du système des réservoirs d'eau.	156

Références

- [Alami *et al.* 98] R. Alami, R. Chatila, S. Fleury, M. Ghallab, et F. Ingrand. *An Architecture for Autonomy*. The Int. Journal of Robotics Research, vol. 17, no. 4, pages 315–337, 1998.
- [Antsaklis & Passino 89] P. Antsaklis et K. Passino. *Towards Intelligent Autonomous Control Systems : Architecture and Fundamental Issues*. Journal of Intelligent and Robotic Systems, pages 315–342, 1989.
- [Arulampalam *et al.* 02] M. S. Arulampalam, S. Maskell, N. Gordon, et T. Clapp. *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*. IEEE Transactions on Signal Processing, vol. 50, no. 2, pages 174–188, 2002.
- [Balaban *et al.* 13] E. Balaban, S. Narasimhan, M. J. Daigle, I. Roychoudhury, A. Sweet, C. Bond, J. R. Celaya, et G. Gorospe. *Development of a Mobile Robot Test Platform and Methods for Validation of Prognostics-Enabled Decision Making Algorithms*. Int. Journal of Prognostics and Health Management, vol. 4, no. 006, 2013.
- [Basile *et al.* 09] F. Basile, P. Chiacchio, et G. D. Tommasi. *Fault diagnosis and prognosis in Petri Nets by using a single generalized marking estimation*. Dans 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Spain, 2009.
- [Bayouhd *et al.* 08] M. Bayouhd, L. Travé-Massuyes, et X. Olive. *Hybrid systems diagnosis by coupling continuous and discrete event techniques*. Dans IFAC World Congress, pages 7265–7270, Korea, 2008.
- [Benedettini *et al.* 08] O. Benedettini, T. Baines, H. Lightfoot, et R. Greenough. *State-of-the-art in integrated vehicle health management*. Journal of Aerospace Engineering, vol. 223, no. G, 2008.
- [Berenjii & Wang 06] H. Berenjii et Y. Wang. *Case-Based Reasoning for Fault Diagnosis and Prognosis*. Dans IEEE International Conference on Fuzzy Systems, Canada, 2006.
- [Bonasso *et al.* 95] R. Bonasso, J. Firby, E. Gat, D. Kortenkamp, D. Miller, et S. M.G. *Experiences with an Architecture for Intelligent, Reactive Agents*. Journal of Experimental & Theoretical Artificial Intelligence, vol. 9, no. 2/3, 1995.
- [Boubour *et al.* 97] R. Boubour, C. Jard, A. Aghasaryan, E. Fabre, et A. Benveniste. *A Petri Net approach to fault detection and diagnosis in distributed systems. Part I : application to telecommunication networks, motivations, and modelling*. Dans 36th Conf. on Decision & Control, USA, 1997.
- [Brotherton *et al.* 00] T. Brotherton, G. Jahns, J. Jacobs, et D. Wroblewski. *Prognosis of Faults in Gas Turbine Engines*. Dans IEEE Aerospace Conference Proceedings, volume 6, pages 163–171, USA, 2000.
- [Buchanan & Shortliffe 84] B. Buchanan et E. Shortliffe. *Rule Based Expert Systems : The Mycin Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley Longman Publishing Co., Inc., 1984.

- [Cabasino *et al.* 14] M. P. Cabasino, A. Giua, et C. Seatzu. *Diagnosability of Discrete-Event Systems Using Labeled Petri Nets*. IEEE Transactions on Automation Science and Engineering, vol. 11, no. 1, pages 144–153, 2014.
- [Camci *et al.* 07] F. Camci, G. Valentine, et K. Navarra. *Methodologies for Integration of PHM Systems with Maintenance Data*. Dans IEEE Aerospace Conference Proceedings, 2007.
- [Cao 89] X.-R. Cao. *The predictability of discrete event systems*. IEEE Transactions on Automatic Control, vol. 34, no. 11, pages 1168–1171, 1989.
- [Cardoso *et al.* 99] J. Cardoso, R. Valette, et D. Dubois. *Possibilistic Petri nets*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 29, no. 5, pages 573–582, 1999.
- [Cassandras & Lafortune 09] C. G. Cassandras et S. Lafortune. *Introduction to Discrete Event Systems*. Springer Science & Business Media, 2009.
- [Chanthery & Ribot 13] E. Chanthery et P. Ribot. *An Integrated Framework for Diagnosis and Prognosis of Hybrid Systems*. Dans 3rd Workshop on Hybrid Autonomous System, Italy, 2013.
- [Chanthery 05] E. Chanthery. *Planification de mission pour un véhicule aérien autonome*. Thèse de doctorat, École nationale supérieure de l'aéronautique et de l'espace (ISAE), Toulouse, France, 2005.
- [Chi *et al.* 13] Q. C. Chi, B. Bole, E. Hogge, S. Vazquez, M. Daigle, J. Celaya, et A. Weber. *Battery charge depletion prediction on an electric aircraft*. Dans Annual Conf. of the PHM Society, 2013.
- [Cordier *et al.* 04] M.-O. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, et L. Travé-Massuyès. *Conflicts versus analytical redundancy relations : a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 5, pages 2163–2177, 2004.
- [Daigle & Goebel 13] M. J. Daigle et K. Goebel. *Model-based prognostics with concurrent damage progression processes*. IEEE Transactions on Systems, Man, and Cybernetics : Systems, vol. 43, no. 3, pages 535–546, 2013.
- [Daigle & Kulkarni 14] M. Daigle et C. S. Kulkarni. *A battery health monitoring framework for planetary rovers*. Dans IEEE Aerospace Conference, pages 1–9, 2014.
- [Daigle & Sankararaman 13] M. J. Daigle et S. Sankararaman. *Advanced methods for determining prediction uncertainty in model-based prognostics with application to planetary rovers*. 2013.
- [Daigle *et al.* 14a] M. Daigle, C. S. Kulkarni, et G. Gorospe. *Application of model-based prognostics to a pneumatic valves testbed*. Dans IEEE Aerospace Conference, pages 1–8, 2014.
- [Daigle *et al.* 14b] M. Daigle, I. Roychoudhury, et A. Bregon. *Integrated Diagnostics and Prognostics for the Electrical Power System of a Planetary Rover*. Dans Annual Conf. of the PHM Society, USA, 2014.
- [Daigle *et al.* 15a] M. Daigle, I. Roychoudhury, et A. Bregon. *Model-based Prognostics of Hybrid Systems*. Dans Annual Conf. of the PHM Society, USA, 2015.
- [Daigle *et al.* 15b] M. Daigle, S. Sankararaman, et C. S. Kulkarni. *Stochastic Prediction of Remaining Driving Time and Distance for a Planetary Rover*. Dans IEEE Aerospace Conf., 2015.

- [Daroogheh et al. 15] N. Daroogheh, A. Baniamerian, N. Meskin, et K. Khorasani. *A Hybrid Prognosis and Health Monitoring Strategy by Integrating Particle Filters and Neural Networks for Gas Turbine Engines*. Dans IEEE Conference on Prognostics and Health Management (PHM), 2015.
- [David & Alla 05] R. David et H. Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, 2005.
- [Debouk et al.] R. Debouk, S. Lafortune, et D. Teneketzis. *A coordinated decentralized protocol for failure diagnosis of discrete event systems*. Dans Proceedings of the Workshop on Discrete Event Systems (WODES'98).
- [Ding 14] S. X. Ding. *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. Springer, 2014.
- [Djeziri et al. 07] M. Djeziri, R. Merzouki, B. Bouamama, et G. Dauphin-Tanguy. *Robust fault diagnosis by using bond graph approach*. IEEE/ASME Transactions on Mechatronics, vol. 12, no. 6, pages 599–611, 2007.
- [Dotoli et al. 08] M. Dotoli, A. Giua, C. Seatzu, et M. P. Fanti. *Modelling systems by hybrid Petri nets : an application to supply chains*. Petri Net, Theory and Application. Advanced Robotic Systems, 2008.
- [Dousson et al. 93] C. Dousson, P. Gaborit, et M. Ghallab. *Situation recognition : representation and algorithms*. Dans IJCAI : International Joint Conference on Artificial Intelligence, France, 1993.
- [Dubuisson 90] B. Dubuisson. *Diagnostic et reconnaissance de formes*. Hermes, 1990.
- [Engel et al. 00] S. Engel, B. Gilmartin, K. Bongort, et A. Hess. *Prognostics, the real issues involved with predicting life remaining*. Dans Aerospace Conf., IEEE, volume 6, pages 457–469, 2000.
- [Gaudel et al. 14a] Q. Gaudel, E. Chanthery, P. Ribot, et E. Le Corronc. *Hybrid systems Diagnosis using modified particle Petri nets*. Dans 25th Int. Workshop on Principles of Diagnosis, Austria, 2014.
- [Gaudel et al. 14b] Q. Gaudel, E. Chanthery, et P. Ribot. *Health Monitoring of Hybrid Systems Using Hybrid Particle Petri Nets*. Dans Annual Conf. of the PHM Society, USA, 2014.
- [Gaudel et al. 15a] Q. Gaudel, E. Chanthery, et P. Ribot. *Hybrid Particle Petri Nets for Systems Health Monitoring under Uncertainty*. Int. Journal of Prognostics and Health Management, vol. 6, no. 022, 2015.
- [Gaudel et al. 15b] Q. Gaudel, P. Ribot, et E. Chanthery. *Vers une architecture de surveillance de santé d'un système hybride sous incertitudes*. Dans Modélisation des Systèmes Réactifs, France, 2015.
- [Gaudel et al. 16] Q. Gaudel, P. Ribot, E. Chanthery, et M. J. Daigle. *Health Monitoring of a Planetary Rover Using Hybrid Particle Petri Nets*. Dans 37th International Conference on Application and Theory of Petri Nets and Concurrency, Poland, 2016.
- [Genc & Lafortune 07] S. Genc et S. Lafortune. *Distributed Diagnosis of Place-Bordered Petri Nets*. IEEE Transactions on Automation Science and Engineering, vol. 4, no. 2, pages 206–219, 2007.
- [Genc & Lafortune 09] S. Genc et S. Lafortune. *Predictability of event occurrences in partially-observed discrete-event systems*. Automatica, vol. 45, no. 2, pages 301–311, 2009.
- [Gertler 98] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Decker, 1998.
- [Giua & Seatzu 02] A. Giua et C. Seatzu. *Observability of Place/Transition Nets*. IEEE Transactions on Automatic Control, vol. 47, no. 9, pages 1424–1437, 2002.

- [Goh et al. 06] K. Goh, B. Tjahjono, T. Baines, et S. Subramaniam. *A Review of Research in Manufacturing Prognostics*. Dans Proceedings of the IEEE International Conference on Industrial Informatics, pages 417–422, USA, 2006.
- [Gouriveau et al. 13] R. Gouriveau, E. Ramasso, et N. Zerhouni. *Strategies to face imbalanced and unlabelled data in PHM applications*. Chemical Engineering Transactions, vol. 33, pages 115–120, 2013.
- [Grabowski 15] R. Grabowski. *Big Picture for Autonomy Research in DoD*. Dans Soft and Secure Systems and Software Symposium, 2015.
- [Hamilton et al. 07] K. Hamilton, D. Lane, K. Brown, J. Evans, et N. Taylor. *An Integrated Diagnostic Architecture for Autonomous Underwater Vehicles*. Journal of Field Robotics, vol. 6, no. 24, pages 497–526, 2007.
- [Hammouriand et al. 99] H. Hammouriand, M. Kinnaert, et E. Yaagoubi. *Observer-based approach to fault detection and isolation for non-linear systems*. IEEE Transactions and Automatic Control, vol. 44, pages 1879–1884, 1999.
- [Henzinger 96] T. Henzinger. *The theory of hybrid automata*. Dans 11th Annual IEEE Symposium on Logic in Computer Science, pages 278–292, 1996.
- [Hofbauer & Williams 02] M. Hofbauer et B. Williams. *Mode estimation of probabilistic hybrid system*. Hybrid Systems : Computation and Control. Lecture Notes in Computer Science, vol. 2289, pages 253–266, 2002.
- [Isermann 97] R. Isermann. *Supervision, fault-detection and fault-diagnosis methods. An introduction*. Control Engineering Practice, vol. 5, pages 639–652, 1997.
- [Jackson 88] P. Jackson. Introduction to Expert Systems. Addison-Wesley Longman Publishing Co., Inc., 1988.
- [Javed et al. 15] K. Javed, R. Gouriveau, et N. Zerhouni. *A new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering*. IEEE transactions on cybernetics, vol. 45, no. 12, pages 2626–2639, 2015.
- [Jianxiong et al. 13] W. Jianxiong, X. Xudong, B. Xiaoying, L. Chuang, K. Xiangzhen, et L. Jianxiang. *Performability analysis of avionics system with multilayer HM/FM using stochastic Petri nets*. Chinese Journal of Aeronautics, vol. 26, no. 2, pages 363–377, 2013.
- [Jouin et al. 16] M. Jouin, R. Gouriveau, D. Hissel, M. Péra, et N. Zerhouni. *Joint Particle Filters Prognostics for Proton Exchange Membrane Fuel Cell Power Prediction at Constant Current Solicitation*. IEEE Transactions On Reliability, vol. 65, no. 1, pages 336–349, 2016.
- [Julier & Uhlmann 97] S. J. Julier et J. K. Uhlmann. *New extension of the Kalman filter to nonlinear systems*. Dans AeroSense'97, pages 182–193. International Society for Optics and Photonics, 1997.
- [Julier & Uhlmann 04] S. J. Julier et J. K. Uhlmann. *Unscented filtering and nonlinear estimation*. volume 92, pages 401–422, 2004.
- [Kirkland et al. 04] L. Kirkland, T. Pombo, K. Nelson, et F. Berghout. *Avionics Health Management : Searching for the Prognostics Grail*. Dans IEEE Aerospace Conference Proceedings, volume 5, pages 3448–3454, 2004.
- [Kumar & Takai 10] R. Kumar et S. Takai. *Decentralized prognosis of failures in discrete event systems*. IEEE Transactions on Automatic Control, vol. 55, no. 1, pages 48–59, 2010.
- [Lachat et al. 06] D. Lachat, A. Krebs, T. Thuerer, et R. Siegwart. *Antarctica Rover Design And Optimization For Limited Power Consumption*. Dans 4th IFAC Symposium on Mechatronic Systems, 2006.

- [Lefebvre 14] D. Lefebvre. *Fault Diagnosis and Prognosis With Partially Observed Petri Nets*. IEEE Trans. on Systems, Man, and Cybernetics, vol. 44, no. 10, pages 1413–1424, 2014.
- [Lesire & Tessier 05] C. Lesire et C. Tessier. *Particle Petri nets for aircraft procedure monitoring under uncertainty*. Dans Applications and Theory of Petri Nets, pages 329–348. Springer, 2005.
- [Liao & Köttig 14] L. Liao et F. Köttig. *Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction*. IEEE Transactions On Reliability, vol. 63, no. 1, pages 191–207, 2014.
- [Liu & West 01] J. Liu et M. West. *Combined parameter and state estimation in simulation-based filtering*. Dans Sequential Monte Carlo methods in practice, pages 197–223. Springer, 2001.
- [Madalinski & Khomenko 12] A. Madalinski et V. Khomenko. *Predictability verification with parallel LTL-X model checking based on Petri net unfoldings*. IFAC Proceedings Volumes, vol. 45, no. 20, pages 1232–1237, 2012.
- [Muscettola et al. 98] N. Muscettola, P. Nayak, B. Pell, et B. Williams. *Remote Agent : To Boldly Go Where No AI System Has Gone Before*. Artificial Intelligence, vol. 103, no. 1-2, pages 5–48, 1998.
- [Narasimhan et al. 12] S. Narasimhan, E. Balaban, M. Daigle, I. Roychoudhury, A. Sweet, J. Celaya, et K. Goebel. *Autonomous Decision Making for Planetary Rovers Using Diagnostic and Prognostic Information*. Dans 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pages 289–294, Mexico, 2012.
- [Olive et al. 11] X. Olive, S. Clerc, et D. Losa. *Smart architecture for highly available, robust and autonomous satellite*. Dans 18th IFAC World Congress, 2011.
- [Orchard et al. 08] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, et G. Vachtsevanos. *Advances in uncertainty representation and management for particle filtering applied to prognostics*. Dans Prognostics and health management, pages 1–6. IEEE, 2008.
- [Ribot 09] P. Ribot. *Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes*. Thèse de doctorat, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), Toulouse, France, 2009.
- [Ribot et al. 13] P. Ribot, Y. Pencolé, et M. Combacau. *Generic characterization of diagnosis and prognosis for complex heterogeneous systems*. Int. Journal of Prognostics and Health Management, vol. 4, no. 023, 2013.
- [Roemer et al. 05] M. Roemer, C. Byington, G. Kacprzynski, et G. Vachtsevanos. *An Overview of Selected Prognostic Technologies with Reference to an Integrated PHM Architecture*. Dans The First International Forum on Integrated System Health Engineering and Management in Aerospace, 2005.
- [Roychoudhury & Daigle 11] I. Roychoudhury et M. Daigle. *An Integrated Model-Based Diagnostic and Prognostic Framework*. Dans 22nd Int. Workshop on Principle of Diagnosis, Germany, 2011.
- [Ru & Hadjicostis 09] Y. Ru et C. N. Hadjicostis. *Fault Diagnosis in Discrete Event Systems Modeled by Partially Observed Petri Nets*. Discrete Event Dynamic Systems, vol. 19, no. 4, pages 551–575, 2009.
- [Saha & Goebel 11] B. Saha et K. Goebel. *Model adaptation for prognostics in a particle filtering framework*. International Journal of Prognostics and Health Management, vol. 2, page 61, 2011.
- [Sampath et al. 95] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, et D. Teneketzis. *Diagnosability of discrete-event systems*. IEEE Transactions on Automatic Control, vol. 40, no. 9, pages 1555–1575, 1995.

- [Sampath *et al.* 96] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, et D. Teneketzis. *Failure Diagnosis Using Discrete-Event Models*. IEEE Transactions on Control Systems Technology, vol. 4, no. 2, pages 105–124, 1996.
- [Soldani 08] S. Soldani. *Vers le diagnostic embarqué de défaillances dans les systèmes à événements discrets : application au domaine automobile*. Thèse de doctorat, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), Toulouse, France, 2008.
- [Soldani *et al.* 07] S. Soldani, M. Combacau, A. Subias, et J. Thomas. *On-board diagnosis system for intermittent fault : Application in automotive industry*. Dans 7th IFAC Int. Conf. on Fieldbuses and Networks in Industrial and Embedded Systems, volume 7-1, pages 151–158, 2007.
- [Sprauel 16] J. Sprauel. *Conception sûre et optimale de systèmes dynamiques critiques auto-adaptatifs soumis à des événements redoutés probabilistes*. Thèse de doctorat, École nationale supérieure de l'aéronautique et de l'espace (ISAE), Toulouse, France, 2016.
- [Staroswiecki *et al.* 91] M. Staroswiecki, V. Cocquempot, et J. Cassar. *Observer based and parity space approaches for failure detection and identification*. Dans IMACSIFAC International Symposium, pages 536–541, 1991.
- [Su *et al.* 02] R. Su, W. Wonham, J. Kurien, et X. Koutsoukos. *Distributed diagnosis for qualitative systems*. Dans Proceedings of international workshop on discrete event systems, 2002.
- [Sweet *et al.* 14] A. Sweet, G. Gorospe, M. Daigle, J. R. Celaya, E. Balaban, I. Roychoudhury, et S. Narasimhan. *Demonstration of Prognostics-Enabled Decision Making Algorithms on a Hardware Mobile Robot Test Platform*. Dans Annual Conf. of the PHM Society, USA, 2014.
- [Thrun *et al.* 01] S. Thrun, D. Fox, W. Burgard, et F. Dellaert. *Robust Monte Carlo localization for mobile robots*. Artificial Intelligence, vol. 128, no. 1, pages 99–141, 2001.
- [U.S Air Force 15] U.S Air Force. *Autonomous Horizon : System Autonomy in the Air Force - A Path to the Future - Volume I : Human-Autonomy Teaming*. United States Air Force - Office of the Chief Scientist, 2015.
- [van der Merwe *et al.* 00] R. van der Merwe, A. Doucet, N. De Freitas, et E. Wan. *The unscented particle filter*. Dans NIPS, volume 2000, pages 584–590, 2000.
- [Venkatasubramanian *et al.* 03] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, et K. Yin. *A review of process fault detection and diagnosis - Part III : Process history based methods*. Computers and Chemical Engineering, vol. 27, pages 327–346, 2003.
- [Vianna & Yoneyama 15] W. O. L. Vianna et T. Yoneyama. *Interactive Multiple-Model Application for Hydraulic Servo Valve Health Monitoring*. Dans Annual Conf. of the PHM Society, USA, 2015.
- [Villemeur 88] A. Villemeur. *Sûreté de fonctionnement des systèmes industriels : fiabilité, facteurs humains, informatisation*. Eyrolles, 1988.
- [Vinson *et al.* 13] G. Vinson, P. Ribot, T. Prado, et M. Combacau. *A generic diagnosis and prognosis framework : application to permanent magnets synchronous machines*. Dans IEEE Prognostics and System Health Management Conf., pages 1039–1044, Italy, 2013.
- [voiture-autonome.net 15] voiture-autonome.net. *Les 5 niveaux d'autonomie d'un véhicule*. <http://www.voiture-autonome.net/legislation/5-niveaux-autonomie-vehicule-241.html>, 2015. Date : 2015-10-15.
- [Wilkinson *et al.* 04] C. Wilkinson, D. Humphrey, B. Vermeire, et J. Houston. *Prognostic and Health Management for Avionics*. Dans IEEE Aerospace Conference Proceedings, volume 5, pages 3435–3446, 2004.

- [Yang et al. 07] X. Yang, K. Shi, T. Huang, et K. Xing. *Combined Parameter and State Estimation in Particle Filtering*. Dans IEEE International Conference on Control and Automation, pages 1036–1039, 2007.
- [Yu et al. 11] M. Yu, D. Wang, M. Luo, et L. Huang. *Prognosis of Hybrid Systems With Multiple Incipient Faults : Augmented Global Analytical Redundancy Relations Approach*. IEEE Trans. on Systems, Man, and Cybernetics, vol. 41, Part A, no. 3, pages 540–551, 2011.
- [Yu et al. 14] M. Yu, D. Wang, et M. Luo. *Model-Based Prognosis for Hybrid Systems With Mode-Dependent Degradation Behaviors*. IEEE Trans. on Industrial Electronics, vol. 61, no. 1, pages 546–554, 2014.
- [Zabi et al. 13] S. Zabi, P. Ribot, et E. Chanthery. *Health Monitoring and Prognosis of Hybrid Systems*. Dans Annual Conf. of the PHM Society, 2013.
- [Zaidi et al. 06] A. Zaidi, N. Zanzouri, et M. Tagina. *Modelling and monitoring of hybrid systems by hybrid Petri nets*. Dans 10th WSEAS Int. Conf. on Systems, 2006.
- [Zhang et al. 09] B. Zhang, C. Sconyers, R. Patrick, et G. Vachtsevanos. *A Multi-Fault Modeling Approach for Fault Diagnosis and Failure Prognosis of Engineering Systems*. Dans Annual Conf. of the PHM Society, 2009.
- [Zouaghi et al. 11a] L. Zouaghi, A. Alexopoulos, A. Wagner, et E. Badreddin. *Modified particle petri nets for hybrid dynamical systems monitoring under environmental uncertainties*. Dans IEEE/SICE Int. Symposium on System Integration, pages 497–502, 2011.
- [Zouaghi et al. 11b] L. Zouaghi, A. Alexopoulos, A. Wagner, et E. Badreddin. *Probabilistic online-generated monitoring models for mobile robot navigation using modified Petri net*. Dans 15th Int. Conf. on Advanced Robotics, pages 594–599, 2011.
- [Zwingelstein 95] G. Zwingelstein. *Diagnostic des défaillances : Théorie et pratique pour les systèmes industriels*. Hermes, 1995.

Approche intégrée de diagnostic et de pronostic pour la gestion de santé des systèmes hybrides sous incertitude

Résumé

Cette étude s'inscrit dans le domaine de la gestion de santé des systèmes, qui vise à développer des outils d'aide à la maintenance, mais également à améliorer les prises de décision en autonomie des systèmes en cas de pannes. Dans ce cadre, des techniques de diagnostic déterminent si et pourquoi le système est en panne, alors que des techniques de pronostic déterminent quand les pannes vont survenir et leurs conséquences. Si elles semblent être corrélées, elles sont généralement étudiées séparément, car les échelles de temps manipulées par les deux processus sont très différentes.

Ces travaux ont pour objectif de développer un outil intégrant les méthodes de diagnostic et de pronostic pour la surveillance des systèmes hybrides, dont les dynamiques sont à la fois continues et discrètes. La méthodologie proposée, basée sur les réseaux de Petri hybrides particuliers, est appliquée sur un rover planétaire pour démontrer son utilisabilité en cas réel à travers la gestion des incertitudes liées au système et aux données.

Mots clés : Diagnostic, Pronostic, Systèmes hybrides, Surveillance basée sur les modèles, Gestion de santé, Incertitude.

Integrated approach of diagnosis and prognosis for hybrid system health management under uncertainty

Abstract

This study takes place in the field of system health management, which aims at developing maintenance aid tools, but also at improving the systems autonomous decision-making in case of failures. In this context, diagnostic techniques determine whether and why the system is down, while prognostic techniques determine when failures will occur and their consequences. If they seem to be correlated, they are usually studied separately because the time scales manipulated by the two processes are very different.

This work aims at developing a tool that integrates both diagnosis and prognosis methods for the monitoring of hybrid systems, whose dynamics are both continuous and discrete. The proposed methodology, based on hybrid particle Petri nets, is applied to a planetary rover to demonstrate its usability in real cases through the management of knowledge-based and data-based uncertainty.

Keywords : Diagnosis, Prognosis, Hybrid systems, Model-based monitoring, Health management, Uncertainty.