

THÈSE

présentée en vue de
l'obtention du titre de

DOCTEUR

de

**L'ÉCOLE NATIONALE SUPÉRIEURE
DE L'AÉRONAUTIQUE ET DE L'ESPACE**

**ÉCOLE DOCTORALE : Systèmes
SPÉCIALITÉ : Systèmes décisionnels**

par

Charles LESIRE

Estimation numérique-symbolique pour le suivi d'activités hybrides

Soutenue le 8 décembre 2006 devant le jury :

MM.	Ph. DAGUE	Président
	P. CASPI	Rapporteur
	J.-M. FLAUS	Rapporteur
	A. GIUA	
Mme	C. TESSIER	Directrice de thèse
M.	R. VALETTE	

Remerciements

Et voilà, on y est... Le moment de la rédaction des remerciements est arrivé. Certes le plus dur est passé : des mois et des mois de rédaction, de corrections, de rédaction, de corrections...

Et pourtant, le moment le plus difficile vient de commencer :

1. d'abord parce que la plupart des gens ne liront jamais de ma thèse que les remerciements :p
2. et puis ces dernières lignes marquent d'une pierre la fin de ma thèse, le passage de l'état de "bébé de la recherche" à l'âge de raison, où il est temps de m'émanciper, de m'épanouir, de m'accomplir.

Tout d'abord, je vais commencer par remercier (ben oui, ce sont des remerciements) ceux qui m'ont permis de franchir la dernière marche. Merci donc à mes deux rapporteurs, messieurs Paul Caspi et Jean-Marie Flaus, pour leurs remarques à la fois motivantes, pertinentes et enrichissantes. Merci à monsieur Philippe Dague d'avoir présidé mon jury. Et finalement merci à monsieur Robert Valette, qui a ainsi "validé" le cursus qu'il a initié quelques années plus tôt, en m'insufflant une curiosité pour les réseaux de Petri alors que je n'étais encore qu'à l'ENAC.

Merci également à tous ceux qui m'ont accueilli à l'ONERA, et bien entendu plus particulièrement à ceux de l'unité CD (et à leurs repas conviviaux), à Patrick¹ (et à ses cravates), aux danseurs du mercredi (et à leurs profs) et aux motards (et leurs sorties moto)... En résumé, en oubliant du monde, et pas par ordre (alphabétique ou autre), merci à Jean-François, Magali, Michel, Gérard, Laurent, Jean-Loup, Charles, Michel, Jean-Bernard, Jean-Henri, Patrick, Carsten, Philippe, Fabrice, Michel, Alain, Daniel, Elodie...

Et bien entendu, comment ne pas y penser, un énorme merci à "maman" Cath, qui durant ces 4 années a su m'élever, m'apprendre à marcher. Merci pour tes conseils, tes

¹Promis, je ne diffuserai pas les vidéos prises au mariage de Florent... je commence quand ? ;)

encouragements, ta bonne humeur, tes remarques, tes coups de stylo (verts pour la plupart). Que l'avenir conserve notre amitié et nous réserve de nombreuses collaborations !

J'ai également une pensée pour tous ceux qui ont distrait au jour le jour ces trois années de thèse : merci aux thésards du DCSD. Outre les soirées (crémaillères, anniversaires, ou sans raison) orgiaques, j'ai une pensée émue pour deux moments forts de toute journée de thésard qui se respectent :

1. la contrée² (à ce sujet, merci à Benjamin pour sa contribution à ce que l'on puisse coincer dans un environnement de qualité), qui a égayé chaque jour de ces trois années, soulevé des discussions, permis d'élaborer des stratégies, fait éclater des coups de gueule... finalement, nous a fait réfléchir quant même (pas plus d'une heure par jour, promis) ; au passage, merci aussi à Nicole qui nous a permis, à plusieurs reprises, d'accompagner ces parties endiablées de quelques cafés rudement gagnés ;
2. le café, moment solennel, où les drogués du labo se retrouvent dans un bureau, où un bizut (pas forcément drogué lui-même) leur a préparé une mixture plus ou moins buvable.

Pour tous ces moments, merci donc aux anciens (Fred, Nico, Sontsada, Yannick), au moins vieux (Sylvain, Flo, Elodie, Fabien, Guilhem, J-B), aux jeunes (Cédric, Sébastian, Olivier), aux minots (Sophie, Sylvain, Clément, Alex, Andreas, Vincent) et aux mioches (Greg, Manu, Greg, Nico).

J'aimerais cependant m'attarder un peu sur certains d'entre eux, qui sont devenus aujourd'hui bien plus que des "collègues", des amis pour la vie :

- Fred, mon grand-frère de lait, qui m'a suivi durant ces quatre années, et avec qui j'ai beaucoup travaillé... j'espère que nous aurons encore de nombreuses occasions de travailler ensemble, et je te souhaite beaucoup de bonheur dans ta récente vie familiale ;
- Olivier, mon "frère jumeau", avec qui j'ai partagé ce bureau (le meilleur bureau qu'on ait vu à l'Onera, la vérité) où régnait la bonne humeur, et où nous avons partagé des moments de doutes, de déceptions, de joie, de farniente, de réflexion, de café... à tes soirées inoubliables (surtout pour JB et Manu), à tes idées farfelues, aux réseaux de Petri ! Je te souhaite bonne chance pour la suite, et je suis persuadé que les kilomètres (et les rancœurs marseillo-parisiennes) n'altéreront en rien notre amitié ;
- Florent, qui a été mon partenaire de coinche préféré de cette dernière année (une fois mon mentor parti...), et m'a surtout fait le plaisir d'être le témoin d'un des moments les plus importants de ma vie (non, c'est pas ma thèse :p) ; une pensée évidemment pour Raoudha (pour Ra ou...) avec qui j'ai la chance de partager une passion excitante et angoissante... le théâtre. Merci à vous deux pour tous les bons

²la "coinche" du midi (sans jeu de mots)

- moments que nous avons partagé, et à tous ceux que nous partagerons encore ;
- Elodie et Fabien avec qui je (nous) partage de si bons moments de jeux, de théâtre, de parties de Bridge pour l’un, de réunions Agata pour l’autre :p Merci à tous les deux pour votre soutien en cette fin de thèse, et j’espère que nos filles seront, à notre image, les meilleures amies du monde.

Quand on lit les remerciements d’une thèse, comment éviter le moment sentimental, parfois “mièvre”. Comme dirais Anaïs :

Ce mélange de sentiments
Aromatisé aux fines herbes
Me fait sourire gentiment
Et finalement...

Et finalement, quand arrive ce moment, on ne peut s’y dérober, parce qu’on a jamais l’occasion de l’écrire, et rarement celle de le dire, et donc, du fond du cœur, merci à mes parents ! Merci, pour tout l’amour que vous m’avez donné, pour avoir fait de moi ce que je suis devenu aujourd’hui, autant professionnellement que personnellement. Merci pour vos encouragements répétés, votre enthousiasme naïf lors de bonnes nouvelles ou votre déception sincère lors des mauvaises. De cette thèse resterons imprégnés en moi les moments où vous avez, en vain, essayé de comprendre les réseaux de Petri et le filtrage particulière, essayé de suivre les questions lors de ma soutenance, donné de (trop) nombreux coups de téléphone, en anglais, aux hôtels lors de nos voyages.

Merci aussi à ma sœurette adorée et à tous ceux qui sont venus assister à ma soutenance.

Last but not least, une pensée émue pour Hélène, qui s’était sacrifiée, acceptant de travailler dur pour faire vivre notre ménage, avec la pensée illusoire qu’un jour je gagnerais beaucoup d’argent parce que j’avais fait une thèse ;)

Merci de m’avoir accompagné jour après jour, de m’avoir soutenu (surtout vers la fin), aimé. A tous les bons moments que l’on a partagé, aux nombreux voyages et choses merveilleuses que nous avons découvertes ensemble. A notre mariage, le plus beau jour de ma vie... avant que tu ne m’offres le plus beau cadeau de thèse que tu pouvais me faire : notre fille !

En conclusion, Hélène, je t’aime.

Et voilà, cette fois c’est bel et bien fini...

ou plutôt, ça commence !... TOURNEZ LA PAGE...

Table des matières

Introduction	xvii
1 Suivi de situation	1
1 Les systèmes continus	1
1.1 Le problème de filtrage	2
1.2 Filtrage bayésien	2
1.2.1 Prédiction	3
1.2.2 Recalage	3
1.3 Le filtre de Kalman	3
1.4 Le filtrage particulaire	5
1.5 Conclusions sur l'estimation continue	6
2 Les systèmes à événements discrets	7
2.1 Systèmes à événements discrets et logique	8
2.1.1 La logique pour la reconnaissance de scénarios	8
2.1.2 La logique possibiliste	9
2.1.3 Représentation des préférences	10
2.2 Modélisation du comportement des systèmes à événements discrets	10
2.2.1 Graphes et arbres	10
2.2.2 Graphes de contraintes	11
2.2.3 Réseaux bayésiens	12
2.2.4 Automates	15
2.2.5 Réseaux de Petri	17
3 Conclusion	21
3.1 Synthèse	21
3.2 Estimation numérique/symbolique	23
2 Systèmes hybrides et estimation	25
1 Système hybride	25
2 Filtrage multimodèle	26

2.1	Filtre de Kalman multimodèle	26
2.2	Filtrage particulière pour les systèmes hybrides	27
3	Réseaux bayésiens hybrides	27
4	Modèles discrets et filtres numériques	28
4.1	Automates hybrides	28
4.2	Réseaux de Petri hybrides	29
4.3	Estimation hybride	30
4.3.1	Automates hybrides probabilistes concurrents	30
4.3.2	Automates hybrides et filtrage particulière	31
4.3.3	Estimation hybride et réseaux de Petri : un bon ménage?	31
5	Vers un estimateur par réseaux de Petri hybrides	32
5.1	Décomposition en composants	33
5.2	Représentation de l'incertitude	33
5.3	Quelques exemples de systèmes hybrides	34
5.3.1	Thermostat : gestion de la température d'une pièce	34
5.3.2	Navigation dans un champ de vitesse	34
5.3.3	Partage de thermostats	35
6	Conclusion	36
3	Réseaux de Petri particuliers	39
1	Les réseaux de Petri particuliers	39
1.1	Définitions	39
1.2	Propriétés	41
1.2.1	Réseau de Petri sous-jacent	41
1.2.2	Graphe d'état numérique	42
1.3	Évolution du marquage	43
1.3.1	Évolution isoparticule	43
1.3.2	Évolution isomarquage	46
2	Quelques exemples	46
2.1	Navigation	46
2.1.1	Modélisation	47
2.1.2	Propriétés	49
2.1.3	Déplacement de l'objet	49
2.2	Le thermostat	52
2.2.1	Modélisation	52
2.2.2	Propriétés	53
2.2.3	Évolution de la température	54
3	Conclusion	55

4	Estimation par filtres de Kalman	57
1	Représentation de l'incertitude	57
1.1	Incertitude numérique	57
1.2	Propriétés	58
1.3	Incertitude symbolique	59
1.4	Propagation de l'incertitude	59
2	Principe d'estimation	60
2.1	Prédiction	60
2.1.1	Première ébauche	60
2.1.2	Améliorations	65
2.1.3	Distributions tronquées	67
2.1.4	Prédiction de la température	69
2.2	Recalage	73
2.2.1	Recalage numérique	73
2.2.2	Recalage symbolique	74
2.2.3	Recalage	76
3	Conclusions	76
5	Estimation par filtrage particulaire	79
1	Marquage particulaire	79
2	Principe d'estimation	80
2.1	Prédiction	80
2.1.1	Évolution isoparticule bruitée	80
2.1.2	Évolution isomarquage bruitée	82
2.1.3	Synthèse	82
2.1.4	Exemple : prédiction de la température	82
2.2	Recalage	86
2.2.1	Recalage numérique	86
2.2.2	Recalage symbolique	88
2.2.3	Synthèse	89
2.2.4	Exemple : recalage de la température	89
3	Conclusions	92
6	Algorithmique de l'estimateur	95
1	Synthèse	95
1.1	Les réseaux de Petri particuliers	95
1.2	Estimation de l'état d'un système hybride	95
2	Prédiction	96
2.1	Évolution isoparticule	96
2.2	Évolution isomarquage	101
2.3	Prédiction : synthèse	102
3	Recalage	102

3.1	Recalage numérique	102
3.2	Recalage symbolique	104
3.3	Recalage : synthèse	105
4	Évaluation expérimentale	106
4.1	Exhost-PIPE	106
4.2	Complexité en fonction de N	106
4.2.1	Problème de navigation	106
4.2.2	Le thermostat	108
4.3	Complexité en fonction de la taille du système	111
4.3.1	Problème de navigation	111
4.3.2	Le thermostat	114
5	Conclusion	115
7	Diagnostic et détection d'incohérences	117
1	Estimation de mode	117
1.1	Recalage numérique	118
1.2	Estimation du mode numérique	118
2	Diagnostic à base de modèles	119
2.1	Modélisation des modes défailants	119
2.2	Diagnostic de comportements défailants	120
3	Détection et prédiction d'incohérences	121
3.1	Incohérence numérique/symbolique	121
3.2	Graphe des marquages cohérents	122
3.3	Graphe de recalage	122
3.4	Stratégies de regroupement	124
3.4.1	Regroupement par place	124
3.4.2	Regroupement par poids	126
3.5	Détection d'incohérences	129
4	Conclusion	131
8	Simulations	133
1	Estimation de l'état d'un système	133
1.1	Navigation	133
1.2	Le thermostat	138
2	Évolution de l'erreur quadratique	139
2.1	Sensibilité au nombre de particules	140
2.2	Sensibilité au bruit	141
2.3	Sensibilité à l'initialisation	143
3	Estimation de mode numérique	145
3.1	Problème de navigation	145
3.2	Le thermostat	147
4	Détection d'incohérences	149

4.1	Détection d'incohérences en mode "nominal"	150
4.1.1	Graphe de recalage par places	150
4.1.2	Graphe de recalage par poids	151
4.1.3	Synthèse	151
4.2	Comportements défailants	152
4.2.1	Thermo24	153
4.2.2	Thermo26	153
5	Conclusion	154
9	Suivi de l'activité de pilotage	157
1	L'activité de pilotage	157
2	Conflits entre l'équipage et le pilote automatique	158
2.1	Déconnexion automatique non perçue	158
2.2	Consignes incohérentes	158
2.3	Buts différents de l'équipage et du pilote automatique	159
2.4	Mise en palier	159
3	Supervision du système hybride avion–équipage	160
3.1	Modélisation du scénario de mise en palier	160
3.2	Détection de conflits	161
3.2.1	Pilote numéro 1	162
3.2.2	Pilote numéro 2	164
3.2.3	Pilote numéro 3	166
4	Conclusion	168
	Conclusion	171
	Bibliographie	188
A	Le filtre de Kalman	189
1	Modèle discret du système	189
1.1	Equation d'évolution	189
1.2	Equation d'observation	190
2	Estimation optimale de l'état	190
2.1	Estimation par récurrence	190
2.2	Equations du filtre de Kalman	191
2.3	Propriétés	191
B	Le filtrage particulière	193
1	Introduction	193
2	Algorithme de base	194
2.1	Modèle d'état non-linéaire non-gaussien	194
2.2	Algorithme	194

C	Exhost-PIPE	197
1	De PIPE à Exhost-PIPE	197
1.1	PIPE	198
1.2	Conception d'Exhost-PIPE	198
1.3	Amélioration de la qualité	198
2	Exhost-PIPE	199
2.1	Réseaux de Petri particuliers	200
2.2	Estimation	201
2.3	Résultats	202
D	Problème de navigation	205
1	Description du problème	205
2	Instances	205
2.1	Caractéristiques	206
2.2	Description des instances	206
2.3	Taille des instances	210
E	Problème du thermostat	211
1	Description du problème	211
2	Instances	211

Table des figures

1	Systèmes critiques	xviii
1.1	Schéma d'une boucle estimation/contrôle	4
1.2	Fonction d'appartenance de "grande vitesse en ville"	9
1.3	Modèle de <i>Object – Deposite</i>	11
1.4	Chronique et graphe de contraintes du chargement d'une machine	12
1.5	Scénario d'attaque de banque	13
1.6	Réseau bayésien d'estimation du temps	14
1.7	Automate d'un thermostat	15
1.8	Modèle partiel représentant l'évolution d'une parcelle	16
1.9	Un réseau de Petri et le franchissement d'une transition	18
1.10	Estimation du marquage d'un réseau de Petri	20
2.1	BIO-Plex	30
2.2	Exemple de grille de navigation 3x3	35
3.1	Réseaux de Petri non-Zénon	45
3.2	Modélisation du problème de navigation	47
3.3	Marquage du réseau de Petri particulière de Navigation au temps 3	51
3.4	Différentes évolutions du mobile.	51
3.5	Modélisation du thermostat	52
3.6	Graphe des marquages accessibles du réseau sous-jacent	54
3.7	Évolution de la température	54
4.1	Marquage gaussien du thermostat	58
4.2	Évolution isomarquage du marquage $\widehat{\mathcal{M}}(0 0)$	61
4.3	Pseudo-tir de t_0	63
4.4	Pseudo-tir de t_3	64
4.5	Pseudo-tir de la transition t_0	68
4.6	Marquages accessibles du réseau du thermostat	70

4.7	Marquage final $\widehat{\mathcal{M}}'(0 0)$ de l'évolution isoparticule.	71
4.8	Calcul du marquage $\widehat{\mathcal{M}}(1 0)$	72
4.9	Marquage numérique recalé à l'instant 1	75
4.10	Marquage recalé au temps 1	75
5.1	Marquage initial $\widehat{\mathcal{M}}(0 0)$ du thermostat	83
5.2	$\widehat{\mathcal{M}}'(0 0)$ issu du pseudo-tir de t_0	84
5.3	Marquages accessibles du réseau du thermostat	84
5.4	Marquage terminal $\widehat{\mathcal{M}}^T(0 0)$	85
5.5	Marquage prédit $\widehat{\mathcal{M}}(1 0)$	86
5.6	Marquage pondéré $\widehat{\mathcal{M}}(1 0)$	90
5.7	Marquage rééchantillonné	91
5.8	Marquage recalé $\widehat{\mathcal{M}}(1 1)$	91
6.1	Réseau non-Zénon	100
6.2	Un exemple de pire cas pour la prédiction.	100
6.3	Évolution des algorithmes de prédiction en fonction de N	108
6.4	Évolution des algorithmes de recalage en fonction de N	108
6.5	Réseau de Petri du thermostat	109
6.6	Évolution des algorithmes de prédiction en fonction de N	109
6.7	Évolution des algorithmes de recalage en fonction de N	110
6.8	Évolution des algorithmes de prédiction en fonction de $ \mathcal{R} $	112
6.9	Évolution des algorithmes de recalage en fonction de $ \mathcal{R} $	113
6.10	Évolution des algorithmes de prédiction en fonction de $ \mathcal{R} $	114
6.11	Évolution des algorithmes de recalage numérique en fonction de $ \mathcal{R} $	115
6.12	Évolution des algorithmes de recalage symbolique en fonction de $ \mathcal{R} $	115
7.1	Marquage recalé $\widehat{\mathcal{M}}(1 1)$	119
7.2	Graphe des marquages cohérents du thermostat.	123
7.3	Graphe de recalage par places du thermostat	125
7.4	Graphe de recalage par poids du thermostat.	129
8.1	Estimation de la position de l'objet pour Nav01	134
8.2	Estimation de l'abscisse pour Nav01	135
8.3	Estimation de l'ordonnée pour Nav01	135
8.4	Estimation de la position de l'objet pour Nav13	136
8.5	Estimation de l'abscisse pour Nav13	136
8.6	Estimation de l'ordonnée pour Nav13	137
8.7	Estimation de la température de la pièce	138
8.8	Estimation de la température extérieure	138
8.9	Erreur d'estimation pour Nav01	139
8.10	Erreur d'estimation pour Nav13	139

8.11	Erreur d'estimation sur la température	140
8.12	Erreur d'estimation de Nav01 en fonction de N	140
8.13	Erreur d'estimation de Nav13 en fonction de N	141
8.14	Erreur d'estimation de Thermo01 en fonction de N	141
8.15	Erreur d'estimation pour Nav01 en fonction du bruit	142
8.16	Erreur d'estimation pour Nav13 en fonction du bruit	142
8.17	Erreur d'estimation pour Nav13 en fonction de \mathcal{M}_0	143
8.18	Erreur d'estimation pour Nav13 en fonction de \mathcal{M}_0	145
8.19	Estimation de mode pour Nav13	146
8.20	Estimation de mode numérique pour Thermo01	147
8.21	Explication de l'erreur d'estimation de mode numérique.	148
8.22	Explication de l'erreur d'estimation de mode numérique.	148
8.23	Probabilité du mode numérique.	149
8.24	Graphe des marquages cohérents du thermostat.	150
8.25	Estimation et diagnostic de Thermo01	152
8.26	Estimation et diagnostic de Thermo24	153
8.27	Estimation et diagnostic de Thermo24	154
9.1	Réversion de mode suite à une mise en palier	160
9.2	Réseau de Petri d'un scénario de mise en palier	161
9.3	Estimation de la mise en palier (Exhost-PIPE)	161
9.4	Estimation de l'altitude de l'avion pour le pilote 1	162
9.5	Estimation de la vitesse verticale de l'avion pour le pilote 1	163
9.6	Estimation de l'altitude de l'avion pour le pilote 2	164
9.7	Estimation de la vitesse verticale de l'avion pour le pilote 2	165
9.8	Estimation de l'altitude de l'avion pour le pilote 3	166
9.9	Estimation de la vitesse verticale de l'avion pour le pilote 3	167
9.10	Ghost	168
C.1	Instance Nav07 modélisée sous Exhost-PIPE	199
C.2	Information de p_9 sous Exhost-PIPE	200
C.3	Panneau de définition des paramètres numériques	201
C.4	Panneau de génération de particules selon une loi uniforme.	201
C.5	Panneau de contrôle du module d'estimation.	202
C.6	Estimation du système Nav07	204

Table des algorithmes

3.1	Évolution isoparticule	44
5.1	Rééchantillonnage	87
6.1	Evolution isoparticule	97
6.2	Pseudo-franchissable	98
6.3	Évolution isomarquage	101
6.4	Pondération	102
6.5	Rééchantillonnage	103
6.6	Classement	104
6.7	Sélection	105
B.1	Algorithme de filtrage particulière	195

Introduction

Systèmes critiques

L'étude des systèmes complexes trouve de nombreuses motivations, tout d'abord financières, car ce sont des systèmes souvent coûteux ou longs à développer. Mais la motivation essentielle est certainement celle de la sécurité, de par les conséquences graves, en termes de dommages humains, environnementaux ou matériels que peuvent avoir certains accidents liés à ces systèmes dits critiques. Ces systèmes sont d'autant plus difficiles à surveiller qu'ils sont largement autonomes, c'est-à-dire qu'ils peuvent fonctionner (au moins en partie) sans intervention humaine.

La figure 1 positionne certains systèmes en fonction de leur niveau d'interaction (communications, autonomie, nombre de modules...) et de leur "criticité". Les systèmes critiques les plus étudiés sont sans aucun doute les systèmes spatiaux (satellites, fusées, environnements de vie dans l'espace) et les systèmes de transport (trafic routier ou aérien).

Les systèmes spatiaux se différencient surtout par la grande difficulté d'agir sur le système (réparation de satellites, commande des robots d'exploration), l'éloignement (problèmes de communication) et un environnement hostile pour l'homme.

L'étude de la sécurité des systèmes de transports est un souci permanent, largement justifié par les conséquences meurtrières qui font l'actualité.

Surveillance des systèmes hybrides

Les systèmes complexes sont souvent des systèmes hybrides, c'est-à-dire des systèmes dont la dynamique est continue et discrète à la fois.

Cette dynamique discrète provient de changements de comportements du système, qui dépendent de la dynamique du système, ou qui sont dus à des événements externes en provenance de l'environnement, d'autres systèmes ou d'opérateurs humains.

Ces systèmes, qui sont donc soumis à des événements et des changements de fonctionnements, ont un comportement nominal, qui est celui qui est demandé ou commandé par un agent externe, ou le plus adapté à la situation, s'il est issu d'un processus de

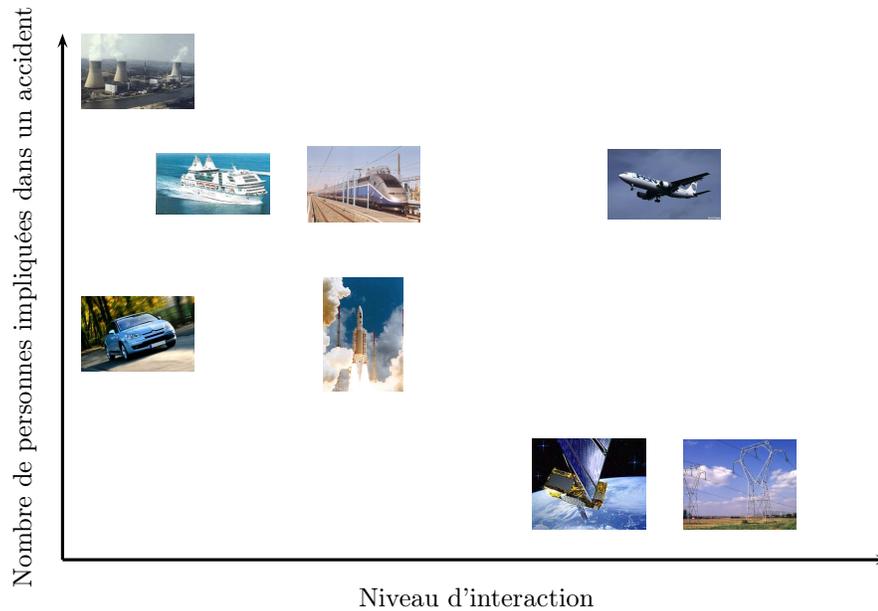


FIG. 1: Systèmes critiques (traduit de Blom et Lygeros, 2005)

planification par exemple.

Des aléas peuvent perturber ce système et modifier son comportement, qui peut devenir anormal. Ces aléas peuvent venir de pannes (perte d'un instrument de mesure), d'actions externes incorrectes (l'ordre de l'opérateur est inadapté) ou de perturbations externes (météo, environnement électromagnétique, autres agents...)

Pour pallier ces aléas, le système est **supervisé**, c'est-à-dire que son comportement est suivi, de façon à comprendre quelle est l'activité en cours, et détecter si cette activité est anormale.

Notre contribution se situe dans ce domaine particulier : proposer un modèle pour les systèmes hybrides et une méthode de supervision pour estimer l'état de ces systèmes et détecter des comportements anormaux.

Plan

Dans le **chapitre 1**, nous étudierons les techniques d'estimation de systèmes continus et de systèmes discrets. Ces techniques, propres à chaque type de systèmes, ont chacune leurs avantages et leurs inconvénients.

Nous détaillerons alors dans le **chapitre 2** les techniques d'estimation spécifiques aux systèmes hybrides. Parmi ces techniques, nous retiendrons les méthodes "mixtes", qui

couplent un modèle discret du système à un filtre numérique.

Nous proposerons alors (**chapitre 3**) un formalisme pour la modélisation des systèmes hybrides : les réseaux de Petri particuliers. Ce formalisme permettra de représenter le comportement discret (par la structure du réseau) et continu (par des équations d'évolution) du système.

Le modèle étant défini, il s'agira d'adopter un estimateur pour assurer le suivi de l'état du système. Dans le **chapitre 4**, nous détaillerons un principe d'estimation basé sur l'utilisation de filtres de Kalman. Nous verrons que ce principe est difficilement utilisable.

Notre proposition définitive (**chapitre 5**) sera axée sur l'utilisation d'une technique de type "filtrage particulière". Nous définirons les différentes étapes de ce principe d'estimation.

Le **chapitre 6** proposera une traduction algorithmique de ce principe. Les complexités théorique et expérimentale de ces différents algorithmes seront évaluées sur différents exemples et en fonction de divers paramètres.

Nous proposerons une méthode de détection d'incohérences (**chapitre 7**) à partir de l'estimation de l'état d'un système hybride, fondée sur la comparaison entre les modes de fonctionnements symboliques du système et les modes numériques, issus des particules représentant les valeurs possibles du vecteur d'état numérique.

Nous analyserons finalement le comportement de l'estimateur dans le **chapitre 8**, à travers plusieurs exemples, qui permettront d'évaluer l'efficacité de l'estimation et le principe de détection d'incohérences définis auparavant.

Le **chapitre 9** présentera une application de cette méthode au suivi de l'activité de pilotage : le système avion-équipage est un système hybride, de par la dynamique continue de l'avion et les actions provenant de l'équipage qui peuvent modifier son comportement. Nous verrons, sur quelques expérimentations, comment nous pouvons suivre l'état de ce système et détecter des conflits entre consignes de l'équipage et comportement de l'avion.

Chapitre 1

Suivi de situation dans les systèmes continus et discrets

Les systèmes automatiques de surveillance ont été conçus tout d'abord pour des problématiques d'estimation de position ou de suivi de trajectoires, avec une dynamique continue facilement identifiable et compréhensible. Puis, le développement de systèmes apportant un niveau d'abstraction adapté à l'opérateur a introduit la notion de systèmes symboliques et à événements discrets. Ce chapitre présente ces deux types de domaines, et détaille les différentes modélisations que l'on peut adopter, et les principes d'estimation qui y sont attachés. Finalement nous verrons pourquoi ces approches sont limitées pour traiter le cas de systèmes présentant à la fois des caractéristiques continues et discrètes.

1 Les systèmes continus

L'estimation d'un signal \mathbf{x} à partir d'un autre signal \mathbf{z} est un problème majeur du traitement du signal. Dans de nombreuses applications, le signal \mathbf{z} est une version bruitée de \mathbf{x} , mais il peut aussi s'agir d'un signal différent, qui contient de l'information sur \mathbf{x} . C'est le cas par exemple lorsqu'on veut reconstituer un signal de parole émis dans un microphone, bruité par l'environnement. On trouve également d'autres exemples en médecine (mesure de rythmes cardiaques) ou en poursuite de cibles (suivi d'échos radar dans le contrôle aérien).

Dans d'autres applications, l'observation du signal \mathbf{x} peut être difficile, ou partielle. Dans les applications biochimiques, il est parfois impossible de mesurer la concentration

d'un élément dans un réacteur, mais on peut essayer de déduire cette quantité à partir, par exemple, de la quantité de gaz émise lors de la réaction.

Le traitement du signal a d'abord été étudié sous sa forme analogique, avec le développement de filtres permettant de reconstruire le signal \mathbf{x} . L'agencement de composants (résistance, condensateur...) permet alors de construire différents filtres analogiques (passe-haut, passe-bas, passe-bande...)

Les traitements numériques ont par la suite permis l'essor de techniques nouvelles, permettant de synthétiser des filtres d'ordre bien supérieur aux filtres analogiques, ou même d'envisager des filtres impossible à synthétiser de manière analogique (filtre FIR¹ par exemple).

1.1 Le problème de filtrage

Le problème de filtrage consiste à déterminer des estimateurs des variables d'un système dynamique sujet à des perturbations (précision des capteurs, observations partielles...) L'évolution de l'état $\mathbf{x}(t)$ de ce système au temps t est décrite par une équation différentielle :

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), b(t)) \quad (1.1)$$

où f est une fonction déterministe caractérisant la dynamique du système et $b(t)$ est un processus de Wiener modélisant les perturbations aléatoires de la dynamique et la méconnaissance du modèle. Le processus $\mathbf{x}(t)$ est alors un processus Markovien. Pour déterminer ce processus (et donc l'état du système), on est amené à construire une équation d'observation qui relie à l'instant k l'observation bruitée $\mathbf{z}(k)$ à l'état $\mathbf{x}(k)$:

$$\mathbf{z}(k) = h(k, \mathbf{x}(k)) + v(k) \quad (1.2)$$

où h est la fonction d'observation et $v(k)$ une variable aléatoire modélisant l'imperfection des observations.

Le but du filtrage est alors de déterminer l'état $\mathbf{x}(k)$ à partir des observations $\mathbf{z}(1), \dots, \mathbf{z}(k)$. Cet estimateur $\hat{\mathbf{x}}(k)$ sera optimal s'il minimise l'erreur moyenne quadratique :

$$\mathbf{E}[\|\mathbf{x}(k) - \hat{\mathbf{x}}(k)\|^2] \quad (1.3)$$

1.2 Filtrage bayésien

Le filtrage optimal consiste à calculer la densité conditionnelle \mathbf{p}_k de l'état $\mathbf{x}(k)$ sachant les observations $\mathbf{z}(1), \dots, \mathbf{z}(k)$:

$$\mathbf{p}_k(d\mathbf{x}) = \mathbf{P}(\mathbf{x}(k) \in d\mathbf{x} | \mathbf{z}(1 : k)) \quad (1.4)$$

¹Finite Impulse Response

La suite des lois (\mathbf{p}_k) est appelée filtre optimal. En pratique on souhaite obtenir un algorithme récursif permettant de calculer \mathbf{p}_k à partir de la loi précédente \mathbf{p}_{k-1} et de la dernière observation $\mathbf{z}(k)$. Ce principe se décompose en deux étapes essentielles :

1.2.1 Prédiction

L'étape de prédiction utilise la connaissance *a priori* du système en réalisant la transition de \mathbf{p}_{k-1} à $\mathbf{p}_{k|k-1}$:

$$\mathbf{p}_{k|k-1}(d\mathbf{x}) = \int \mathbf{p}_{k-1}(d\mathbf{x}') Q_k(\mathbf{x}', d\mathbf{x}) \quad (1.5)$$

où Q_k est le noyau de transition du système associé à l'équation (1.1).

1.2.2 Recalage

Cette étape utilise l'observation $\mathbf{z}(k)$ à travers la densité conditionnelle de $\mathbf{z}(k)$ sachant $\mathbf{x}(k)$, appelée **vraisemblance** et notée $g_k(\mathbf{x})$. La formule de Bayes permet alors de corriger la densité prédite :

$$\mathbf{p}_k(d\mathbf{x}) = \frac{g_k(\mathbf{x}) \mathbf{p}_{k|k-1}(d\mathbf{x})}{\int g_k(\mathbf{x}') \mathbf{p}_{k|k-1}(d\mathbf{x}')} \quad (1.6)$$

L'application de ces deux étapes permet théoriquement d'obtenir la solution au problème du filtrage optimal. En pratique, dès que les équations du système et d'observation sont non-linéaires, les intégrales ne se calculent pas analytiquement. Il faut alors recourir à des approximations du filtre optimal.

1.3 Le filtre de Kalman

Le filtre de Kalman (1960) est certainement le filtre bayésien le plus utilisé aujourd'hui dans les systèmes de suivi d'objets, d'estimation ou de contrôle de systèmes continus.

Le filtre de Kalman (voir annexe A) repose sur une représentation gaussienne de l'état du système. Dans les cas les plus simples, l'évolution de l'état \mathbf{x} est décrite par une équation linéaire :

$$\mathbf{x}(k+1) = F_k \mathbf{x}(k) + \mathbf{u}(k) + b(k) \quad (1.7)$$

où $\mathbf{u}(k)$ est la commande appliquée sur le système au temps k et $b(k)$ le bruit du système au temps k .

De même, l'observation \mathbf{z} est décrite par une équation linéaire en l'état :

$$\mathbf{z}(k) = H_k \mathbf{x}(k) + v(k) \quad (1.8)$$

où $v(k)$ est le bruit d'observation.

Ces équations s'insèrent dans le schéma général d'un estimateur récursif, qui peut être représenté par la figure 1.1.

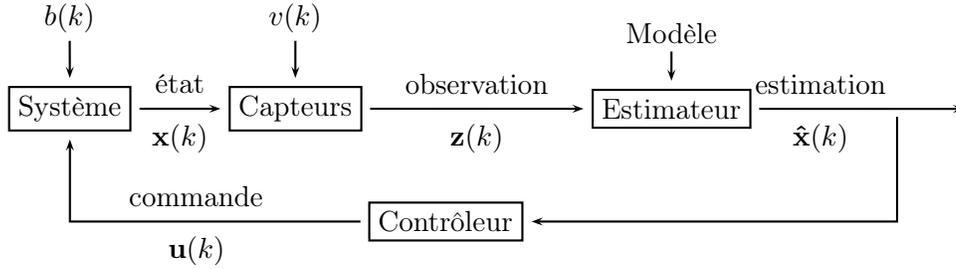


FIG. 1.1: Schéma d'une boucle estimation/contrôle (traduit de Gelb, 2001)

Dans le cadre du filtre de Kalman, les bruits considérés sont blancs gaussiens. Ils peuvent donc s'écrire

$$b(k) \sim \mathcal{N}(0, Q_k) \quad (1.9)$$

$$v(k) \sim \mathcal{N}(0, R_k) \quad (1.10)$$

On montre dans ce cas que les lois conditionnelles \mathbf{p}_k sont elles aussi gaussiennes. Ainsi, l'estimation de \mathbf{x} peut s'écrire

$$\mathbf{p}_k = \mathcal{N}(\hat{\mathbf{x}}_k, P_k) \quad (1.11)$$

Calculer l'estimée $\hat{\mathbf{x}}$ du signal \mathbf{x} revient à calculer sa moyenne $\hat{\mathbf{x}}_k$ et sa covariance P_k .

L'application du principe récursif présenté auparavant permet alors d'écrire les étapes de prédiction et de recalage :

Prédiction

$$\hat{\mathbf{x}}_{k|k-1} = F_k \hat{\mathbf{x}}_{k-1} + \mathbf{u}(k) \quad (1.12)$$

$$P_{k|k-1} = F_k P_{k-1} {}^t F_k + Q_k \quad (1.13)$$

Recalage

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}(k) - H_k \hat{\mathbf{x}}_{k|k-1}) \quad (1.14)$$

$$P_k = (I - K_k H_k) P_{k|k-1} \quad (1.15)$$

$$K_k = P_{k|k-1} {}^t H_k (H_k P_{k|k-1} {}^t H_k + R_k)^{-1} \quad (1.16)$$

Le filtre de Kalman ainsi défini est à **variance minimale** et **optimal**.

Le filtre de Kalman dispose de nombreuses extensions (Welch et Bishop, 2001), permettant de s'adapter à des cas beaucoup plus généraux que celui représenté par les équations (1.7) et (1.8). Parmi ces extensions, on pourra trouver :

- un filtre de Kalman étendu (Extended Kalman Filter), dont l’objectif est de traiter les systèmes aux équations non linéaires. Pour cela, les équations sont linéarisées autour de l’estimation courante. Ce type d’estimation est utilisé dans (Broida *et al.*, 1990) pour le suivi de trajectoire d’objets 3D depuis une séquence d’images ;
- un filtre de Kalman sans biais (Unscented Kalman Filter), qui traite les non-linéarités en utilisant un ensemble de points pour calculer la probabilité prédite en faisant évoluer ces points selon l’équation du système. Ces points sont ensuite utilisés pour reconstruire une distribution gaussienne recalée.

1.4 Le filtrage particulaire

Tout comme les autres techniques d’estimation, l’essor du filtrage particulaire a été motivé par les applications relatives à la poursuite d’objets, à l’analyse de séquences vidéo, aux télécommunications...

La spécificité du filtrage particulaire (voir annexe B), proposé pour la première fois dans (Gordon *et al.*, 1993), est d’estimer la probabilité sur l’état du système par une simulation de type Monte-Carlo.

Cette probabilité est discrétisée en un ensemble de **particules** qui explorent l’espace d’état en suivant la dynamique du système. L’estimation de l’état du système est assurée par un processus de sélection des particules “intéressantes” : une particule a d’autant plus de chances de “survivre” qu’elle est cohérente avec l’observation.

Cette méthode a de nombreux avantages sur les autres estimateurs : elle est facile à mettre en œuvre, puisqu’il suffit de pouvoir simuler le système sur un ensemble de trajectoires indépendantes. Ainsi elle permet d’envisager l’estimation de systèmes complexes, en particulier avec des dynamiques non linéaires et des bruits non gaussiens.

Le système étudié peut être représenté d’une façon générale par les équations (1.1) et (1.2).

Divers algorithmes de filtrage particulaire ont été étudiés. On pourra en trouver une description plus détaillée dans (Arulampalam *et al.*, 2002), ainsi qu’un résumé de résultats théoriques dans (Crisan et Doucet, 2002). Dans cette section, nous nous contenterons de décrire l’algorithme standard du filtrage particulaire.

Comme nous venons de le voir, l’idée du filtrage particulaire est de représenter la distribution de probabilité \mathbf{p}_k sur l’estimation $\hat{\mathbf{x}}(k)$ sachant les observations $\mathbf{z}(1), \dots, \mathbf{z}(k)$ par un ensemble de particules pondérées, notées $(\mathbf{x}_k^1, \omega_k^1, \dots, \mathbf{x}_k^N, \omega_k^N)$, avec \mathbf{x}_k^i la i -ème particule et ω_k^i son poids. Ensuite, il est nécessaire de pouvoir générer les particules à partir de la connaissance sur l’état initial du système. Puis le processus d’estimation comprend deux étapes :

1. une **prédiction**, qui calcule les particules au temps suivant $(\mathbf{x}_{k+1|k}^1, \dots, \mathbf{x}_{k+1|k}^N)$;

cette prédiction se fait en simulant l'équation d'évolution du système :

$$\forall i \in \llbracket 1, N \rrbracket, \quad \mathbf{x}_{k|k-1}^i = F(k, \mathbf{x}_{k-1}^i, b(k)) \quad (1.17)$$

2. une mise à jour, ou recalage, qui comporte deux étapes :

(a) une **pondération**, qui calcule, à partir de l'observation, le poids de chaque particule prédite :

$$\omega_k^i = g_k(\mathbf{x}_{k|k-1}^i) \quad (1.18)$$

où g_k est la fonction de vraisemblance entre prédiction et observation ;

(b) un **rééchantillonnage**, qui sélectionne (ou génère) N particules :

$$\forall i \in \llbracket 1, N \rrbracket, \quad \mathbf{x}_k^i \sim \sum_{j=1}^N \omega_k^j \delta_{\mathbf{x}_{k|k-1}^j} \quad (1.19)$$

où δ est la fonction de Kronecker.

L'algorithme de filtrage particulaire ainsi défini **converge asymptotiquement**, c'est-à-dire que l'ensemble des particules pondérées s'approche de l'état du système quand N tend vers l'infini :

$$\lim_{N \rightarrow \infty} \sum_{i=1}^N \omega_k^i \delta_{\mathbf{x}_k^i} = \mathbf{p}_k(\mathbf{x}) \quad (1.20)$$

Il existe de nombreuses variantes de cet algorithme. Ces variantes portent sur la prise en compte d'équations continues (Ng *et al.*, 2005), la variation du nombre de particules (Soto, 2005), ou encore des méthodes permettant d'appréhender des problèmes de grande taille.

Dans ce dernier cas, la **Rao-Blackwellisation** (Casella et Robert, 1996) est largement utilisée. Cette méthode repose sur le fait qu'à chaque particule est attaché un filtre de Kalman, qui estime la valeur de cette particule sur une partie linéaire de l'évolution du système. Les particules sont alors distribuées de sorte à prendre en compte les parties non linéaires de l'espace d'état.

Ainsi, on tire avantage d'un algorithme de filtrage particulaire pour traiter la non linéarité, mais chaque particule est détaillée en une gaussienne, et contient plus d'information sur la distribution de l'état du système.

1.5 Conclusions sur l'estimation continue

L'estimation des systèmes continus peut être assurée, comme nous l'avons vu, par des techniques de filtrage numérique, parmi lesquelles :

- le filtre de Kalman, optimal pour des systèmes linéaires à bruits gaussiens auquel sont associées des techniques permettant d'approcher des systèmes non linéaires ;

- le filtrage particulière, facile à utiliser de par sa nature échantillonnée, qui peut prendre en compte des systèmes aux dynamiques et aux bruits complexes, mais qui ne réalise qu’une approximation de l’état du système.

Le diagnostic d’un système continu est essentiellement basé sur l’utilisation d’un observateur, ou d’un filtre, comme le filtre de Kalman par exemple. Dans ce cas, l’erreur de l’estimation est utilisée comme un **résidu** pour la détection de fautes (Frank, 1990). Cette erreur est

$$e = \mathbf{z}(k) - H_k \hat{\mathbf{x}}_{k|k-1}$$

Cette méthode de diagnostic présente des limitations : d’une part, il peut arriver qu’une anomalie soit détectée alors que tout est normal. En effet, la mesure peut être erronée à cause des différents bruits affectant le processus d’estimation. D’autre part, la détection à partir de mesures d’erreur entraîne des effets de seuil : une variation infime de la valeur de l’observation ou de l’estimation peut faire basculer le diagnostic d’un état normal à un état anormal.

Les méthodes de *résidus* peuvent utiliser plusieurs estimateurs pour évaluer différentes hypothèses (pas de faute, erreur sur le premier capteur, ...) L’utilisation de plusieurs estimateurs entre alors dans un cadre multimodèles, que nous discuterons dans le chapitre suivant.

2 Les systèmes à événements discrets

Nous venons de voir diverses techniques adaptées au suivi de systèmes continus. Ces techniques, bien qu’efficaces, nécessitent cependant de représenter la dynamique fine du système, en terme de vecteur d’état et d’équation d’évolution.

Cependant, de nombreuses applications deviennent vite complexes lorsque l’on veut représenter leur fonctionnement “interne”. Cette constatation a mené au développement de méthodes de suivi “abstraites” : le système n’est plus représenté par son vecteur d’état, mais par une représentation symbolique de son état. Ainsi, on ne surveillera plus le fait que la température à l’intérieur du réacteur est de x degrés, ou que le véhicule roule à telle vitesse, mais on se limitera à une abstraction du type “la température du réacteur est suffisamment chaude”, ou “la vitesse du véhicule n’est pas excessive”.

Parmi les systèmes symboliques, on retrouve également les systèmes dont le comportement est conditionné par l’accomplissement d’actions ou la réception d’événements. Ces systèmes sont regroupés sous le terme de systèmes à événements discrets.

2.1 Systèmes à événements discrets et logique

Lorsque l'on considère les systèmes de surveillance à base de logique, deux types de modèles apparaissent : les logiques valuées, dont le représentant principal est la logique floue (ou logique possibiliste), et les logiques non valuées, qui peuvent soit représenter l'ensemble des états possibles, soit fournir une relation de préférence sur ces états.

2.1.1 La logique pour la reconnaissance de scénarios

Les approches à base de logique fournissent un cadre formel à la représentation d'une situation. Ainsi il est possible de représenter un état comme une formule logique, et un scénario comme une suite ou une conjonction de propositions.

La mise à jour de l'état estimé d'un système, appelé croyance, peut alors être abordée par les notions de *révision et mise à jour* : on parle de **révision** lorsque le monde est statique, alors que le terme **mise à jour** correspond à un monde dynamique.

L'opérateur servant à accomplir cette modification de la croyance courante doit vérifier un certain nombre de postulats pour assurer une révision correcte (postulats AGM (Alchourrón *et al.*, 1985)) ou une mise à jour correcte (postulats de Katsuno et Mandelzon (1991)) selon les cas.

L'étude de ces processus, et leur comparaison avec le filtre de Kalman, ont mené aux conclusions suivantes (Cossart et Tessier, 1999) : révision et mise à jour ont pour principal défaut l'absence de représentation de l'incertitude. Elles disposent de peu de modèles pour représenter la dynamique du système (pour la mise à jour), et l'observation n'est prise en compte que si elle ne remet pas en cause le modèle et la croyance courante.

Dans un cadre différent, une logique du premier ordre, le calcul de situation, est utilisée dans (Demolombe et Hamon, 2002) pour représenter une procédure, c'est-à-dire une suite d'activités que doit accomplir un pilote d'avion. Cette logique permet de représenter l'enchaînement des actions à effectuer ou interdites (sortir/rentre les volets, le train...) Si cette méthode permet de reconnaître une procédure à partir des événements reçus, elle ne prend pas en compte l'incertitude sur ces événements.

D'autres travaux associent une valeur de probabilité à la formule logique représentant un scénario reconnu. Shanahan (2002) représente l'ensemble des observations par une formule logique, et cherche une explication rationnelle qui implique cette observation. Ce raisonnement permet de représenter la reconnaissance d'objets observés par les différents capteurs d'un robot. Chaque hypothèse (explication possible) est évaluée par une mesure de probabilité, qui permet alors de fournir une information sur l'environnement observé. Ces probabilités sont mises à jour selon un principe récursif que Shanahan appelle "feedback and expectation".

Finalement, les raisonnements sur des croyances ou plus généralement des formules logiques apportent un cadre formel à la représentation d'un système. Malheureusement, ces techniques souffrent d'un manque d'adaptation, notamment lorsqu'il s'agit de considérer une dynamique complexe, la présence de bruits, et un grand nombre de scénarios.

2.1.2 La logique possibiliste

Le fondement de la logique possibiliste (Zadeh, 1965) repose sur la considération d'informations imparfaites et qualitatives. La particularité de cette logique est d'associer à chaque proposition un degré de possibilité. Ainsi à une variable a est associé un ensemble flou A défini par sa fonction d'appartenance μ_A .

Par exemple une fonction d'appartenance au terme "grande vitesse en ville" peut-être celle de la figure 1.2.

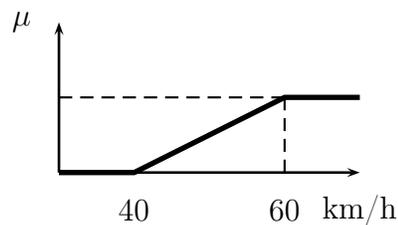


FIG. 1.2: Fonction d'appartenance de "grande vitesse en ville" : la vitesse 40 appartient à A avec un degré $\mu_A(40) = 0$; la vitesse 50 a un degré d'appartenance $\mu_A(50) = 0,5$.

Cette valeur d'appartenance, comprise entre 0 et 1, permet ainsi d'évaluer à quel point une proposition est possible et permet de représenter des informations imparfaites.

La logique floue est de plus en plus utilisée pour les systèmes de diagnostic où l'on veut une réponse symbolique non probabiliste. C'est le cas dans de nombreux systèmes experts, où l'appréciation humaine, utilisée comme source de données, est souvent "floue", tout comme dans les domaines où la quantification de l'incertitude est difficile (économie, sciences naturelles et humaines).

Dans le cadre de l'estimation, la relation entre logique possibiliste et probabilités a évidemment conduit à des recherches reprenant le cadre de l'estimation bayésienne. Ainsi Benferhat, Dubois, et Prade (2000) ont proposé un estimateur reprenant le fonctionnement du filtre de Kalman dans un cadre possibiliste : l'estimation de l'état du système, tout comme la représentation des "bruits", est donnée par une distribution de possibilité. La phase de recalage met à jour ces possibilités à partir d'une observation floue.

Parmi les applications de la logique possibiliste, on trouve également les technologies de l'Internet, notamment les techniques de recherche d'information. Hunter et Liu (2005) proposent ainsi d'utiliser la logique possibiliste pour évaluer la qualité d'un document

XML d'après des mesures de nécessité sur ses éléments. Ces mesures sont utilisées pour calculer un degré de cohérence du document, ce qui permet de déterminer les “meilleurs” documents.

Ces méthodes, même si elles permettent d'enrichir la représentation de l'incertitude (par rapport à une formule logique classique), souffrent cependant :

- de l'absence d'un modèle riche représentant la dynamique du système étudié (ce qui est le cas pour la logique de manière générale) ;
- de la difficulté liée à l'évaluation des valeurs de possibilité (ce qui est également le cas dans un cadre probabiliste).

2.1.3 Représentation des préférences

Les approches logiques classiques fournissent une croyance ou un ensemble d'états possibles du système. À l'opposé, la logique floue fournit une distribution de possibilité sur ces états.

D'autres méthodes cherchent un compromis, permettant de donner une préférence sur les états possibles, sans pour autant fournir une distribution de valeurs. C'est le cas par exemple pour la mise à jour de relations de préférences, où l'incertitude sur les croyances est représentée par une relation d'ordre. Diverses approches sont détaillées dans (Benferhat *et al.*, 2004) sur l'utilisation de préordres, puis dans (Benferhat *et al.*, 2005) pour les préordres partiels.

Ces méthodes souffrent également d'un manque de modèle d'évolution. De plus, l'utilisation de relations de préférence peut parfois limiter la pertinence des résultats, par exemple dans des cas où tout devient également possible (ou incomparable) (Dubois *et al.*, 2002).

2.2 Modélisation du comportement des systèmes à événements discrets

Ce paragraphe présente d'autres méthodes de représentation et d'estimation des systèmes à événements discrets. Ces méthodes tirent leur avantage dans la modélisation du système et de son évolution, ce qui constitue le défaut principal des techniques à base de logique.

2.2.1 Graphes et arbres

Surveiller un système c'est reconnaître l'activité en cours et être capable de diagnostiquer une situation défailante. Dans cette optique, certains travaux proposent de modéliser directement ces activités (normales ou anormales), sous la forme de scénarios. Les travaux de Rota et Thonnat (2000) sont basés sur la reconnaissance de modèles. Un scénario est représenté par un arbre dont les feuilles permettent l'identification d'événements à observer.

Ces événements sont ensuite représentés par un ensemble de propriétés qui vont permettre la reconnaissance d'un scénario. Si une séquence d'observations instancie complètement un modèle, celui-ci est reconnu. Ce principe est appliqué à la surveillance vidéo (figure 1.3).

$$(x_0 : +, x_1 : +, x_2 : -)$$

$$\left\{ \begin{array}{l} c_1 \quad \text{category}(x_0) = \text{pedestrian} \\ c_2 \quad \text{name}(x_1) = \text{Object} - \text{Appears} \\ c_3 \quad \text{time}(x_0) = \text{time}(x_1) \\ c_4 \quad \text{distance}(\text{hull}(x_0), \text{hull}(x_1)) < 300 \\ c_5 \quad \text{name}(x_2) = \text{Object} - \text{Disappears} \\ c_6 \quad \text{name}(\text{ref}(x_1, 1)) = \text{name}(\text{ref}(x_2, 1)) \\ c_7 \quad \text{time}(x_1) - \text{time}(x_2) > 0 \end{array} \right.$$

FIG. 1.3: Modèle de *Object – Deposit* (Rota, 2001) : ce modèle est basé sur trois événements (x_0 , x_1 , x_2) dont un qui est exclu ($x_2 : -$). Si l'événement x_2 est reconnu, le scénario n'est pas reconnu. La condition c_1 indique qu'un piéton doit être en x_0 , c_2 que x_1 correspond à la reconnaissance d'un autre scénario (Object-Appears), c_3 que les événements x_0 et x_1 sont simultanés, et c_4 que le piéton et l'objet doivent être proches. Puis les conditions restantes indiquent que l'on veut éviter, au même instant, la perte de l'objet en question.

Le système CATS (Callantine, 2002) est consacré à l'analyse de l'activité de pilotage. Pour cela, les procédures ou règles sont modélisées sous la forme d'arbres et/ou qui représentent l'ensemble des tâches que doit effectuer l'équipage. L'observation des tâches effectuées au cours d'un vol permet alors de progresser dans ces arbres, et de reconnaître l'activité en cours, et éventuellement de détecter une erreur, par exemple l'omission d'une tâche.

On peut reprocher à ces méthodes l'utilisation d'arbres, qui sont des structures peu expressives, et le manque de prise en compte de l'incertitude.

2.2.2 Graphes de contraintes

Pour enrichir les méthodes de suivi à base de graphes ou d'arbres, certains travaux prennent en compte dans le modèle des contraintes, temporelles ou de ressources, entre différentes activités ou différents événements du système.

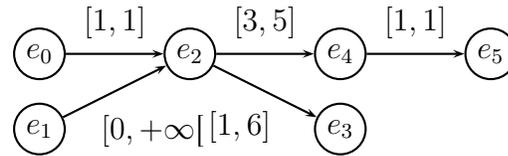
Parmi ces méthodes, on peut distinguer la modélisation sous forme de chroniques (Dousson *et al.*, 1993). Une chronique est un ensemble d'événements. Ces événements sont dépendants les uns des autres à travers un ensemble de contraintes temporelles (figure 1.4).

```

chronicle LoadMachine()() {
  timepoint e0, e1, e2, e3, e4, e5;
  event (Robot : (OutRoom, InRoom), e1);
  event (Robot : (InRoom, OutRoom), e4);
  event (Machine : (UnLoaded, Loaded), e2);
  event (Machine : (Stopped, Running), e3);
  hold (Machine : Stopped, (e0, e2));
  hold (Alarm : None, (e2, e5));
  e1 < e2;
  (e3 - e2) in [1.00, 6.00];
  (e4 - e2) in [3.00, 5.00];
  (e5 - e4) = 1.00;
  (e2 - e0) = 1.00;
}

```

(a) Chronique



(b) Graphe de contraintes

FIG. 1.4: Chronique et graphe de contraintes du chargement d'une machine (Dousson, 1994)

Le système de surveillance dispose d'un ensemble de chroniques représentant les différents scénarios possibles dans le système. L'observation de chaque événement permet alors, en mettant à jour le graphe de contraintes, de conclure à la reconnaissance de certains scénarios. Ce principe est utilisé pour la surveillance d'un réseau de télécommunications (Dousson et Duong, 1999). Dans ces travaux, une approche pour apprendre un ensemble de chroniques depuis un ensemble d'événements est également proposée.

Vu, Brémond, et Thonnat (2003) proposent d'étendre la représentation adoptée dans (Rota et Thonnat, 2000) en y ajoutant la gestion des contraintes temporelles entre les différentes propriétés d'un modèle. Cette amélioration a mené à la décomposition d'un scénario en scénarios élémentaires, et à l'utilisation de techniques inspirées de la reconnaissance de chroniques. Cette méthode est appliquée à la reconnaissance de séquences vidéos pour la détection d'intrusions ou d'agressions (figure 1.5).

Les modèles présentés ci-dessus ont fait leur preuve pour la reconnaissance d'événements dans un réseau ou dans des séquences vidéos. On peut cependant leur reprocher de ne baser leur reconnaissance que sur l'observation d'événements, et de ne pas mettre en œuvre un principe de prédiction. Les événements considérés ne sont pas bruités, et la seule incertitude repose sur l'arrivée (la perte) de ces événements. De plus, il est impossible, avec de tels systèmes, de détecter une situation défailante si celle-ci n'est pas modélisée sous la forme de scénarios.

2.2.3 Réseaux bayésiens

Un réseau bayésien est un graphe acyclique dans lequel chaque nœud représente une variable. Un arc entre deux nœuds représente l'influence d'une variable sur une autre. À chaque arc est également associée une distribution de probabilité conditionnelle.

```

Scenario (Bank-attack,
  Characters ((cashier : Person), (robber : Person))
  SubScenarios (
    (cas-at-pos, inside-zone, cashier, Back-Counter)
    (rob-enters, changes-zone, robber, Entrance-zone, Infront-Counter)
    (cas-at-safe, inside-zone, cashier, Safe)
    (rob-at-safe, inside-zone, robber, Safe))
  ForbiddenSubScenarios (
    (any-in-branch, inside-zone, any-p, Branch))
  Constraints (
    Temporal (
      (rob-enters during cas-at-pos)
      (rob-enters before cas-at-safe)
      (cas-at-pos before cas-at-safe)
      (rob-enters before rob-at-safe)
      (rob-at-safe during cas-at-safe))
    Atemporal ((cashier ≠ robber))
    Forbidden (
      (any-p ≠ cashier)
      (any-p ≠ robber)
      (any-in-branch during rob-at-safe))))

```

FIG. 1.5: Scénario d’attaque de banque (Vu *et al.*, 2003) : il définit les sous-scénarios nécessaires et interdits et les contraintes qui les relient.

Le processus d’inférence, qui permet de déterminer la probabilité d’un événement (la valeur d’une variable par exemple) sous certaines conditions (une connaissance sur l’état de certaines variables par exemple), est basé sur la formule de Bayes.

L’utilisation des réseaux bayésiens dans les systèmes de surveillance est souvent mise en œuvre en adoptant une modélisation proche de celle de l’estimation bayésienne (paragraphe 1.2). Les réseaux bayésiens dynamiques (Murphy, 2002) permettent ainsi de représenter l’évolution des variables d’un système en représentant l’influence entre leurs valeurs au temps $k - 1$ et leurs valeurs au temps k . On peut retrouver les notions de modèle de transition (ou d’évolution) et de modèle d’observation. Par exemple la figure 1.6 représente le réseau bayésien associé à un processus d’estimation : il s’agit d’estimer s’il pleut à partir de l’observation du port de parapluie.

Un système de surveillance de trafic routier (Gong et Buxton, 1992) utilise le formalisme des réseaux bayésiens. Chaque réseau représente le comportement d’un véhicule et l’influence des différentes variables (position, vitesse, orientation, proximité avec un autre véhicule ou avec le trottoir...). Il est ainsi possible d’estimer la configuration du trafic et de reconnaître des situations d’embouteillage ou d’accidents.

Intille (1999) a également proposé un principe de reconnaissance de situation à partir de réseaux bayésiens. Ce principe est appliqué à la reconnaissance de phases de jeu à partir de vidéos de football américain. Une première étape permet de déterminer le rôle de chaque joueur : la position de ce joueur en début de partie permet d’instancier un réseau bayésien qui fournit une distribution de probabilité sur les postes possibles (quart-arrière, receveur, ailier...). D’autres réseaux bayésiens représentent différentes phases de

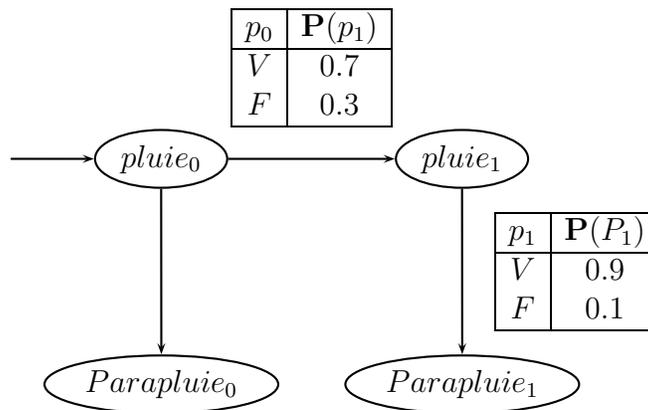


FIG. 1.6: Réseau bayésien d'estimation du temps (traduit de Russell et Norvig, 2002) : les variables *pluie* et *Parapluie* peuvent prendre les valeurs "vrai" (V) ou "faux" (F). La fonction d'observation est définie par le tableau de probabilité reliant les variables *pluie* aux variables *Parapluie* : par exemple, lorsqu'il pleut, la probabilité de voir un parapluie est de 0.9. La fonction de transition est définie par le tableau de probabilité reliant la variable $pluie_0$ à $pluie_1$: s'il pleut à l'instant 0, la probabilité qu'il pleuve à l'instant 1 est de 0.7.

jeu : les postes des joueurs et la reconnaissance d'actions élémentaires (course, passe...) permettent de déduire la probabilité de reconnaissance d'une phase de jeu donnée ou d'une stratégie.

Une forme de réseaux bayésiens dynamiques, les réseaux bayésiens récurrents, est utilisée dans (Moënné-Loccoz *et al.*, 2003) pour la reconnaissance de comportements dans des séquences vidéo. Le réseau bayésien est agencé en modules qui représentent l'évolution de différents paramètres (mouvement de l'objet, orientation, grossissement...) puis ces modules influent sur l'état global de la prochaine image. Cet état, représenté par une variable d'agitation, permet de détecter des scènes violentes dans une vidéo de surveillance d'une station de métro.

À partir d'un réseau bayésien modélisant le fonctionnement d'une usine de production (et de ses capteurs), Nielsen et Jensen (2005) définissent une mesure de conflit basée sur l'agrégation de certaines probabilités. Ainsi, il est possible, en plus d'estimer l'état du système, de diagnostiquer une situation conflictuelle lorsque cette mesure de conflit dépasse un certain seuil.

Les réseaux bayésiens permettent de représenter graphiquement la dépendance entre variables du système et de mettre en œuvre un principe d'inférence bayésienne. Cependant, on peut leur reprocher :

1. Une lecture complexe : il peut être assez difficile de comprendre le fonctionnement d'un système à partir du modèle de réseau bayésien qui en a été fait. Ce modèle représente les dépendances entre variables, et la compréhension de la dynamique du système nécessite d'appréhender les probabilités conditionnelles qui guident cette dynamique ;
2. Une représentation probabiliste (possibiliste dans certaines variantes), qui nécessite la définition de valeurs représentant l'incertitude. Il est donc impossible de modéliser une ignorance totale sur le système, comme le fait que d'un état a on puisse se retrouver dans un état b ou c , sans savoir lequel est le plus "probable", ou sans pouvoir quantifier le fait que b soit plus "vraisemblable" que c .

Les formalismes basés sur les automates permettent de répondre en partie aux limites posées par les réseaux Bayésiens.

2.2.4 Automates

Définition 1.1 (Automate)

Un **automate** est un quadruplet $A = (M, \Sigma, \phi, M_0)$ avec

- M un ensemble fini d'états, ou **modes**,
- Σ un alphabet,
- $M_0 \subset M$ l'ensemble des états initiaux de l'automate,
- $\phi : M \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(M)$ la fonction de transition de l'automate.

Le formalisme des automates est largement utilisé pour représenter le fonctionnement d'un système. Chaque état $\mu \in M$ de l'automate représente un mode de fonctionnement du système. La fonction de transition ϕ , qui représente les relations liant ces différents modes, est dirigée par un alphabet Σ qui est, dans la plupart des applications, représentatif des événements ou actions qui peuvent influencer le système.

Ce formalisme a pour avantages d'être assorti de nombreuses méthodes d'analyse ou de résolution (déterminisation, langage reconnu, états accessibles...) et de pouvoir être représenté graphiquement. Ainsi la figure 1.7 représente l'automate d'un thermostat, qui est géré par quatre événements : **on**, **off**, respectivement pour la mise en route et l'arrêt, **hot** et **cold** qui indiquent respectivement que la température de la pièce est trop chaude ou trop froide.

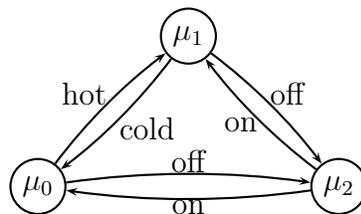


FIG. 1.7: Automate d'un thermostat

Dans cet automate, μ_2 correspond au mode “arrêt” du thermostat, μ_0 correspond à l’état “marche” et “chauffe” alors que μ_1 correspond uniquement à un état de “veille” : le thermostat est allumé mais la pièce n’est pas chauffée.

Ce formalisme est largement utilisé pour représenter le fonctionnement discret d’un système pour le suivi d’activités ou la reconnaissance de scénarios.

Dans (André *et al.*, 1988), des automates sont utilisés pour représenter des scénarios de jeu pour le commentaire de matchs de football. Certains automates représentent des phases de course, de passe, de tir... A partir des événements reçus d’un traitement des séquences vidéo, les automates sont joués, et lorsqu’un automate arrive dans un état final, la phase de jeu est reconnue. Cette reconnaissance permet alors de générer un commentaire lié à cette phase de jeu.

Largouët et Cordier (2000) proposent un principe d’estimation récursive pour déterminer la nature d’une zone observée par satellite, comme par exemple le type de culture, la présence d’une forêt... Un automate temporel représente les différents états possibles du terrain et leur évolution (figure 1.8).

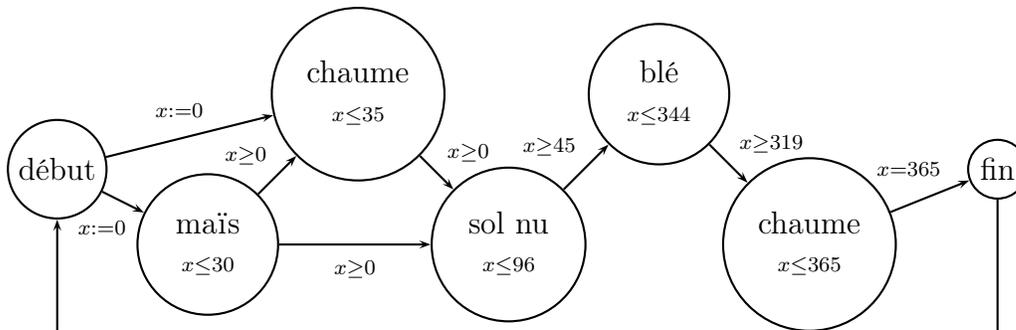


FIG. 1.8: Modèle partiel représentant l’évolution d’une parcelle (Largouët et Cordier, 2000). La variable x représente le temps. Les inégalités représentent les contraintes temporelles : par exemple, le maïs ne peut pas être cultivé plus de 30 jours ($x \leq 30$) et il faut au moins 45 jours avant de cultiver du blé ($x \geq 45$). Au début d’un cycle de culture, la variable x est initialisée ($x := 0$).

Cette modélisation est utilisée pour prédire, à partir de la connaissance courante, les états possibles de la zone lors de la prochaine observation. Puis cette prédiction est mise à jour (recalée) à partir de la nouvelle observation. Ce principe d’estimation permet alors d’affiner la reconnaissance des types de terrain : la connaissance apportée par la phase de prédiction permet de lever d’éventuelles ambiguïtés quant à l’observation de cette zone (couleurs similaires pour deux cultures différentes par exemple).

Dans (Williams *et al.*, 2003), le fonctionnement d’un module spatial est modélisé par un ensemble d’automates. Chaque automate représente le fonctionnement d’un système (moteurs, caméra...) Les modèles des systèmes comportent des transitions probabilistes :

la transition d'un mode à un autre se fait selon une certaine probabilité. L'automate ainsi défini est une chaîne de Markov. Il est alors possible de calculer la probabilité de défaillance d'un système en fonction de son état courant estimé. Williams *et al.* (2003) proposent un langage de représentation et d'interprétation de ces systèmes.

Grastien, Cordier, et Largouët (2004) proposent un principe de diagnostic décentralisé, basé sur la modélisation du système par différents automates pour chaque module, et la modélisation de l'incertitude sur l'arrivée des événements par un automate d'observation. La synchronisation de ces modèles permet alors d'estimer l'état courant du système, et de savoir si un état défaillant est possible.

En conclusion, les automates fournissent un formalisme permettant de représenter la dynamique d'un système par les transitions entre modes. Ces transitions peuvent être conditionnelles, temporelles, stochastiques... et représenter un grand nombre de comportements. Cependant, chaque mode du système doit être représenté. De plus, le diagnostic d'une situation anormale nécessitant la modélisation du comportement anormal, il est nécessaire de modéliser explicitement ces comportements. Le modèle créé peut devenir très complexe.

2.2.5 Réseaux de Petri

Proposés pour la première fois par Petri (1962) dans le but de représenter des processus de communication en utilisant des modèles de type automate, les réseaux de Petri prennent une place de plus en plus importante dans le monde scientifique, de par les nombreuses extensions qui en existent, et leur application à divers domaines : supervision, contrôle, vérification, sûreté de fonctionnement...

Définition 1.2 (Réseau de Petri)

Un **réseau de Petri** est un quadruplet $\langle P, T, Pre, Post \rangle$, où

- P est un ensemble fini de places,
- T est un ensemble fini de transitions,
- $Pre : P \times T \rightarrow \mathbb{N}$ est l'application places précédentes,
- $Post : P \times T \rightarrow \mathbb{N}$ est l'application places suivantes.

Un réseau de Petri est un graphe composé de deux types de nœuds : des places et des transitions. Les arcs de ce graphe peuvent relier des places à des transitions (représentés par l'application Pre) ou des transitions à des places (application $Post$).

Le marquage $\mathcal{M}(p)$ d'une place p est le nombre de marques (ou **jetons**) contenus dans la place p . L'évolution de ce marquage se fait selon des règles de franchissement :

- une transition t est **franchissable** si et seulement si

$$\forall p \in P, \quad \mathcal{M}(p) \geq Pre(p, t) \quad (1.21)$$

– le **tir** d'une transition franchissable t crée un marquage \mathcal{M}' défini par :

$$\forall p \in P, \quad \mathcal{M}'(p) = \mathcal{M}(p) - Pre(p, t) + Post(p, t) \quad (1.22)$$

Un arc (valué) reliant une place p à une transition t représente ainsi le nombre de jetons de p consommés par le tir de la transition t (nécessaires à son franchissement). Un arc reliant une transition t à une place p représente le nombre de jetons créés dans p par le tir de t .

La figure 1.9(a) représente un réseau de Petri marqué et la figure 1.9(b) le même réseau après le tir de la transition t_3 .

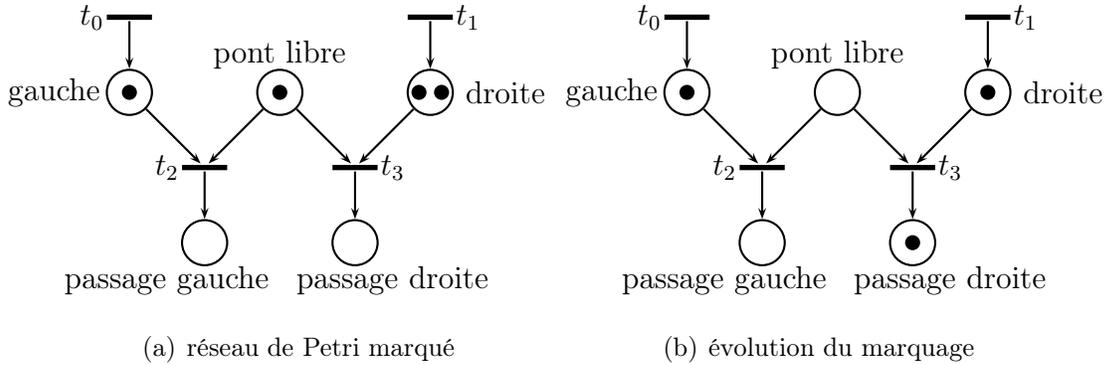


FIG. 1.9: Un réseau de Petri et le franchissement d'une transition (modèle traduit de David et Alla, 2005)

Les réseaux de Petri permettent, de par leur structure, de modéliser un grand nombre de comportements comme le parallélisme, la synchronisation, le partage de ressource.

Les nombreuses extensions des réseaux de Petri (réseaux temporels, temporisés, stochastiques, colorés, à priorité, à objets...) permettent également d'envisager l'utilisation des réseaux de Petri dans les problèmes les plus divers et les plus complexes.

On trouve diverses approches d'estimation ou de supervision utilisant des réseaux de Petri. Dans ces applications, le réseau est *interprété*, c'est-à-dire qu'il correspond au comportement d'un système. De manière générale, une place du réseau correspond alors à une activité ou un sous-état du système. Une transition correspond à l'arrivée d'un événement permettant son franchissement et de faire évoluer le marquage du réseau de façon conjointe avec l'évolution du système.

D'autres travaux portent sur l'estimation du marquage d'un réseau de Petri, c'est-à-dire l'estimation de l'état du système à partir de l'observation de certains événements.

Ces travaux se distinguent des précédents par la prise en compte de l'incertitude, que ce soit sur l'observabilité ou sur l'arrivée d'événements.

Dans (Fanni *et al.*, 1997), l'incertitude ne provient pas des événements, mais du marquage initial : un macromarquage est alors défini pour représenter un ensemble d'états possibles. L'observation du tir des transitions permet d'affiner la connaissance sur le marquage du réseau. La définition de places d'observation, qui représentent des macro-marquages, permet de mettre en œuvre un contrôleur qui assurera que le système n'atteindra pas d'état redouté.

Lorsque l'arrivée de certains événements n'est pas observable, Corona *et al.* (2004) proposent un algorithme de construction d'un automate fini déterministe. Chaque état de cet automate représente un ensemble des marquages. Ce principe permet, à partir d'une séquence d'événements observés, d'estimer l'ensemble des marquages possibles à un instant donné.

Bourjij et Koenig (1999) utilisent la forme matricielle d'un réseau de Petri ainsi que l'équation fondamentale (1.22) pour estimer le marquage de ce réseau. Cette représentation permet d'utiliser des observateurs pour mettre à jour le marquage à partir des observations.

Le formalisme des réseaux de Petri flous permet de combiner la modélisation du comportement d'un système par réseaux de Petri et la prise en compte de l'incertitude par des valeurs de possibilité. Différentes façons de rendre un réseau flou sont envisageables (voir Cardoso *et al.*, 1996).

L'utilisation des ensembles flous peut par exemple servir à représenter l'incertitude sur la date d'arrivée d'un événement (par opposition au réseaux de Petri temporels où l'on dispose d'un intervalle de tir). À partir de ce modèle, Kurkovsky et Loganantharaj (2005) proposent un moyen de déterminer une distribution de possibilité sur le marquage du réseau, et donc sur l'état du système.

Les réseaux de Petri possibilistes (Cardoso *et al.*, 1999) permettent l'association d'un marquage flou à un réseau de Petri : l'appartenance d'un jeton à une place est définie par un degré de possibilité. Ce marquage évolue dans le réseau en propageant l'incertitude, modélisée par un degré de possibilité sur le franchissement de certaines transitions. Ces règles de pseudo-tirage permettent alors de calculer une distribution de possibilité sur l'état du système. Puis l'observation du franchissement d'une transition (de l'arrivée d'un événement) est utilisée pour mettre à jour le réseau : le tir de la transition en question, qui était possible dans le calcul précédent, est maintenant certain. Cette considération permet de mettre à jour la distribution de possibilité sur l'état du système.

D'une façon similaire, les travaux de (Rombaut *et al.*, 1999) utilisent la théorie de l'évidence (Smets et Kennes, 1996) pour représenter l'incertitude sur le marquage d'un réseau de Petri. On y retrouve l'association d'un degré de franchissement aux transitions et l'utilisation d'un principe d'estimation bayésienne pour mettre à jour la distribution de croyances.

Cossart (1999) et Tessier (2003) proposent un principe de suivi d'activité également basé sur les réseaux de Petri. Contrairement aux approches précédentes, le principe d'estimation mis en œuvre est fondé sur la mise à jour de croyances pour maintenir l'information sur l'état du système. L'estimation de l'état du système à un instant donné est alors définie par une relation de préférence sur des marquages. Ce principe d'estimation se base sur l'utilisation des réseaux de Petri pour la prédiction des états futurs, puis sur une comparaison entre observation et marquages prédits pour mettre à jour la relation de préférence. Ce principe est appliqué à la surveillance vidéo d'un parking (figure 1.10).

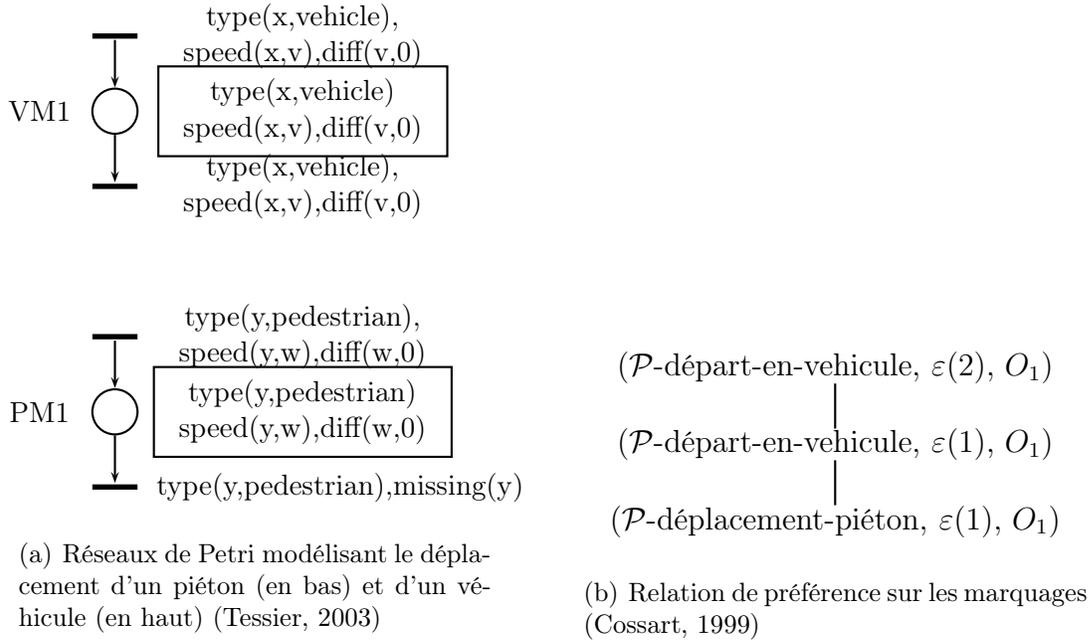


FIG. 1.10: Principe d'estimation du marquage d'un réseau de Petri mis en œuvre pour la surveillance de parking. Aux transitions et places du réseau de Petri sont associés des prédicats représentant l'activité en cours (présence d'un véhicule x , avec une vitesse v non nulle par exemple). L'estimation consiste en une relation de préférence dont chaque élément est un triplet contenant le plan en cours \mathcal{P} , le marquage estimé ε et l'objet O_1 effectuant l'activité.

Le formalisme des réseaux de Petri est de plus en plus utilisé pour superviser le fonctionnement d'un ou de plusieurs systèmes. Les premières méthodes envisagées ne fournissent qu'un ensemble de marquages possibles à partir de l'observation d'événements. L'information fournie est alors "pauvre", et il peut être difficile de prendre une décision (diagnostic de panne par exemple) lorsque l'on ne dispose pas de plus d'information.

À l'opposé, nous avons vu des méthodes où l'estimation consiste en une distribution de valeurs (possibilités, croyances) sur l'ensemble des marquages. Cette représentation est riche et facilite la prise de décision, car on sait quels sont les états les plus vraisemblables à

un instant donné. Cependant, il peut parfois s'avérer difficile de fournir un degré de franchissement qui soit significatif de l'arrivée de certains événements. De plus, l'observation en elle-même n'est pas bruitée.

Les derniers travaux (Cossart, 1999; Tessier, 2003) considérés constituent en quelque sorte un compromis entre ces deux approches : on ne dispose pas d'information sur le degré de possibilité de franchissement d'une transition, mais il est cependant possible d'avoir une estimation "informée" définie comme une relation de préférence entre les marquages possibles.

3 Conclusion

3.1 Synthèse

Le tableau 1.1 constitue une synthèse des méthodes passées en revue dans ce chapitre, en présentant une description succincte de chaque méthode (modèle dynamique, incertitude considérée, principe de reconnaissance), ainsi que leurs avantages et inconvénients.

Méthode	Dynamique	Incertitude	Reconnaissance	Avantages	Inconvénients
Filtre de Kalman	Équation différentielle	Bruit gaussien	Estimation bayésienne	Cas linéaire optimal	Information numérique
Filtrage particulaire		Bruit stochastique		Discrétisation	
Mise à jour de croyances	-	Croyance ensembliste	Opérateur de mise à jour	Représentation formelle	Pas de modèle dynamique
Logique		-	Satisfaction de formule	Raisonnement symbolique	
Logique floue		Mesure de possibilité	Opérateur de fusion	Représentation floue	
Graphes	Succession d'actions	-	Instanciation de modèle	Intégration de contraintes	Dynamique pauvre
Réseaux bayésiens	Probabilités conditionnelles	Bruit stochastique	Estimation bayésienne	Modèle graphique	Dynamique peu compréhensible
Automates	Modèle état/transition	-	État final atteint	Comportement riche	Pas d'incertitude
Réseaux de Petri et observabilité	Jeu du réseau	Macromarquage	Observateur/automate	Gestion de l'incertitude	Principalement basé sur événements
Réseaux de Petri possibilistes	Pseudo-tir	Distribution possibiliste	Mise à jour du marquage	Gestion de l'incertitude en prédiction	Observation non bruitée
Réseaux de Petri et estimation symbolique	Jeu du réseau	Relation de préférence	Estimation récursive	Gestion de l'incertitude	Peu d'information si marquages équivalents

TAB. 1.1: Synthèse des méthodes de suivi de situation

3.2 Estimation numérique/symbolique

Les filtres numériques permettent d'estimer de façon efficace l'état continu d'un système.

Automates et réseaux de Petri fournissent une représentation des modes de fonctionnement et des transitions entre ces modes, ainsi qu'une panoplie d'extensions permettant de modéliser des systèmes stochastiques, contraints, distribués, contrôlés. . . Si ces modèles sont peu utilisés pour l'estimation, ils proposent en revanche des méthodes de diagnostic basées sur le comportement du système, et non sur une simple mesure de l'erreur entre prédiction et observation.

De cette réflexion sont issues de nombreuses techniques : utilisation de plusieurs filtres numériques pour plusieurs comportements, utilisation conjointe d'un filtre numérique et d'un modèle discrets. . . Le chapitre suivant présente ces différentes techniques, qui visent à analyser, estimer et diagnostiquer un **système hybride**, c'est-à-dire un système dont le comportement est à la fois continu et discret (également appelé *discontinu* ou *numérique/symbolique*).

Chapitre 2

Systèmes hybrides et estimation

Un système est **hybride** lorsque sa dynamique est continue et discrète. Si la modélisation est certes plus complexe, le résultat attendu est plus riche. Les techniques d'estimation de systèmes hybrides présentées dans ce chapitre englobent les filtres multimodèles, les réseaux bayésiens hybrides et les méthodes mixtes, qui couplent un modèle hybride (automates hybrides) avec une technique d'estimation numérique (filtre de Kalman, filtrage particulière). Nous concluons ce chapitre sur la contribution proposée : le couplage d'un réseau de Petri hybride à un filtre numérique.

1 Système hybride

Un système hybride a une dynamique à la fois continue et discrète. La dynamique continue est décrite par un ensemble d'équations d'évolution. La dynamique discrète correspond à des changements de modes de fonctionnement provoqués par l'arrivée d'événements internes (conditions sur l'état du système) ou externes (actions d'un opérateur, état de l'environnement).

Pour estimer l'état d'un tel système, il est nécessaire de définir des méthodologies appropriées. En effet, l'utilisation directe de techniques spécifiques des systèmes continus ou des systèmes à événements discrets convient mal :

- les filtres numériques sont basés sur une équation d'évolution, et ne peuvent donc pas prendre en compte les changements de mode de fonctionnement du système ; nous verrons dans ce chapitre comment contourner cette lacune ;

- Les principes d’estimation des systèmes à événements discrets pourraient être appliqués à des systèmes hybrides, mais sans tenir compte de la dynamique continue ; pour enrichir ces méthodes, nous présenterons des travaux couplant des modèles discrets à des filtres numériques.

2 Filtrage multimodèle

2.1 Filtre de Kalman multimodèle

Le filtre de Kalman (chapitre 1, paragraphe 1.3) permet d’estimer l’état d’un système décrit par une équation d’évolution. Lorsque le système est hybride, il est régi par des équations différentes selon le mode de fonctionnement dans lequel il se trouve. Ainsi, pour estimer l’état d’un système hybride, on peut utiliser une batterie de filtres de Kalman qui suivent chacun un mode de fonctionnement précis.

L’estimation de l’état du système est construite à partir des différents filtres, selon des méthodes de moyenne pondérée ou de sélection de filtres. Pour le suivi d’un véhicule en déplacement, Lee *et al.* (2002) calculent la moyenne pondérée de plusieurs filtres de Kalman suivant différents comportements du système, et mettent à jour non seulement l’estimation de l’état du système, mais également les poids associés aux différents estimateurs.

Veeraraghavan et Papanikolopoulos (2004) utilisent plusieurs filtres de Kalman étendus dont les estimations sont comparées de façon à déterminer quels sont les modes les plus probables. Cette étape permet ensuite de sélectionner les modèles correspondant le mieux au comportement du système, et donc de conserver les filtres les plus adaptés. Ce principe est appliqué à la surveillance d’intersections pour la gestion du trafic routier.

Pour suivre la trajectoire de cibles à partir d’images satellites, Pannetier *et al.* (2005) proposent de modéliser les différentes trajectoires possibles par plusieurs filtres. Ainsi chaque segment d’une route correspond à un filtre de Kalman, et l’arrivée à une intersection engendre la création de plusieurs filtres correspondant aux différentes hypothèses possibles. L’estimation de la position de la cible est évaluée à partir d’une méthode IMM (Interacting Multiple Models) basée sur les filtres associés aux trajectoires possibles.

Le filtre de Kalman multimodèle est en soit peu adapté à l’estimation des systèmes hybrides, et il faut donc utiliser autant de filtres que de modes. Il existe cependant plusieurs méthodes permettant de sélectionner les filtres de façon efficace, ou d’utiliser un modèle discret pour enrichir cette sélection, qui entre dans le cadre des estimateurs hybrides (paragraphe 4.3).

2.2 Filtrage particulaire pour les systèmes hybrides

L'aspect échantillonné du filtrage particulaire apporte un avantage certain pour l'estimation des systèmes hybrides (Blom et Bloem, 2004). En effet, on dispose de plusieurs particules qui peuvent évoluer de façon indépendante, contrairement à une distribution gaussienne (filtre de Kalman) qui ne peut suivre qu'une seule dynamique.

L'estimation de l'état d'un système hybride se fait donc en associant à chaque particule π^i à la fois un mode de fonctionnement et un vecteur d'état numérique : $\pi^i = (\mathbf{x}^i, \boldsymbol{\mu}^i)$.

L'étape de prédiction consiste en l'évolution du vecteur d'état de chaque particule en fonction de l'équation associée à son mode. L'étape de recalage se fait de la même façon que dans un cadre continu : les particules sont pondérées en fonction de leur vraisemblance à partir de l'observation du vecteur d'état numérique, puis sont rééchantillonnées.

Une étape supplémentaire consiste à calculer le mode le plus probable : on somme les poids de toutes les particules par mode de fonctionnement, et on peut donc déterminer la probabilité de chaque mode. Cette information peut soit servir uniquement à estimer le mode, soit à modifier le mode des particules recalées.

Le filtrage particulaire est facilement utilisable pour modéliser et estimer un système hybride. Les seuls inconvénients sont les suivants :

1. il n'y a pas de modélisation des transitions entre modes ;
2. l'observation est uniquement numérique, il n'y a pas d'observation symbolique du mode de fonctionnement ou de réception d'événements.

3 Réseaux bayésiens hybrides

La modélisation des systèmes hybrides par réseaux bayésiens (Lerner, 2002) permet de représenter des variables discrètes, dont les valeurs sont basées sur des tables de probabilités conditionnelles, ainsi que des variables continues, définies par des densités de probabilité conditionnelle. Le principe d'inférence bayésienne (chapitre 1, paragraphe 1.2) reste valide : le calcul de l'état du système est assuré par un principe d'estimation récursive basé sur la règle de Bayes.

Cette méthode d'inférence souffre des mêmes inconvénients que les réseaux bayésiens dynamiques : un manque de lisibilité des modèles et une limitation aux incertitudes évaluées (probabilités, possibilités).

Certains travaux (Doucet *et al.*, 2000) couplent un réseau bayésien modélisant le comportement du système à une méthode particulaire. Le réseau bayésien dynamique permet de modéliser le processus d'estimation : il permet d'une part de calculer la probabilité prédite sur l'état du système, et d'autre part, dans l'étape de recalage, de calculer la valeur de vraisemblance des particules à partir de règles d'inférence. Comme dans le filtrage

particulaire, l'incertitude sur l'état du système est échantillonnée, et chaque particule suit une évolution propre, ce qui permet d'évaluer le comportement hybride du système.

Le couplage entre réseau bayésien et filtrage particulaire permet d'apporter une information supplémentaire par rapport à une technique purement particulaire. Cependant cette modélisation est stochastique (nécessité d'attribuer une probabilité aux états et transitions) et la représentation est difficilement appréhendable.

4 Modèles discrets et filtres numériques

De nombreux travaux sont basés sur l'utilisation conjointe d'un modèle discret et d'un filtre numérique. Ces méthodes permettent d'utiliser les connaissances et les travaux sur les systèmes à événements discrets, et les performances des filtres numériques.

Parmi les modèles de systèmes à événements discrets, nous avons vu les techniques à base de logique, les graphes (éventuellement avec contraintes), les automates et les réseaux de Petri.

Les modèles à base de logique sont peu appropriés à un couplage avec un estimateur numérique, car ils peinent à modéliser le comportement hybride d'un système. Il en est de même pour les méthodes à base de graphes ou d'arbres.

En revanche, de nombreuses extensions ont été apportées aux automates et aux réseaux de Petri pour la modélisation des systèmes hybrides.

4.1 Automates hybrides

Nous avons vu dans le chapitre précédent le formalisme des automates, qui permet de représenter les modes de fonctionnement d'un système ainsi que les transitions entre ces modes. Le formalisme des automates hybrides, classiquement utilisé dans la littérature, l'étend pour représenter des systèmes hybrides : à chaque mode de l'automate est associée une équation d'évolution. L'état continu du système évolue alors selon le mode courant de l'automate.

Définition 2.1 (Automate hybride (de Dearden *et al.*, 2004; Zhao *et al.*, 2001))

Un automate hybride H est un n -uplet $\langle M, X, Init, \Sigma, \phi, \mathcal{F}, R \rangle$ avec

- $M = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots\}$ un ensemble d'états discrets (ou modes),
- $X \subset \mathbb{R}^n$ un ensemble d'états continus,
- $Init \subset M \times X$ l'ensemble des états initiaux,
- Σ un ensemble d'événements,
- $\phi : M \times \Sigma \rightarrow \mathcal{P}(M)$ la fonction de transition entre les modes de l'automate,
- \mathcal{F} qui à chaque mode associe une fonction d'évolution numérique,
- $R : M \times X \rightarrow 2^X$ la fonction de réinitialisation.

L'état de H est un couple $(\boldsymbol{\mu}_i, \mathbf{x}) \in M \times X$.

Ce modèle hybride peut également être étendu pour représenter d'autres comportements : automates hybrides temporels, automates hybrides stochastiques. . .

Oishi *et al.* (2002) utilisent ce formalisme pour modéliser une procédure d'approche. Les différents modes de l'automate correspondent aux modes de pilotage automatique, et l'état continu est le vecteur d'état de l'avion. L'utilisation des automates hybrides permet alors de vérifier le bon fonctionnement du système en calculant l'espace atteignable du système.

Glover et Lygeros (2004) proposent une méthode de simulation de l'évolution du trafic aérien pour le contrôle. Cette simulation est assurée à partir d'un ensemble d'automates hybrides, chacun de ces automates représentant le comportement d'un avion.

4.2 Réseaux de Petri hybrides

Contrairement aux automates hybrides, il existe plusieurs définitions d'un réseau de Petri hybride.

Les réseaux de Petri continus (David et Alla, 2005) permettent de modéliser un flux continu de jetons : au lieu qu'un jeton soit entièrement consommé ou ajouté à une place, il est transféré avec une certaine vitesse. Lorsqu'un réseau comporte des nœuds continus et des nœuds classiques, on dit qu'il est hybride. Ces réseaux sont particulièrement utilisés pour la modélisation de flots, par exemple pour la représentation de systèmes de réservoirs ou de valves. Un autre formalisme est adapté des réseaux de Petri hybrides pour la modélisation des systèmes de production (Balduzzi *et al.*, 2001).

Un formalisme assez proche, appelé réseau de Petri *fluide*, est proposé par Horton *et al.* (1998). Les arcs d'un réseau de Petri fluide portent des débits qui limitent le passage de jetons, et créent un marquage continu. Ce formalisme est également couplé avec des franchissements de transitions stochastiques (Fluid Stochastic Petri Nets).

Les réseaux de Petri à équations algébriques différentielles (Champagnat, 1998) modélisent l'évolution continue d'un système en associant à certaines places des équations différentielles. Le vecteur d'état du système évolue selon la place marquée. Ce formalisme, couplé à la modélisation orientée objet d'un jeton, est utilisé dans (Villani *et al.*, 2003) pour l'analyse du fonctionnement du train d'atterrissage d'un avion.

Les différents modèles permettent de modéliser le comportement d'un système hybride. Cependant, les modèles à base de vitesses de franchissement (réseaux continus, fluides) sont plutôt adaptés à la modélisation de flots (réseaux, trafic routier, systèmes de production. . .) Ils proposent une dynamique uniquement basée sur des vitesses et ne permettent donc pas de représenter n'importe quel système hybride. Les réseaux différentiels, qui associent une équation à certaines places du réseau, constituent un modèle beaucoup plus

général de système hybride.

4.3 Estimation hybride

L'estimation du comportement d'un système hybride à partir d'une modélisation par automate hybride se retrouve principalement dans deux approches, l'une utilisant des filtres des Kalman, l'autre le filtrage particulaire.

4.3.1 Automates hybrides probabilistes concurrents

Les travaux de Hofbaur et Williams (2002) sont appliqués à la surveillance de BIO-Plex, un environnement de type "biosphère artificielle" destiné à permettre la survie d'une équipe d'hommes sur une planète, en leur fournissant de quoi se nourrir et respirer. Pour cela, BIO-Plex est constitué d'un ensemble de chambres, chacune ayant une fonction propre. La chambre qui nous intéresse ici est la chambre de culture de plantes (figure 2.1) qui permet à la fois de nourrir l'équipe et de convertir le gaz carbonique en oxygène.

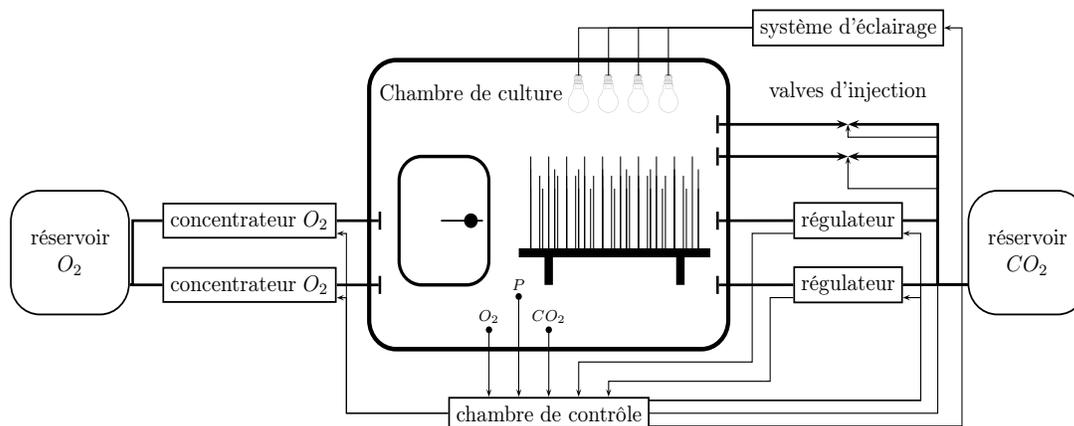


FIG. 2.1: BIO-Plex (traduit de Hofbaur, 2004)

Le but du système de surveillance est de suivre le fonctionnement de la chambre de contrôle, qui doit garantir un taux "normal" d'oxygène si un humain entre dans la chambre, et une ambiance permettant la croissance des plantes lorsque la chambre est vide.

Le comportement du système est modélisé par un ensemble d'automates hybrides probabilistes. Pour estimer l'état du système, plusieurs filtres de Kalman sont utilisés.

Chaque filtre est associé à une trajectoire possible du système. L'estimation de l'état associé à ce filtre est opérée de façon classique, par une prédiction de cet état à partir des

transitions discrètes et des fonctions d'évolution, puis par un recalage qui met à jour la distribution de probabilité à partir de l'observation.

Pour éviter d'avoir une méthode complexe utilisant un filtre par trajectoire, un processus de sélection est mis en œuvre. Ainsi, à chaque observation, les trajectoires les plus probables (en fonction de l'observation) sont conservées. Les filtres correspondant aux autres trajectoires sont supprimés, de façon à assurer une complexité limitée.

Le diagnostic d'un état défaillant repose sur le modèle de cette défaillance, c'est un *diagnostic à base de modèles*. Pour assurer un bon diagnostic du système, il est donc impératif de représenter le plus de comportements défaillants possibles. Il est donc totalement impossible de diagnostiquer un comportement non modélisé, et donc "inconnu".

4.3.2 Automates hybrides et filtrage particulière

Les travaux de Koutsoukos *et al.* (2003) se basent également sur un automate hybride pour représenter le fonctionnement d'un système de réservoirs. Ils utilisent une approche particulière pour développer un processus d'estimation : chaque particule correspond à un état hybride du système, c'est-à-dire qu'elle porte non seulement le vecteur d'état continu, mais également le mode estimé du système.

Il s'agit donc ici d'enrichir les techniques d'estimation classiques (paragraphe 2.2) par un modèle discret permettant de décrire les transitions entre modes.

Comme dans le cas précédent, le diagnostic d'états défaillants est à base de modèles de ces défaillances.

4.3.3 Estimation hybride et réseaux de Petri : un bon ménage ?

À notre connaissance, il n'existe aucune méthode d'estimation de systèmes hybrides mettant en œuvre un filtre numérique couplé à un réseau de Petri hybride.

Le seul estimateur utilisant un réseau de Petri est celui proposé dans (Zhao *et al.*, 2001). Un réseau de Petri temporel modélise le comportement discret du système. Ce modèle permet de prédire les modes dans lesquels peut se trouver le système. Cette prédiction est cependant uniquement basée sur l'évolution temporelle, et non pas sur son évolution continue (le réseau n'est pas hybride).

Afin d'évaluer la pertinence d'une méthode d'estimation hybride utilisant un réseau de Petri hybride, le tableau 2.1 reprend les estimateurs basés sur des automates vus dans les paragraphes précédents.

Les deux approches sont assez similaires : elles sont toutes les deux basées sur un processus de sélection des meilleures estimations. Elles ont également les mêmes inconvénients :

	(Hofbaur et Williams, 2002)	(Koutsoukos <i>et al.</i> , 2003)
Modèle discret	Automates hybrides probabilistes	Automates hybrides
Estimation numérique	Filtres de Kalman	Filtrage particulaire
Sélection	Trajectoires les plus probables	Rééchantillonnage
Observations symboliques	non	non
Remarques	Transitions stochastiques Estimation du mode Diagnostic à base de modèles	

TAB. 2.1: Synthèse des estimateurs hybrides

- aucune prise en compte d’observations symboliques ou d’événements, et donc une estimation de mode uniquement basée sur une mesure numérique,
- une représentation probabiliste de l’incertitude sur les variables, les transitions et les observations,
- un diagnostic à base de modèles, basé sur la probabilité du mode défaillant.

Ces deux derniers points constituent un des inconvénients majeurs des méthodes de diagnostic à base de modèle dans un cadre stochastique. Cet inconvénient repose sur le fait que la probabilité de défaillance du système est faible, dans la plupart des cas. Hors, la faible valeur de probabilité associée à cette défaillance peut entraîner la non reconnaissance de cette défaillance, car la trajectoire correspondante du système a pu être écartée par l’estimateur.

Proposer un estimateur hybride utilisant des réseaux de Petri hybrides peut combler ces lacunes :

- les notions de macro-marquages et de marquages possibilistes permettent d’envisager un principe d’estimation utilisant des observations symboliques ;
- la prise en compte d’événements non valués est possible ;
- la modélisation plus riche apportée par un réseau de Petri et les techniques d’analyse associées peuvent permettre un diagnostic mixte du système, par la détection d’états défaillants modélisés et non modélisés.

La contribution que nous apportons concerne donc une méthode hybride couplant un filtre numérique à un réseau de Petri.

5 Vers un estimateur par réseaux de Petri hybrides

Le premier objectif de nos travaux est de définir un modèle de réseau de Petri adapté aux systèmes hybrides et à la représentation de l’incertitude associée. Pour cela les points

suivants sont à prendre en compte :

5.1 Décomposition en composants

L'étude de systèmes complexes peut nécessiter une modélisation par composants. Cette décomposition du système a des avantages certains, au niveau de la phase de modélisation, de vérification, de compréhension du modèle, de distribution d'algorithmes. . .

C'est pourquoi il semble intéressant de pouvoir décomposer le mode de fonctionnement du système en sous-modes. Dans l'exemple des thermostats (paragraphe 5.3.3), les sous-modes peuvent par exemple correspondre à la présence d'un thermostat dans chaque pièce, au mode de fonctionnement de ces thermostats, aux températures des pièces. . .

Le système hybride SH est donc un ensemble de sous-systèmes $\{S_1, \dots, S_h\}$. Par conséquent, le mode du système se décompose en sous-modes : $\boldsymbol{\mu} = (\mu_1, \dots, \mu_h)$.

Parmi ces sous-systèmes, certains sont purement discrets (mode du thermostat, état d'une valve, d'un bouton), d'autres ont une dynamique continue ou hybride (température de la pièce, position d'un mobile, pression dans un réservoir).

Le modèle de réseau de Petri hybride doit permettre une représentation par composants discrets et/ou continus.

5.2 Représentation de l'incertitude

La définition d'un principe d'estimation implique la modélisation de l'incertitude sur l'état du système. Cette incertitude est hybride : numérique, sur le vecteur d'état du système, et symbolique, sur le mode de fonctionnement du système. L'incertitude numérique, gérée par l'utilisation d'un filtre numérique, sera conditionnée par le filtre utilisé et la représentation associée.

En revanche, l'incertitude symbolique est directement représentée dans le réseau de Petri. Les méthodes d'estimation du marquage d'un réseau de Petri, (paragraphe 2.2.5, chapitre 1), reposent principalement sur l'utilisation d'un macro-marquage. Un macro-marquage représente l'incertitude sur l'état du système par la présence de jetons dans différentes places : l'état réel du système correspond à une partie de ce marquage.

Cette modélisation est incompatible avec la représentation de l'état du système mise en œuvre par Champagnat (1998) : le vecteur d'état numérique ne peut pas évoluer selon un "sur-ensemble" d'équations, et il devient donc nécessaire d'inclure ce vecteur dans les jetons. Ainsi, il devient possible de représenter les différentes évolutions possibles de ce vecteur d'état.

Le modèle de réseau de Petri hybride doit permettre la représentation de l'incertitude en gérant les paramètres liés à l'état du système de façon locale.

5.3 Quelques exemples de systèmes hybrides

Les systèmes hybrides présentés ci-après serviront, tout au long de notre proposition, d'illustrations et de cas d'études pour appuyer nos réflexions et illustrer nos résultats.

5.3.1 Thermostat : gestion de la température d'une pièce

L'exemple du thermostat est un classique de la modélisation des systèmes non linéaires ou des systèmes hybrides. L'état du système que l'on veut estimer est composé d'un vecteur numérique contenant la température de la pièce contrôlée par le thermostat ($\mathbf{x} = \theta$) et d'un mode de fonctionnement $\boldsymbol{\mu}$. Les modes possibles du système sont :

- en chauffe ($\boldsymbol{\mu} = on$)
- en veille ($\boldsymbol{\mu} = idle$)
- arrêté ($\boldsymbol{\mu} = off$)

Le thermostat doit, lorsqu'il est en marche, réguler la température de la pièce entre θ_{min} et θ_{max} , ce qui constitue les conditions liées aux changements de mode entre *on* et *idle*. L'arrêt du thermostat peut être effectué par un opérateur humain par exemple, et correspond au fait qu'une action ou qu'un événement σ_{off} se produise.

L'évolution de la température de la pièce est régie par l'équation :

$$\dot{\theta} = \alpha + \beta(\theta_{ext} - \theta) \quad (2.1)$$

où α est une constante de chauffage, égale à 0 si le thermostat est arrêté ou en veille, β une constante et θ_{ext} la température extérieure.

5.3.2 Navigation dans un champ de vitesse

Ce cas traite d'un objet en déplacement (Fehnker et Ivancic, 2004) dont le vecteur d'état contient sa position sur un plan et sa vitesse. Son mode de fonctionnement correspond à la zone de l'espace sur laquelle il se trouve.

Les transitions entre modes se font en fonction de la position de l'objet et du franchissement de frontières entre zones. Dans les exemples traités, le plan est découpé en zones selon une grille (voir figure 2.2). L'évolution du vecteur d'état de l'objet est régie par les équations :

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v} &= \alpha(v_d - v) \end{aligned} \quad (2.2)$$

où v_d est la vitesse désirée, dépendant de la zone sur laquelle se trouve le mobile, v_x et v_y les composantes en x et en y de la vitesse v , et α une constante.

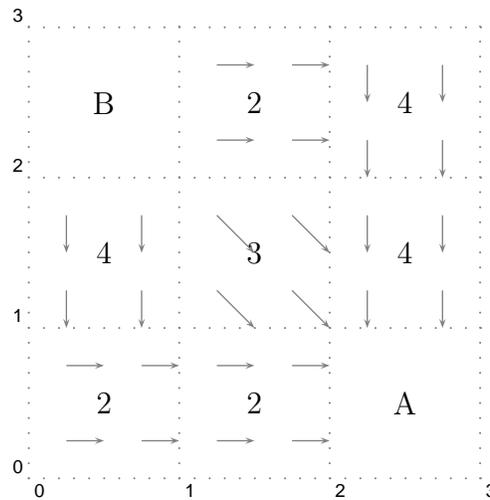


FIG. 2.2: Exemple de grille de navigation 3x3 incluant les vitesses désirées (direction) ainsi qu'une zone cible (A) et une zone dangereuse (B).

Dans la figure 2.2, la vitesse désirée est donnée par la valeur d'une direction, indiquée dans la case correspondant à la zone. À partir de ce nombre i , la vitesse désirée est calculée selon l'équation :

$$v_d = \begin{pmatrix} \sin(i\frac{\pi}{4}) \\ \cos(i\frac{\pi}{4}) \end{pmatrix} \quad (2.3)$$

Ce benchmark permet d'étudier le comportement du système, notamment en fonction du nombre de modes du système. En effet, créer une grande instance de ce problème revient à créer une carte comportant un grand nombre de zones (donc de modes). En revanche, ni la taille du vecteur numérique, ni le nombre de branchements (transitions possibles depuis un mode) n'augmentent lorsque la taille du problème augmente.

5.3.3 Partage de thermostats

Nous avons vu que le thermostat est un exemple classique de système hybride, qui permet d'illustrer clairement les méthodologies mises en œuvre. Il apparaît cependant quelquefois limité ou trop "simpliste".

Le système décrit ici permet d'enrichir cet exemple (Fehnker et Ivancic, 2004) : on considère un ensemble de pièces, que l'on souhaite chauffer avec un ensemble de thermostats. Le problème est que l'on dispose de moins de chauffages que de pièces. Il est cependant possible de déplacer des chauffages d'une pièce dans une autre.

Le système est décrit par la présence d'un chauffage dans une pièce et la température de chaque pièce, qui est gérée par un thermostat. La température de la i -ème pièce dépend de la température extérieure et de la présence d'un chauffage en train de chauffer. Elle

dépend également des températures des pièces adjacentes :

$$\dot{\theta}_i = \alpha_i + \beta_i(\theta_{ext} - \theta_i) + \sum_{i \neq j} \gamma_{i,j}(\theta_j - \theta_i) \quad (2.4)$$

où $\gamma_{i,j}$ vaut 0 si les pièces i et j ne sont pas voisines.

Chaque thermostat régule la température des pièces entre $\theta_{i_{min}}$ et $\theta_{i_{max}}$. Il est possible de déplacer un chauffage d'une pièce i à une pièce j si :

1. i et j sont adjacentes ;
2. i a un chauffage ;
3. j n'en a pas ;
4. $\theta_j \leq \theta_{j_{get}}$;
5. $\theta_i - \theta_j \geq \theta_{j_{dif}}$.

où $\theta_{j_{get}}$ est la température à partir de laquelle la pièce j "réclame" un chauffage, et $\theta_{j_{dif}}$ la différence de température nécessaire à un échange de thermostat vers la pièce j .

Ce système est intéressant pour l'évaluation des systèmes hybrides car il présente des transitions non-déterministes : il se peut que plusieurs échanges de chauffage soient possibles à un instant donné. Ainsi, augmenter la taille du problème en augmentant le nombre de pièces et de chauffages augmente la taille du vecteur d'état numérique (il faut considérer plus de températures) et complexifie la dynamique du système en augmentant le "niveau" de non-déterminisme.

6 Conclusion

Nous avons vu dans ce chapitre les méthodes d'estimation des systèmes hybrides. Ces méthodes sont principalement de deux types :

- les filtres multimodèles, qui utilisent plusieurs filtres numériques pour estimer les différents modes de fonctionnement du système ; ces méthodes ont une représentation pauvre (voire inexistante) des transitions entre modes, et n'observent pas le mode de fonctionnement du système ;
- les techniques hybrides, couplant un automate hybride à un filtre numérique ; ces méthodes enrichissent le modèle dynamique du système, mais utilisent une modélisation stochastique de cette dynamique, n'observent pas le mode de fonctionnement du système, et le diagnostic est à base de modèles.

Ces constatations nous mènent donc à proposer une technique originale, couplant un réseau de Petri hybride à un filtre numérique. Cet estimateur devra répondre aux critiques précédentes, en modélisant des événements sans information probabiliste, en acceptant

l'observation du mode du système, et en diagnostiquant aussi bien des défaillances modélisées que non modélisées.

La suite de ce manuscrit s'organise de la façon suivante :

- dans le chapitre 3, nous définissons le modèle des réseaux de Petri particulières, les règles d'évolution qui y sont associées et illustrons ce formalisme au travers des différents exemples présentés ci-dessus ;
- le chapitre 4 propose un principe d'estimation basé sur le modèle des réseaux de Petri particulières, et mettant en œuvre une batterie de filtres de Kalman ;
- le chapitre 5 propose un principe d'estimation similaire, inspiré du filtrage particulière ;
- à partir de ce principe d'estimation, nous proposons dans le chapitre 7 une méthode de diagnostic permettant de détecter des états anormaux sans modèle explicite de leur comportement ;
- nous verrons dans la partie finale de cette thèse comment ce principe d'estimation et de diagnostic se comporte sur plusieurs cas d'études, et notamment sur le suivi de l'activité de pilotage et la détection de conflits.

Chapitre 3

Réseaux de Petri particulières

L'état d'un **système hybride** est défini par un **vecteur** d'état numérique et par un **mode** de fonctionnement discret. L'utilisation des réseaux de Petri permet de modéliser la dynamique hybride d'un tel système, en utilisant conjointement les règles de franchissement pour la représentation de la dynamique discrète et des équations d'évolution pour la dynamique continue. Dans ce chapitre nous définissons un réseau de Petri particulière, son comportement, ses propriétés, et présentons quelques exemples permettant d'illustrer la modélisation et la simulation des systèmes hybrides.

1 Les réseaux de Petri particulières

1.1 Définitions

Nous proposons un modèle de réseau de Petri permettant de représenter un système hybride. Ce modèle doit représenter le comportement numérique par les équations d'évolution du système, et le comportement discret par les changements de marquages liés à la satisfaction de conditions ou la réception d'événements.

Définition 3.1 (Réseau de Petri particulière)

Un réseau de Petri particulière est un n -uplet $\langle P, T, Pre, Post, X, \mathcal{F}, \delta, A_c, \mathcal{M}_0 \rangle$ où

- P est l'ensemble des places, partitionné en places numériques P_N et places symboliques P_S ;*

- T est l'ensemble des transitions ;
- Pre et $Post$ sont les fonctions "places amont" et "place aval" du réseau, définies par

$$\forall t \in T, \forall p \in P_S, \quad Pre(t, p) (\text{resp. } Post(t, p)) \in \mathbb{N} \quad (3.1)$$

$$\forall t \in T, \forall p \in P_N, \quad Pre(t, p) (\text{resp. } Post(t, p)) \in \mathbf{x}.\mathbb{N} \quad (3.2)$$

$Pre(t, p)$ (respectivement $Post(t, p)$) représente le nombre de jetons (ou de particules dans le cas d'une place numérique) consommés (respectivement produits) par le franchissement de la transition t ; nous noterons également $Pre(t)$, ou encore $\bullet t$, l'ensemble des places p telles que $Pre(p, t) \neq 0$;

- $C = Post - Pre$ est la matrice d'incidence du réseau ;
- $X \subset \mathbb{R}^n$ est l'espace d'état numérique du système ;
- \mathcal{F} est l'application qui, à chaque place numérique, associe une fonction d'évolution discrète :

$$\forall p \in P_N, \mathcal{F}(p) : X \times \mathbb{N} \longrightarrow X \quad (3.3)$$

- δ est l'application qui à chaque place symbolique associe une configuration du système sous la forme d'un ensemble de propriétés :

$$\forall p \in P_S, \delta(p) \in 2^\Gamma \quad (3.4)$$

où Γ est ensemble fini de propriétés possibles du système ;

- A_c est l'application qui à chaque transition t associe une condition $A_c(t)$ définie par :

$$A_c(t) = \sigma \in \Sigma \quad (3.5)$$

si la condition de franchissement est basée sur la réception d'un événement externe (Σ est l'ensemble des événements acceptés par le système), ou bien

$$A_c(t) = [a, b] \subset \mathbb{R}^N \quad (3.6)$$

c'est-à-dire que la condition de la transition t est un intervalle de \mathbb{R}^N , et donc $A_c(t)$ est satisfaite par le vecteur x si et seulement si $x \in A_c(t)$ ¹ ;

- \mathcal{M}_0 est le marquage initial du réseau, défini par :

$$\forall p \in P_S, \quad \mathcal{M}_0(p) \in \mathbb{N} \quad (3.7)$$

$$\forall p \in P_N, \quad \mathcal{M}_0(p) \text{ un ensemble fini de particules } \pi^i \in X \quad (3.8)$$

où une particule est un vecteur d'état numérique du système.

Un réseau de Petri particulière est donc composé de deux types de places. Chaque place numérique représente un mode de fonctionnement numérique et peut être marquée par un ensemble de vecteurs d'état numériques du système. À chaque place numérique est associée une fonction d'évolution. L'évolution d'un vecteur \mathbf{x} dans la place p du temps k au temps $k + 1$ est décrite par l'équation :

$$\mathbf{x}(k + 1) = \mathcal{F}(p, k, \mathbf{x}(k)) \quad (3.9)$$

Remarque 3.2

Discrétisation Pour simuler le comportement d'un système hybride, nous avons fait le choix d'échantillonner l'évolution continue. Ainsi, l'évolution continue est appliquée sur un intervalle temporel de longueur 1 (une unité de temps).

1.2 Propriétés

1.2.1 Réseau de Petri sous-jacent

La modélisation d'un système par réseaux de Petri met souvent en œuvre des réseaux interprétés, dans lesquels les éléments du réseau ont une sémantique, parfois représentée par des attributs particuliers (couleurs, prédicats, gardes...)

L'analyse des propriétés d'un réseau de Petri hybride passe par la simplification du modèle : on appelle alors **réseau sous-jacent** le réseau de Petri classique ayant la même structure que le réseau initial. Nous reprenons cette définition pour le cas des réseaux de Petri particulières.

Définition 3.3 (Réseau sous-jacent)

Soit $\mathcal{R} = \langle P, T, C, X, \mathcal{F}, \delta, A_c, \mathcal{M}_0 \rangle$ un réseau de Petri particulière. Son **réseau sous-jacent** est $\mathcal{R}_{sj} = \langle P, T, C, \mathcal{M}_{0sj} \rangle$, avec \mathcal{M}_{0sj} le marquage initial sous-jacent, défini par

$$\forall p \in P, \mathcal{M}_{0sj}(p) = |\mathcal{M}_0(p)| \quad (3.10)$$

Le réseau sous-jacent a donc perdu les informations de fonctions, de configurations et de conditions associées aux places et transitions du réseau particulière. Le marquage d'une place correspond au nombre de jetons présents dans cette place, indépendamment de leur type.

¹nous noterons parfois dans la suite $A_c(t)(x)$ la satisfaction de la condition $A_c(t)$ par le vecteur x , i.e. le fait que $x \in A_c(t)$.

1.2.2 Graphe d'état numérique

Un réseau de Petri ordinaire est un graphe d'état si chaque transition n'est reliée qu'à une seule place en entrée et en sortie. Un graphe d'état représente le fonctionnement d'un système en associant une place à chaque composante d'état de ce système.

Dans le cas de la modélisation d'un système hybride, l'un des objectifs de la définition d'un réseau de Petri particulière est d'associer un mode de fonctionnement numérique à chaque place numérique du réseau. Il est donc souhaitable que la partie numérique du réseau soit un graphe d'état.

Définition 3.4 (Projection numérique)

Soit \mathcal{R} un réseau de Petri particulière. La **projection numérique** de \mathcal{R} est le sous-réseau $\mathcal{R}_N = \langle P_N, T_N, C_N \rangle$, défini par :

- P_N est l'ensemble des places numériques du réseau particulière,
- T_N est l'ensemble des transitions du réseau qui sont connectées à au moins une place numérique

$$T_N = \{t \in T \mid (\bullet t \cup t \bullet) \cap P_N \neq \emptyset\}^2 \quad (3.11)$$

- C_N est la matrice d'incidence projetée, de dimension $|P_N| \times |T_N|$, définie par

$$\forall p \in P_N, \forall t \in T_N, C_N(p, t) = C(p, t) \quad (3.12)$$

Définition 3.5 (Graphe d'état numérique)

Un réseau de Petri particulière \mathcal{R} est un **graphe d'état numérique** si et seulement si sa projection numérique \mathcal{R}_N est un graphe d'état.

De plus, si le marquage initial du réseau ne contient qu'un seul vecteur numérique, alors le réseau est **numériquement sauf** :

Définition 3.6 (Réseau numériquement sauf)

Un réseau de Petri particulière \mathcal{R} initialement marqué est **numériquement sauf** si et seulement si

$$\forall p \in P_N, \forall \mathcal{M} \text{ accessible à partir de } \mathcal{M}_0, \mathcal{M}_{sj}(p) \leq 1 \quad (3.13)$$

² $\bullet t$ est l'ensemble des places amont de t , $t \bullet$ est l'ensemble des places aval de t .

Les règles d'évolution que nous proposons dans la suite de ce chapitre sont définies pour un réseau particulière *graphe d'état numérique, numériquement sauf*. Cette propriété assure en effet que le vecteur d'état numérique du système est unique et donc qu'à tout instant, ce vecteur d'état est dans un seul mode de fonctionnement et ne suit donc qu'une équation d'évolution.

1.3 Évolution du marquage

L'évolution du marquage d'un réseau de Petri particulière représente l'évolution de l'état du système. Elle est donc hybride : l'évolution discrète représente les changements de mode du système, l'évolution continue représente l'évolution du vecteur d'état numérique du système.

Cette évolution se fait donc de deux manières différentes :

- sur réception d'événements (conditions satisfaites, actions extérieures) pour la dynamique discrète, correspondant au franchissement des transitions,
- de façon continue pour la dynamique numérique, où la valeur du vecteur d'état numérique évolue selon une équation différentielle.

La notion de temps vient donc enrichir le modèle : le marquage d'un réseau de Petri n'évolue plus uniquement par franchissement des transitions, mais en fonction du temps. Il devient donc nécessaire de noter le marquage au temps k par $\mathcal{M}(k)$ ³.

1.3.1 Évolution isoparticule

L'évolution des jetons dans le réseau représente la dynamique discrète du système. Elle se fait en utilisant les règles de franchissement des transitions. Lors de ces franchissements, les jetons se déplacent, mais les valeurs des vecteurs n'évoluent pas : il s'agit d'une **évolution isoparticule**.

Définition 3.7 (Transition franchissable)

Soit \mathcal{R} un réseau de Petri particulière et $\mathcal{M}(k)$ son marquage au temps k . Une transition $t \in T$ est **franchissable** si et seulement si

$$\forall p \in P_S, \quad \mathcal{M}(k, p) \geq Pre(p, t) \quad (3.14)$$

$$\forall p \in P_N \cap \bullet t, \quad A_c(t)(\mathcal{M}(k, p)) \quad (3.15)$$

Une transition t est franchissable si chacune de ses places amont dispose de suffisamment de jetons, et si le vecteur d'état satisfait sa condition $A_c(t)$.

³Le marquage au temps k de la place p est noté $\mathcal{M}(k, p)$.

Définition 3.8 (Tir d'une transition)

Soit un réseau de Petri particulière et $\mathcal{M}(k)$ son marquage au temps k . Le tir d'une transition franchissable t conduit à un marquage $\mathcal{M}'(k)$, défini par

$$\forall p \in P, \mathcal{M}'(k, p) = \mathcal{M}(k, p) + C(p, t) \quad (3.16)$$

Le tir d'une transition t représente un changement de mode de fonctionnement du système. Le tir est instantané.

Algorithme 3.1 Évolution isoparticule

ENTRÉES: \mathcal{R}

tantque $\exists t \in T$ franchissable **faire**
 tir de t
fin tantque

L'évolution isoparticule consiste à faire évoluer le marquage du réseau, tant que cela est possible. Cette évolution représente les changements de modes de fonctionnement du système à l'instant k .

Cette évolution est finie si le modèle du système est **non-Zénon**. Un modèle est dit **Zénon** s'il autorise, pendant un temps fini, une infinité de transitions (Ames *et al.*, 2005). Dans le cas d'un réseau de Petri particulière, ce comportement se traduit par le tir d'une infinité de transitions lors d'une évolution isoparticule, où le temps n'évolue pas.

Définition 3.9 (Réseau Zénon)

Soit \mathcal{R} un réseau de Petri particulière, graphe d'état numérique. \mathcal{R} est **Zénon** si et seulement si $\exists \mathcal{M}$ tel que $\langle \mathcal{R}, \mathcal{M} \rangle$ est numériquement sauf et \mathcal{R} admet une séquence de tirs infinie à partir de \mathcal{M} .

Remarque 3.10 (Zénon)

Un modèle Zénon d'un système est un "mauvais" modèle. En effet, dans la réalité, le temps s'écoule, et il est donc impossible que le système suive une évolution infinie dans un temps fini. Cette mauvaise modélisation correspond le plus souvent à une simplification du comportement du système.

Considérons par exemple le réseau de Petri particulière du problème de navigation (figure 3.2). Si nous réécrivons les conditions des transitions t_0 et t_2 en $x \geq 1$ et $x \leq 1$ respectivement (au lieu d'avoir des inégalités strictes), alors nous sommes en présence d'un modèle

Zénon. En effet, un particule dont la valeur est $x = 1$ et marquant la place p_0 va pouvoir franchir successivement t_0 et t_2 indéfiniment.

La caractérisation des réseaux de Petri non-Zénon n'est pas simple. Il faut en effet, pour respecter la définition, calculer le graphe des marquages accessibles pour tout marquage possible du réseau. Même si on se limite aux marquages numériquement saufs, le nombre de cas est encore infini (il faut en particulier parcourir tout l'espace d'état numérique).

Il est cependant possible de trouver des conditions suffisantes pour montrer qu'un réseau de Petri particulière initialement marqué est non-Zénon.

Théorème 3.11 (Réseau non-Zénon)

Soit \mathcal{R} un réseau de Petri particulière et \mathcal{M}_0 son marquage initial. \mathcal{R} est non-Zénon pour \mathcal{M}_0 si l'une des propriétés suivantes est vérifiée :

1. \mathcal{G}_{s_j} le graphe des marquages accessibles à partir de \mathcal{M}_0 du réseau sous-jacent \mathcal{R}_{s_j} est un arbre⁴ ;
2. \mathcal{G}_{s_j} est acyclique ;
3. \mathcal{G}_{s_j} ne contient aucune séquence répétitive :

$$\exists y \in \mathbb{N}^{|T|} / C.y \geq 0 \quad (3.17)$$

Exemple 3.12 (Réseau non-Zénon) Considérons les réseaux de Petri de la figure 3.1. Dans le réseau de gauche (figure 3.1(a)), le graphe des marquages accessibles est un arbre,

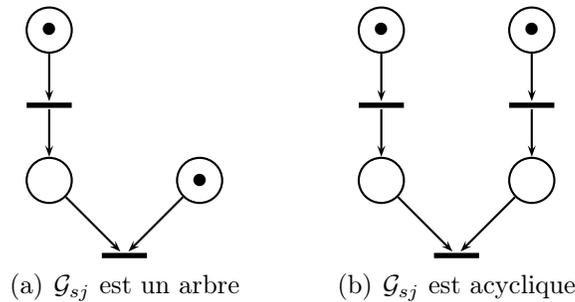


FIG. 3.1: Réseaux de Petri non-Zénon

puisque, depuis le marquage initial (1 0 1) les marquages accessibles sont :

$$(1\ 0\ 1) \rightarrow (0\ 1\ 1) \rightarrow (0\ 0\ 0)$$

⁴Un arbre est un graphe connexe et acyclique.

Le réseau est non-Zénon.

Pour le réseau de droite (figure 3.1(b)), le graphe des marquages n'est pas un arbre, cependant le graphe des marquages est acyclique, et donc le réseau est non-Zénon.

Remarque 3.13 (Réseau non-Zénon)

Les conditions suffisantes énoncées dans le théorème 3.11 ne concernent que la structure du réseau. D'autres conditions doivent être définies en utilisant notamment les conditions liées aux transitions du réseau de Petri particulière.

1.3.2 Évolution isomarquage

L'évolution continue du système est régie par les équations différentielles associées aux places numériques. Durant cette évolution les jetons ne se déplacent pas dans le réseau, le marquage reste fixe. Cette étape est une **évolution isomarquage**.

Définition 3.14 (Évolution isomarquage)

L'évolution isomarquage d'un marquage $\mathcal{M}(k)$ du temps k au temps $k + 1$ conduit à un marquage $\mathcal{M}(k + 1)$ défini par

$$\forall p \in P_S, \mathcal{M}(k + 1, p) = \mathcal{M}(k, p) \quad (3.18)$$

$$\forall p \in P_N, \mathcal{M}(k + 1, p) = \mathcal{F}(p, k, \mathcal{M}(k, p)) \quad (3.19)$$

La simulation d'un réseau particulière peut se faire en résolvant les équations différentielles pour calculer les instants de sensibilisation des transitions, comme cela est proposé dans (Champagnat, 1998) pour les réseaux prédicat-transition-différentiels.

2 Quelques exemples

2.1 Navigation

Le problème de navigation représente le déplacement d'un objet dans un champ de vecteurs (figure 2.2). Le vecteur d'état numérique du système est :

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} \in \mathbb{R}^4 \quad (3.20)$$

2.1.1 Modélisation

La modélisation de ce problème (figure 3.2) consiste à associer une place à chaque zone du plan de navigation, et à associer aux transitions les conditions de passage entre ces zones.

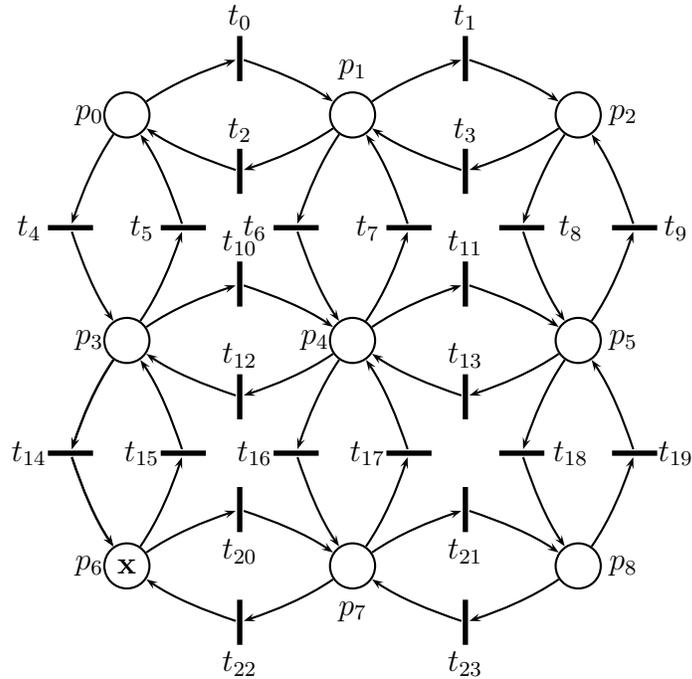


FIG. 3.2: Modélisation du problème de navigation, grille de la figure 2.2

Reprenons la définition d'un réseau particulière (définition 3.1) :

– toutes les places sont numériques

$$P = \{p_0, \dots, p_8\} = P_N, \quad P_S = \emptyset \quad (3.21)$$

– les transitions sont

$$T = \{t_0, \dots, t_{23}\} \quad (3.22)$$

– la matrice d'incidence est définie par

$${}^tC = \begin{matrix} & p_0 & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 \\ \begin{matrix} t_0 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \\ t_{10} \\ t_{11} \\ t_{12} \\ t_{13} \\ t_{14} \\ t_{15} \\ t_{16} \\ t_{17} \\ t_{18} \\ t_{19} \\ t_{20} \\ t_{21} \\ t_{22} \\ t_{23} \end{matrix} & \begin{pmatrix} -\mathbf{x} & \mathbf{x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mathbf{x} & \mathbf{x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x} & -\mathbf{x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{x} & -\mathbf{x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{x} & 0 & 0 & \mathbf{x} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x} & 0 & 0 & -\mathbf{x} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mathbf{x} & 0 & 0 & \mathbf{x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{x} & 0 & 0 & -\mathbf{x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{x} & 0 & 0 & \mathbf{x} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{x} & 0 & 0 & -\mathbf{x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{x} & \mathbf{x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\mathbf{x} & \mathbf{x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{x} & -\mathbf{x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{x} & 0 & 0 & \mathbf{x} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{x} & 0 & 0 & -\mathbf{x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\mathbf{x} & 0 & 0 & \mathbf{x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{x} & 0 & 0 & -\mathbf{x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mathbf{x} & 0 & 0 & 0 & \mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{x} & 0 & 0 & 0 & -\mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{x} & \mathbf{x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mathbf{x} & \mathbf{x} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x} & -\mathbf{x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{x} & -\mathbf{x} \end{pmatrix} & \end{matrix} \quad (3.23)$$

– les équations associées aux places correspondent à l'équation (2.2) et, par exemple, l'équation de la place p_6 est :

$$\mathbf{x}(k+1) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 - \alpha_{11} & -\alpha_{12} \\ 0 & 0 & -\alpha_{21} & 1 - \alpha_{22} \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ 0 \\ \alpha_{11} \\ \alpha_{21} \end{pmatrix} \quad (3.24)$$

où $\alpha \in \mathbb{R}^{2 \times 2}$ est la constante du problème ;

- $\delta = \emptyset$ puisque aucune place n'est symbolique ;
- les conditions associées aux transitions portent sur le franchissement d'une frontière entre deux zones et, par exemple, la transition t_{20} porte la condition :

$$A_c(t_{20}) = (\mathbf{x} > \begin{pmatrix} 1 \\ -\infty \\ -\infty \\ -\infty \end{pmatrix}) \quad (3.25)$$

– le marquage initial est :

$$\mathcal{M}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{x}_0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{avec } \mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.26)$$

2.1.2 Propriétés

Le réseau de Petri de la figure 3.2 est un graphe d'état : chaque transition n'a qu'une place en entrée et une place en sortie. Toutes les places de ce réseau sont numériques. C'est donc un **graphe d'état numérique**.

De plus, son marquage initial \mathcal{M}_0 ne contient qu'un seul jeton (dans p_6), donc ce réseau est également **numériquement sauf**.

2.1.3 Déplacement de l'objet

Considérons le réseau de Petri de la figure 3.2, avec pour marquage initial \mathcal{M}_0 (3.26).

L'évolution de l'état du système s'opère en plusieurs étapes :

1. une **évolution isoparticule** : on fait évoluer le mode de fonctionnement du système à l'instant 0 ; aucune transition n'est franchissable (pour t_{15} et t_{20} , les conditions ne sont pas satisfaites) ;
2. une **évolution isomarquage** : on fait évoluer le vecteur d'état entre l'instant 0 et l'instant 1

$$\mathcal{M}(1, p_6) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 - \alpha_{11} & -\alpha_{12} \\ 0 & 0 & -\alpha_{21} & 1 - \alpha_{22} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \alpha_{11} \\ \alpha_{21} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \alpha_{11} \\ \alpha_{21} \end{pmatrix} \quad (3.27)$$

et pour $\alpha = 0.4I$,

$$\mathcal{M}(1, p_6) = \begin{pmatrix} 0 \\ 0 \\ 0.4 \\ 0 \end{pmatrix} \quad (3.28)$$

3. une évolution isoparticule à l'instant 1 : aucune transition n'est franchissable ;
4. une évolution isomarquage entre 1 et 2 :

$$\mathcal{M}(2, p_6) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 - 0.4 & 0 \\ 0 & 0 & 0 & 1 - 0.4 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0.4 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0.4 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0 \\ 0.64 \\ 0 \end{pmatrix} \quad (3.29)$$

5. une évolution isoparticule à l'instant 2 : aucune transition n'est franchissable ;
6. une évolution isomarquage entre 2 et 3 :

$$\mathcal{M}(3, p_6) = \begin{pmatrix} 1.04 \\ 0 \\ 0.78 \\ 0 \end{pmatrix} \quad (3.30)$$

7. une évolution isoparticule à l'instant 3 : la transition t_{20} est sensibilisée, puisque p_6 contient un jeton et que

$$\mathbf{x} = \begin{pmatrix} 1.04 \\ 0 \\ 0.78 \\ 0 \end{pmatrix} > \begin{pmatrix} 1 \\ -\infty \\ -\infty \\ -\infty \end{pmatrix} \quad (3.31)$$

On a donc $A_c(t_{20})(\mathbf{x})$. La transition t_{20} est donc tirée, et le marquage résultant est celui de la figure 3.3

8. et ainsi de suite...

Différentes évolutions du système sont représentées sur la figure 3.4.

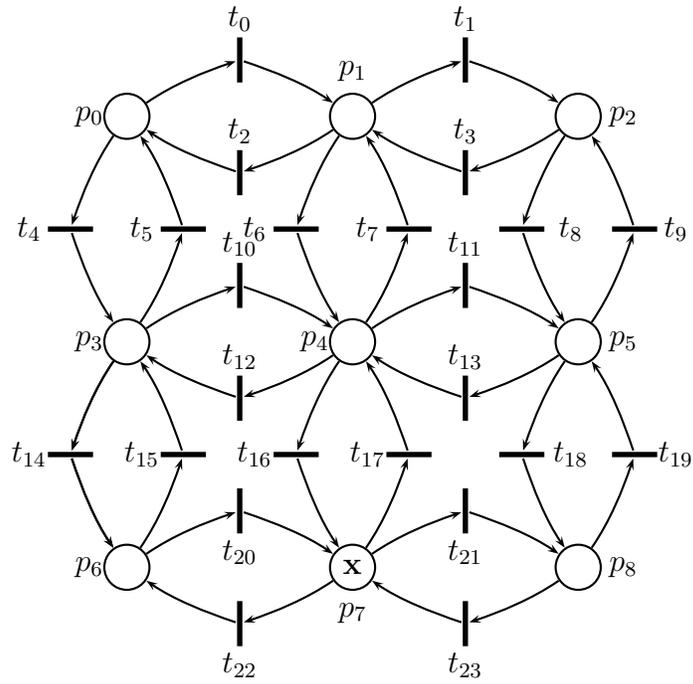


FIG. 3.3: Marquage du réseau de Petri particulière de Navigation au temps 3

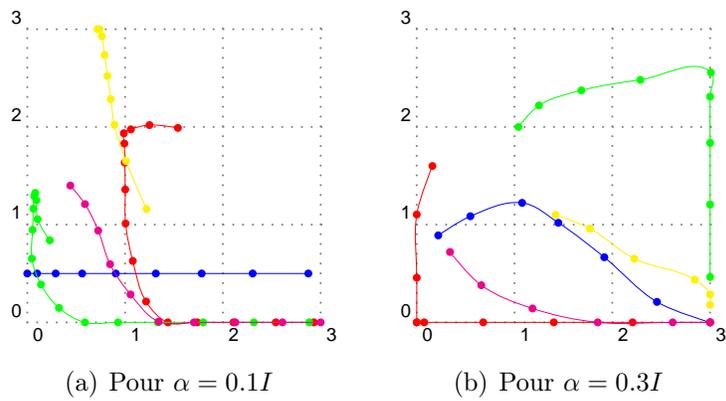


FIG. 3.4: Différentes évolutions du mobile.

2.2 Le thermostat

Le thermostat, dont le fonctionnement a été présenté au paragraphe 5.3.1 (chapitre 2), est un exemple classique de système hybride.

2.2.1 Modélisation

L'état du thermostat est décrit par un vecteur d'état contenant la température de la pièce et la température extérieure, et par un mode de fonctionnement, dépendant de la valeur de la température de la pièce, et de la valeur de marche du thermostat (*on* ou *off*). Ce thermostat est modélisé par le réseau de Petri particulière de la figure 3.5(a).

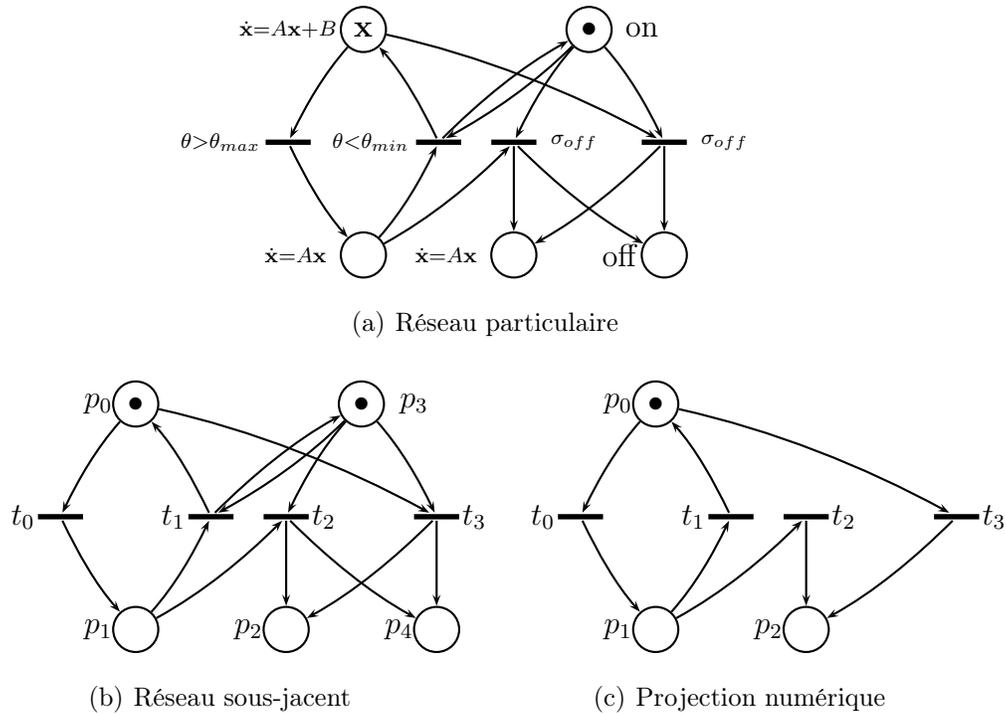


FIG. 3.5: Modélisation du thermostat

– les places du réseau sont

$$P_N = \{p_0, p_1, p_2\}, \quad P_S = \{p_3, p_4\} \quad (3.32)$$

– les transitions sont

$$T = \{t_0, t_1, t_2, t_3\} \quad (3.33)$$

– la matrice d'incidence est définie par

$$C = \begin{matrix} & t_0 & t_1 & t_2 & t_3 \\ p_0 & \left(\begin{array}{cccc} -\mathbf{x} & \mathbf{x} & 0 & -\mathbf{x} \\ \mathbf{x} & -\mathbf{x} & -\mathbf{x} & 0 \\ 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 \end{array} \right) \end{matrix} \quad (3.34)$$

– les équations $\dot{\mathbf{x}} = A\mathbf{x}$ et $\dot{\mathbf{x}} = A\mathbf{x} + B$ décrivent l'évolution de la température, avec

$$\mathbf{x} = \begin{pmatrix} \theta \\ \theta_{ext} \end{pmatrix} \in \mathbb{R}^2 \quad (3.35)$$

$$A = \begin{pmatrix} -\beta & \beta \\ 0 & 0 \end{pmatrix} \quad (3.36)$$

$$B = \begin{pmatrix} \alpha \\ 0 \end{pmatrix} \quad (3.37)$$

– les conditions associées aux transitions t_0 et t_1 définissent les changements de mode :

$$A_c(t_0) = (\mathbf{x} > \begin{pmatrix} \theta_{max} \\ -\infty \end{pmatrix}), \quad A_c(t_1) = (\mathbf{x} < \begin{pmatrix} \theta_{min} \\ +\infty \end{pmatrix}) \quad (3.38)$$

– les conditions associées aux transitions t_2 et t_3 portent sur la réception de l'événement σ_{off} :

$$A_c(t_2) = A_c(t_3) = \sigma_{off} \quad (3.39)$$

– le fonctionnement du thermostat est représenté par les configurations associées aux places du réseau :

$$\delta(p_3) = on, \quad \delta(p_4) = off \quad (3.40)$$

– le marquage initial est

$$\mathcal{M}_0 = (\mathbf{x} \ 0 \ 0 \ 1 \ 0) \quad (3.41)$$

2.2.2 Propriétés

La projection numérique (figure 3.5(c)) du réseau de Petri particulière (figure 3.5(a)) est un graphe d'état : chaque transition n'a qu'une place d'entrée et qu'une place de sortie.

Le réseau particulière est donc un **graphe d'état numérique**.

De plus, sa projection numérique n'est marquée que par un seul jeton, le réseau est donc également **numériquement sauf**.

2.2.3 Évolution de la température

Tant que l'événement σ_{off} n'est pas reçu, l'état du thermostat évolue à partir de \mathcal{M}_0 selon les équations associées aux places p_0 et p_1 . Cette évolution se fait, comme nous l'avons vu dans l'exemple de navigation, en alternant évolution isoparticule et évolution isomarquage. La figure 3.6 représente les marquages sous-jacents accessibles.

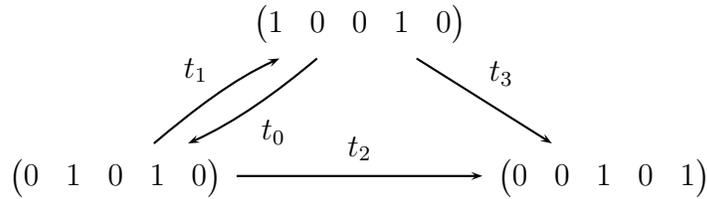


FIG. 3.6: Graphe des marquages accessibles du réseau sous-jacent (figure 3.5(b))

Dès que l'événement σ_{off} est reçu, le marquage du réseau devient $(0\ 0\ 1\ 0\ 1)$. Aucune transition n'est alors franchissable, et l'évolution se fait uniquement en suivant l'équation de la place p_2 .

Une évolution possible de la température de la pièce est présentée en figure 3.7.

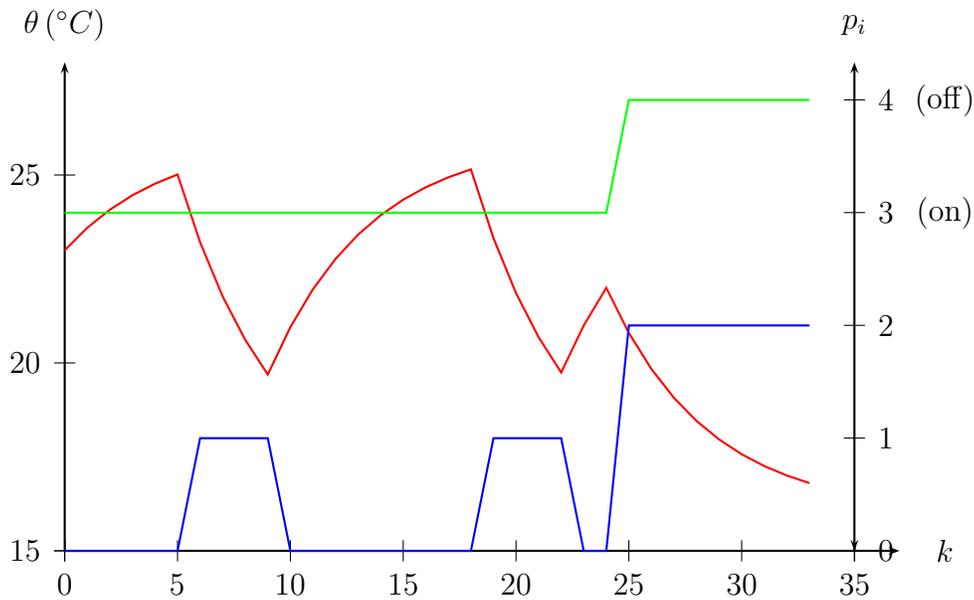


FIG. 3.7: Évolution de la température θ (en rouge) avec $\theta_{ext} = 16^\circ C$. À l'instant 25, l'événement σ_{off} est reçu. En bleu (respectivement en vert) est représentée la place numérique (respectivement symbolique) marquée.

3 Conclusion

Nous avons proposé dans ce chapitre la définition d'un formalisme pour la représentation des systèmes hybrides : *les réseaux de Petri particuliers*. Ce formalisme couple un modèle discret basé sur les réseaux de Petri à un ensemble d'équations associées à certaines places et modélisant le comportement continu du système.

L'état du système est représenté par un marquage hybride, dont la partie symbolique représente un mode de fonctionnement (décrit par des configurations) et la partie numérique est représentée par un ensemble de vecteurs d'état numériques du système.

L'évolution du marquage est également hybride : une *évolution isoparticule*, qui représente la dynamique discrète régie par la satisfaction de conditions et la réception d'événements, et une *évolution isomarquage*, qui représente la dynamique continue selon l'évolution des équations différentielles.

Nous avons également proposé des propriétés qui permettent de garantir une bonne modélisation du comportement numérique du système : les notions de *graphe d'état numérique* et de *réseau numériquement sauf*.

L'utilisation des exemples de la navigation d'un mobile et du thermostat ont également permis tout d'abord d'illustrer les définitions présentées, et également de juger de la pertinence du modèle proposé.

La problématique initiale était l'estimation des systèmes hybrides. Nous disposons désormais d'un modèle adapté à la modélisation des systèmes hybrides. L'étape suivante est donc la définition d'un principe d'estimation permettant de construire un marquage représentant au mieux l'état réel du système à partir d'observations.

Dans ce cadre, il nous a semblé judicieux d'envisager différents filtres numériques. Ainsi, le chapitre 4 présente et critique un principe d'estimation basé sur le filtre de Kalman. Puis, pour répondre aux inconvénients posés par ce dernier, nous définissons dans le chapitre 5 une méthode d'estimation basée sur le filtrage particulière, qui permettra d'utiliser au mieux le formalisme des réseaux de Petri particuliers, et d'apporter une réponse aux problèmes d'estimation et de diagnostic des systèmes hybrides.

Chapitre 4

Estimation du marquage d'un réseau de Petri particulière par filtres de Kalman

Nous avons défini au chapitre précédent le modèle des réseaux de Petri particulières pour la représentation de l'état et de la dynamique d'un système hybride. Pour pouvoir évaluer l'état d'un tel système à partir d'observations, il est nécessaire de mettre en œuvre un principe d'estimation. Dans ce chapitre nous proposons un principe d'estimation du marquage hybride par l'utilisation d'un ensemble de filtres de Kalman. Ce principe d'estimation est défini, illustré, évalué puis critiqué.

1 Représentation de l'incertitude

L'estimation du marquage d'un réseau de Petri particulière est hybride : elle porte à la fois sur le mode de fonctionnement du système (la "position" des jetons) et sur la valeur du vecteur d'état numérique. Une "bonne" estimation d'un système hybride repose essentiellement sur la modélisation qui en a été faite. Ainsi, dans la suite de ce chapitre, nous considérerons un réseau de Petri particulière qui est un *graphe d'état numérique*.

1.1 Incertitude numérique

Le filtre de Kalman repose sur certaines hypothèses : la linéarité des équations et la considération de bruits blancs gaussiens. Nous faisons donc l'hypothèse que la fonction d'évolution associée à chaque place numérique est linéaire et s'écrit :

$$\forall p \in P_N, \forall k, \mathbf{x}_{k+1} = F_k(p) \mathbf{x}_k + U_k(p) \quad (4.1)$$

La marquage d'une place numérique représente l'estimation du vecteur d'état numérique du système. Cette estimation est représentée par une distribution de probabilité sur les valeurs possibles de ce vecteur d'état.

Définition 4.1 (Marquage gaussien)

Un marquage gaussien $\mathcal{M}_G(k)$ au temps k d'un réseau de Petri particulière est une application qui à chaque place numérique p associe une variable aléatoire $\mathcal{M}_G(k, p)$ de densité de probabilité

$$\mathbf{p}(\mathcal{M}_G(k, p)) = \omega_k(p) \mathcal{N}(\hat{\mathbf{x}}_k(p), \mathbf{C}_k(p)) \quad (4.2)$$

L'équation 4.2 décrit le marquage d'une place p comme une variable aléatoire suivant une loi normale de moyenne $\hat{\mathbf{x}}_k(p)$ et de covariance $\mathbf{C}_k(p)$ pondérée par un coefficient $\omega_k(p)$. La figure 4.1 représente le marquage initial du thermostat et sa densité de probabilité.

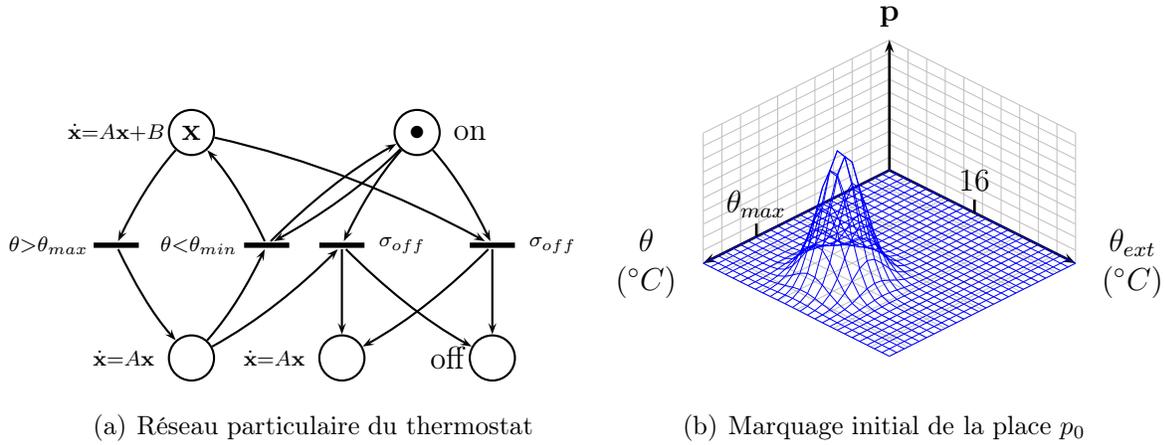


FIG. 4.1: Marquage gaussien du thermostat : le marquage de la place p_0 suit la distribution gaussienne de moyenne $\begin{pmatrix} \theta_{max} \\ 16 \end{pmatrix}$, de covariance $\begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}$ et de poids 1. Les équations linéaires associées au réseau sont $\mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{A}) \mathbf{x}_k + \mathbf{B}$ (pour p_0) et $\mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{A}) \mathbf{x}_k$ (pour p_1 et p_2).

1.2 Propriétés

Le réseau particulier est un *graphe d'état numérique*, ce qui signifie que l'ensemble des places numériques n'est marqué que par un seul vecteur, le vecteur d'état \mathbf{x} .

Dans le cadre d'un marquage gaussien, cette propriété se traduit en la définition d'une distribution de probabilité :

$$\forall k, \sum_{p \in P_N} \int_X \mathbf{p}(\mathcal{M}_G(p, k)) = 1 \quad (4.3)$$

Remarque 4.2

L'équation (4.3) est équivalente à

$$\sum_{p \in P_N} \omega_k(p) = 1 \quad (4.4)$$

En effet,

$$\begin{aligned} \forall k, \quad 1 &= \sum_{p \in P_N} \int_X \mathbf{p}(M_G(k, p)) \\ &= \sum_{p \in P_N} \omega_k(p) \int_X \mathcal{N}(\hat{\mathbf{x}}_k(p), \mathbf{C}_k(p)) \\ &= \sum_{p \in P_N} \omega_k(p) \end{aligned}$$

1.3 Incertitude symbolique

De même que l'incertitude numérique est représentée par plusieurs distributions évoluant dans diverses places numériques, l'incertitude symbolique est représentée par plusieurs jetons, correspondant aux diverses configurations possibles du système.

On construit donc un macro-marquage, comme c'est le cas dans (Cardoso *et al.*, 1999; Fanni *et al.*, 1997), qui contient un ensemble de jetons et représente les différents modes de fonctionnement possibles à un instant donné.

Remarque 4.3 (Conjonction et disjonction)

Le marquage de différentes places symboliques dans un marquage classique (définition 3.1) s'interprète comme une conjonction de configurations, par exemple "il y a un chauffage dans la pièce 1" et "dans la pièce 2". En revanche, il faut interpréter le macro-marquage comme une disjonction ("le thermostat est allumé" ou "éteint") qui représente l'incertitude sur le mode de fonctionnement du système.

1.4 Propagation de l'incertitude

L'incertitude hybride décrite ci-dessus couple donc une représentation gaussienne pour l'incertitude numérique et un macro-marquage pour l'incertitude symbolique.

Cette incertitude doit être propagée dans le réseau pour modéliser l'évolution incertaine du système. Il est donc nécessaire de définir pour l'étape de prédiction une évolution isomarquage bruitée et une évolution isoparticule bruitée.

2 Principe d'estimation

On considère un système hybride modélisé par un réseau de Petri particulière \mathcal{R} . L'estimation de l'état courant du système est modélisée par un macro-marquage gaussien au temps k , $\widehat{\mathcal{M}}(k)$.

À partir d'une observation \mathbf{z}_{k+1} au temps $k+1$ on souhaite mettre à jour l'estimation de l'état du système.

Cette estimation, reprenant le cadre bayésien (chapitre 1, paragraphe 1.2), est un processus itératif : l'étape de prédiction calcule l'état estimé suivant, et l'étape de recalage met à jour cette prédiction à partir de l'observation.

2.1 Prédiction

L'étape de prédiction consiste à calculer le marquage estimé $\widehat{\mathcal{M}}(k+1|k)$ au temps $k+1$ à partir d'un marquage $\widehat{\mathcal{M}}(k|k)$ au temps k et du modèle du système. Le marquage estimé résulte d'une évolution en deux étapes, une évolution isomarquage bruitée et une évolution isoparticule bruitée.

2.1.1 Première ébauche

Évolution isomarquage bruitée L'évolution isomarquage concerne l'évolution continue du vecteur d'état du système. Elle correspond à l'étape de prédiction du filtre de Kalman. Pour représenter l'incertitude liée à l'évolution du système et au modèle, nous introduisons un bruit de système pour chaque place, qui est un bruit blanc gaussien de covariance $Q_k(p)$ pour la place p au temps k .

Définition 4.4 (Évolution isomarquage bruitée)

L'évolution isomarquage bruitée d'un marquage $\widehat{\mathcal{M}}(k|k)$ fournit un marquage prédit $\widehat{\mathcal{M}}(k+1|k)$. Soit $p \in P_N$ une place numérique et $\mathbf{p}(\widehat{\mathcal{M}}(k|k, p)) = \omega_{k|k}(p) \mathcal{N}(\hat{\mathbf{x}}_{k|k}(p), \mathbf{C}_{k|k}(p))$. Alors

$$\begin{aligned} \mathbf{p}(\widehat{\mathcal{M}}(k+1|k, p)) &= \omega_{k+1|k}(p) \mathcal{N}(\hat{\mathbf{x}}_{k+1|k}(p), \mathbf{C}_{k+1|k}(p)), \text{ avec} & (4.5) \\ \omega_{k+1|k}(p) &= \omega_{k|k}(p) \\ \hat{\mathbf{x}}_{k+1|k}(p) &= F_k(p) \hat{\mathbf{x}}_{k|k}(p) + U_k(p) \\ \mathbf{C}_{k+1|k}(p) &= F_k(p) \mathbf{C}_{k|k}(p)^t F_k(p) + Q_k(p) \end{aligned}$$

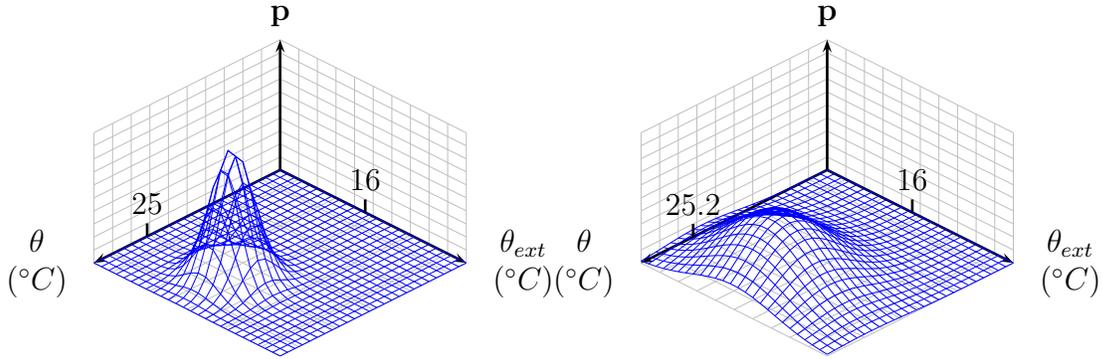
Exemple 4.5 *Considérons le réseau de Petri particulière de la figure 4.1 et son marquage gaussien au temps 0. Son évolution isomarquage fournit le marquage $\widehat{\mathcal{M}}(1|0)$ (figure 4.2), défini par :*

$$\mathbf{p}(\widehat{\mathcal{M}}(1|0, p_0)) = \mathcal{N}(\widehat{\mathbf{x}}_{1|0}, \mathbf{C}_{1|0}), \text{ avec} \quad (4.6)$$

$$\widehat{\mathbf{x}}_{1|0} = (I + A)\widehat{\mathbf{x}}_0 + B = \begin{pmatrix} 25.2 \\ 16 \end{pmatrix}$$

$$\mathbf{C}_{1|0} = (I + A)\mathbf{C}_0^t(I + A) + Q_k = \begin{pmatrix} 0.48 & 0.06 \\ 0.06 & 0.6 \end{pmatrix}$$

avec $\alpha = 2$, $\beta = 0.2$, $\theta_{max} = 25$, $Q_k = \begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}$.



(a) Densité du marquage au temps 0

(b) Densité du marquage au temps 1

FIG. 4.2: Évolution isomarquage du marquage $\widehat{\mathcal{M}}(0|0)$

Pseudo-tir des transitions L'évolution isoparticule représente les changements de modes de fonctionnement du système. La prédiction de ces changements de modes consiste à envisager les différents modes possibles en fonction de la distribution de probabilité sur le vecteur d'état (marquage gaussien) et de l'éventualité de l'arrivée d'événements. Cette prédiction est réalisée au moyen d'un **pseudo-tir** des transitions.

Définition 4.6 (Transition pseudo-franchissable)

Soit \mathcal{R} un réseau particulière et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé au temps k . Une transition t est pseudo-franchissable avec une probabilité $\mathbf{P}(t)$, ou **$\mathbf{P}(t)$ -franchissable** si

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}(k|k, p) \geq Pre(p, t) \quad (4.7)$$

$$\forall p \in P_N \cap \bullet t, \quad \mathbf{P}(t) = \mathbf{P}(A_c(t)(\widehat{\mathcal{M}}(k|k, p))) > 0 \quad (4.8)$$

Une transition t est pseudo-franchissable si les places symboliques amont contiennent assez de jetons, et si la probabilité de satisfaire sa condition est non nulle.

Définition 4.7 (Pseudo-tir (version 1))

Le pseudo-tir de t fournit alors un marquage $\widehat{\mathcal{M}}'(k|k)$, défini par

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}'(k|k, p) = \begin{cases} \widehat{\mathcal{M}}(k|k, p) + C(p, t) & \text{si } p \in P_S \cap t^\bullet \\ \widehat{\mathcal{M}}(k|k, p) & \text{sinon} \end{cases} \quad (4.9)$$

$$\forall p \in P_N, \quad \mathbf{p}(\widehat{\mathcal{M}}'(k|k, p)) = \begin{cases} \omega'_{k|k}(p) \mathcal{N}(\hat{\mathbf{x}}_{k|k}(p), \mathbf{C}_{k|k}(p)) & \text{si } p \in \bullet t \\ \omega'_{k|k}(p) \mathcal{N}(\hat{\mathbf{x}}'_{k|k}(p), \mathbf{C}'_{k|k}(p)) & \text{si } p \in t^\bullet \\ \mathbf{p}(\widehat{\mathcal{M}}(k|k, p)) & \text{sinon} \end{cases} \quad (4.10)$$

$$\text{avec} \begin{cases} \omega'_{k|k}(p) = \omega_{k|k}(p) - \mathbf{P}(t) \\ \omega'_{k|k}(q) = \omega_{k|k}(q) + \mathbf{P}(t) \\ \omega'_{k|k}(q) \hat{\mathbf{x}}'_{k|k}(q) = \omega_{k|k}(q) \hat{\mathbf{x}}_{k|k}(q) + \mathbf{P}(t) \hat{\mathbf{x}}_{k|k}(p) \\ \omega'_{k|k}(q) \mathbf{C}'_{k|k}(q) = \omega_{k|k}(q) \mathbf{C}_{k|k}(q) + \mathbf{P}(t) \mathbf{C}_{k|k}(p) \end{cases}, \quad \begin{matrix} p \in P_N \cap \bullet t \\ q \in P_N \cap t^\bullet \end{matrix}$$

Le pseudo-tir de t :

- marque les places symboliques aval ;
- réduit le poids de la place numérique amont ;
- fusionne la distribution de la place aval avec celle de la place amont.

En particulier, ce pseudo-tir ne modifie pas le marquage des places symboliques amont : le marquage résultant est donc un macro-marquage, qui contient les configurations du système avant et après franchissement.

Remarque 4.8

Le pseudo-tir de t conserve la propriété de l'équation (4.3). En effet,

$$\begin{aligned} \sum_{i \in P_N} \omega'_{k|k}(i) &= \omega'_{k|k}(p) + \omega'_{k|k}(q) + \sum_{i \in P_N \setminus \{p, q\}} \omega'_{k|k}(i) \\ &= (\omega_{k|k}(p) - \mathbf{P}(t)) + (\omega_{k|k}(q) + \mathbf{P}(t)) + \sum_{i \in P_N \setminus \{p, q\}} \omega_{k|k}(i) \\ &= \omega_{k|k}(p) + \omega_{k|k}(q) + \sum_{i \in P_N \setminus \{p, q\}} \omega_{k|k}(i) \\ &= \sum_{i \in P_N} \omega_{k|k}(i) \\ &= 1 \end{aligned}$$

Remarque 4.9 (Place numérique aval vide)

Dans le cas où la place numérique aval q de t est vide, la distribution du marquage résultant du pseudo-tir de t s'écrit :

$$\mathbf{p}(\widehat{\mathcal{M}}'(k|k, q)) = \frac{\mathbf{P}(t)}{\omega_{k|k}(p)} \mathbf{p}(\widehat{\mathcal{M}}(k|k, p)) \text{ où } p \in P_N \cap \bullet t \quad (4.11)$$

Ce marquage correspond à la distribution de la place amont, pondérée par la probabilité de tir de la transition.

Exemple 4.10 Reprenons le réseau de Petri de la figure 4.1.

$$\mathbf{P}(t_0) = \mathbf{P}(\theta > \theta_{max}) \quad (4.12)$$

$$= \int_{] \theta_{max}, +\infty[} \mathbf{p}(\widehat{\mathcal{M}}'(0|0, p_0)) d\theta \quad (4.13)$$

$$= \int_{] \theta_{max}, +\infty[} \mathcal{N}((\theta_{max}, 16), 0.3I_2) d\theta \quad (4.14)$$

$$= \frac{1}{2} \quad (4.15)$$

La transition t_0 est donc $\frac{1}{2}$ -franchissable. Son pseudo-tir fournit le marquage de la figure 4.3

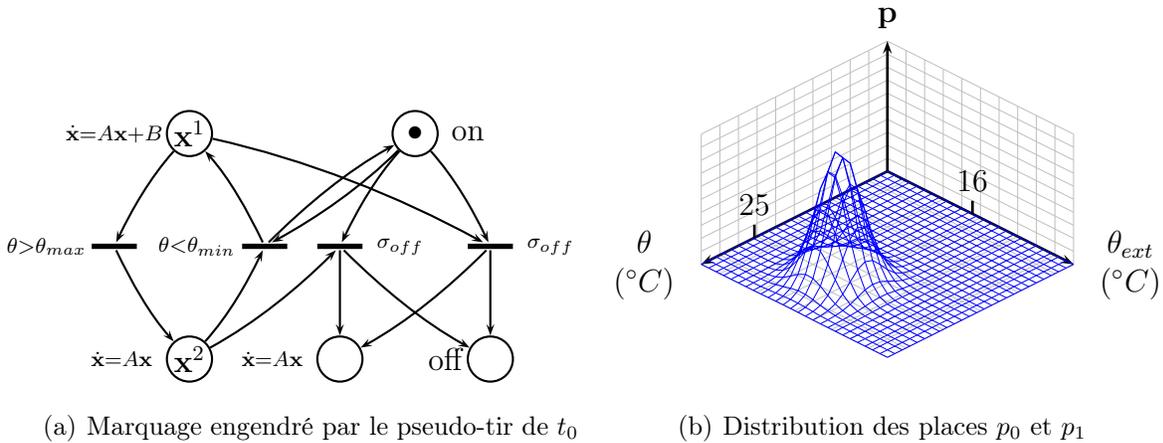


FIG. 4.3: Pseudo-tir de t_0 : les marquages des places p_0 et p_1 suivent la même distribution 4.3(b), avec un poids de $\frac{1}{2}$.

Remarque 4.11 (Transition 1-franchissable)

Une transition est 1-franchissable lorsque sa condition est satisfaite. Le marquage résultant de son pseudo-tir consiste alors à supprimer la distribution amont ($\omega'_{k|k}(p) = 0$), et l'ajouter à la place aval. Le pseudo-tir de t (qui est 1-franchissable) ne correspond pas à un tir classique : les jetons des places symboliques amont ne sont pas supprimés.

Remarque 4.12 (Événements)

Certaines transitions du réseau peuvent être conditionnées par la réception d'événements. C'est le cas par exemple des transitions t_2 et t_3 du réseau du thermostat (figure 4.1(a)) qui doivent recevoir l'événement σ_{off} pour être franchissables.

En prédiction, l'évolution isoparticule consiste à construire tous les modes de fonctionnements possibles à l'instant suivant. L'arrivée de l'événement est alors possible (mais pas certaine). Pour représenter cette incertitude sur l'arrivée des événements, on définit la pseudo-sensibilisation d'une transition t par :

- si $A_c(t)$ ne dépend pas de l'arrivée d'événement, t est pseudo-franchissable si elle respecte la définition 4.6 ;
- si $A_c(t)$ dépend de l'arrivée d'un événement σ , t est $\mathbf{P}(t)$ -franchissable, avec $\mathbf{P}(t) = \mathbf{P}(\sigma) \mathbf{P}$, \mathbf{P} étant la probabilité définie dans 4.6.

$\mathbf{P}(\sigma)$ représente la probabilité d'arrivée de l'événement σ . Si on ne connaît pas cette probabilité, il est néanmoins nécessaire de représenter cette ignorance par une valeur. On dira alors que l'arrivée et l'absence de σ sont équiprobables, et donc $\mathbf{P}(\sigma) = \frac{1}{2}$.

Cette modélisation n'est toutefois pas satisfaisante : la valeur attribuée aux événements est empirique, et aucune valeur de probabilité ne peut représenter l'ignorance sur l'arrivée de l'événement (Chachoua et Pacholczyk, 2000).

Exemple 4.13 Reprenons le réseau de Petri de la figure 4.1. La transition t_3 est donc $\frac{1}{2}$ -franchissable, puisque sa condition ne dépend que de l'arrivée de l'événement σ_{off} . Son pseudo-tir fournit le marquage de la figure 4.4.

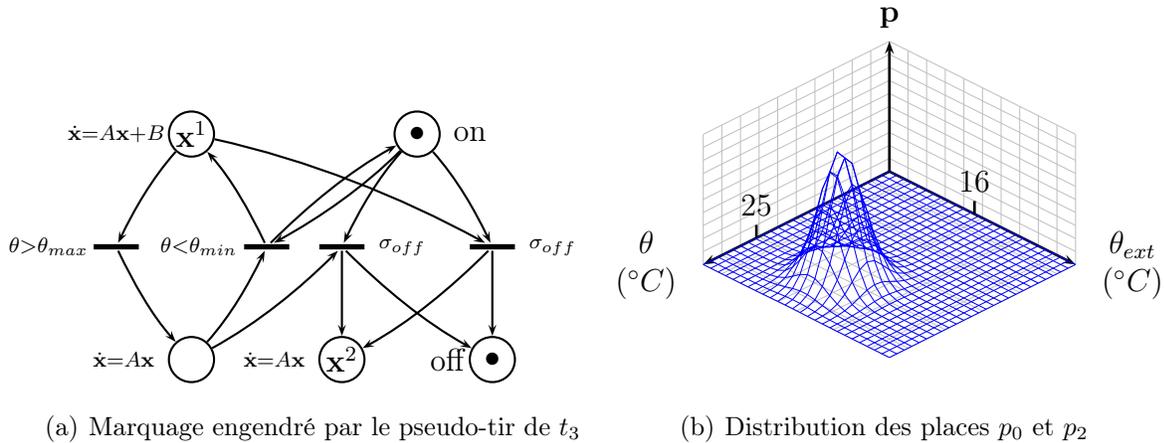


FIG. 4.4: Pseudo-tir de t_3 : les marquages des places p_0 et p_2 suivent la même distribution 4.4(b), avec un poids de $\frac{1}{2}$, et les configurations *on* et *off* sont marquées.

Le marquage prédit indique donc que le thermostat peut être dans les configurations *on* ou *off* et qu'en conséquence, le vecteur d'état, représenté par une gaussienne, peut évoluer dans les places p_0 ou p_2 .

Évolution isoparticule bruitée Nous avons défini le pseudo-tir d'une transition, qui permet de propager l'incertitude en modifiant les poids et les distributions des places numériques, et en modifiant le marquage des places symboliques aval. Ce pseudo-tir permet ainsi de prédire les changements de mode éventuels.

Dans un cas certain, l'évolution isoparticule consiste à tirer toutes les transitions qui sont franchissables, jusqu'à bloquer le réseau (algorithme 3.1).

De même, l'évolution isoparticule bruitée consiste en la prédiction de tous les changements de modes possibles. Il est donc nécessaire, pour assurer une bonne prédiction de l'état du système, de pseudo-tirer toutes les transitions du réseau qui sont pseudo-franchissables.

Or, la définition du pseudo-tir d'une transition (définition 4.6) est incompatible avec cette évolution isoparticule. En effet, considérons le réseau de la figure 4.3. Le marquage de ce réseau est issu du pseudo-tir de la transition t_0 . Or, dans ce nouveau marquage, t_0 est toujours pseudo-franchissable :

$$\mathbf{P}(t_0) = \omega(p_0)\mathbf{P}(\theta > \theta_{max}) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \quad (4.16)$$

Le pseudo-tir de t_0 va engendrer un marquage similaire à celui de la figure 4.3, avec les poids $\omega(p_0) = \frac{1}{4}$ et $\omega(p_1) = \frac{3}{4}$. Puis t_0 est $\frac{1}{8}$ -franchissable...

On voit donc clairement qu'en appliquant le pseudo-tir de la définition 4.6 l'évolution isoparticule est infinie puisqu'une transition \mathbf{P} -franchissable, avec $\mathbf{P} < 1$, restera pseudo-franchissable pour tout marquage accessible.

2.1.2 Améliorations

Redéfinissons donc la partie du pseudo-tir d'une transition qui modifie le marquage des places numériques :

Définition 4.14 (Pseudo-tir (version 2))

Soit \mathcal{R} un réseau particulière et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé au temps k . Une transition t est pseudo-franchissable avec une probabilité $\mathbf{P}(t)$, ou $\mathbf{P}(t)$ -franchissable si

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}(k|k, p) \geq Pre(p, t) \quad (4.17)$$

$$\mathbf{P}(t) = \mathbf{P}(A_c(t)(\widehat{\mathcal{M}}(k|k, p))) > 0, \quad \text{où } p = P_N \cap \bullet t \quad (4.18)$$

Le **pseudo-tir** de t fournit alors un marquage $\widehat{\mathcal{M}}'(k|k)$, défini par

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}'(k|k, p) = \begin{cases} \widehat{\mathcal{M}}(k|k, p) + C(p, t) & \text{si } p \in P_S \cap t^\bullet \\ \widehat{\mathcal{M}}(k|k, p) & \text{sinon} \end{cases} \quad (4.19)$$

$$\forall p \in P_N \cap t^\bullet, \quad \mathbf{p}(\widehat{\mathcal{M}}'(k|k, p))(x)^1 = \begin{cases} \mathbf{p}(\widehat{\mathcal{M}}(k|k, p))(x) & \text{si } \overline{A_c(t)}(x) \\ 0 & \text{sinon} \end{cases} \quad (4.20)$$

$$\forall q \in P_N \cap t^\bullet, \quad \mathbf{p}(\widehat{\mathcal{M}}'(k|k, q))(x) = \mathbf{p}(\widehat{\mathcal{M}}(k|k, q))(x) + \begin{cases} \mathbf{p}(\widehat{\mathcal{M}}(k|k, p))(x) & \text{si } A_c(t)(x) \\ 0 & \text{sinon} \end{cases} \quad (4.21)$$

La distribution sur le marquage de la place numérique amont est tronquée : la probabilité $\mathbf{p}(\widehat{\mathcal{M}}'(k|k, p))$ correspond à la probabilité conditionnelle $\mathbf{p}(\widehat{\mathcal{M}}(k|k, p) = x \mid \overline{A_c(t)}(x))$. Il ne reste donc plus dans la place amont que le “morceau” de distribution qui correspond aux valeurs qui ne satisfont pas la transition.

Ce pseudo-tir est beaucoup plus satisfaisant que le précédent : comme la distribution est tronquée, la transition t n’est plus franchissable dans le marquage $\widehat{\mathcal{M}}'(k|k)$, puisque $\mathbf{P}(t) = 0$.

Remarque 4.15

Vérifions la propriété de l’équation (4.3) pour le marquage $\widehat{\mathcal{M}}'(k|k)$:

$$\begin{aligned} \sum_{i \in P_N} \int_X \mathbf{p}(\widehat{\mathcal{M}}'(k|k, i)) &= \sum_{i \in P_N \setminus \{p, q\}} \int_X \mathbf{p}(\widehat{\mathcal{M}}'(k|k, i)) + \int_X \mathbf{p}(\widehat{\mathcal{M}}'(k|k, p)) + \int_X \mathbf{p}(\widehat{\mathcal{M}}'(k|k, q)) \\ &= \sum_{i \in P_N \setminus \{p, q\}} \int_X \mathbf{p}(\widehat{\mathcal{M}}(k|k, i)) + \int_{\overline{A_c(t)}} \mathbf{p}(\widehat{\mathcal{M}}(k|k, p)) \\ &\quad + \int_{A_c(t)} \mathbf{p}(\widehat{\mathcal{M}}(k|k, p)) + \int_X \mathbf{p}(\widehat{\mathcal{M}}(k|k, q)) \\ &= \sum_{i \in P_N \setminus \{p, q\}} \omega_{k|k}(i) + \int_X \mathbf{p}(\widehat{\mathcal{M}}(k|k, p)) + \omega_{k|k}(q) \\ &= \sum_{i \in P_N \setminus \{p\}} \omega_{k|k}(i) + \omega_{k|k}(p) \\ &= 1 \end{aligned}$$

¹ $\mathbf{p}(\widehat{\mathcal{M}}(k|k, p))$ est la densité de probabilité du marquage estimé de p au temps k . C’est donc une fonction de $X \subset \mathbb{R}^n$ dans $[0, 1]$.

La nouvelle définition du pseudo-tir d'une transition permet donc de résoudre le problème de franchissement infini, mais apporte également plus de rigueur à l'évolution iso-particule : les densités de probabilité résultantes sont mathématiquement fondées puisqu'elles correspondent aux probabilités conditionnelles dépendant de la condition associée à la transition tirée.

2.1.3 Distributions tronquées

Ce pseudo-tir remet cependant en cause le fait que chaque marquage gaussien suit une loi de probabilité normale, puisque les distributions résultantes sont tronquées. Ces distributions ne sont plus gaussiennes. Cela remet également en cause l'utilisation de filtres de Kalman, dans lesquels la représentation de l'état du système doit être gaussienne. Nous définissons alors :

Définition 4.16 (Marquage gaussien)

Soit p une place numérique. Le marquage gaussien $\mathcal{M}_G(k, p)$ de cette place au temps k est une variable aléatoire de densité de probabilité :

$$\mathbf{p}(\mathcal{M}_G(k, p))(x) = \begin{cases} \omega(k, p) \mathcal{N}(\hat{\mathbf{x}}_k(p), \mathbf{C}_k(p), x) & \text{si } x \in D_k(p) \\ 0 & \text{sinon} \end{cases} \quad (4.22)$$

Cette densité est une loi normale tronquée. L'ensemble $D_k(p)$ est le domaine de p au temps k , dans lequel le marquage de p peut prendre ses valeurs. Le marquage gaussien d'une place p au temps k est alors caractérisé par le quadruplet $(\omega_k(p), \hat{\mathbf{x}}_k(p), \mathbf{C}_k(p), D_k(p))$.

Définition 4.17 (Pseudo-tir (version 3))

Soit \mathcal{R} un réseau particulière et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé au temps k . Une transition t est pseudo-franchissable avec une probabilité $\mathbf{P}(t)$, ou **$\mathbf{P}(t)$ -franchissable** si

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}(k|k, p) \geq \text{Pre}(p, t) \quad (4.23)$$

$$\mathbf{P}(t) = \mathbf{P}(A_c(t)(\widehat{\mathcal{M}}(k|k, p))) > 0, \quad \text{où } p \in P_N \cap \bullet t \quad (4.24)$$

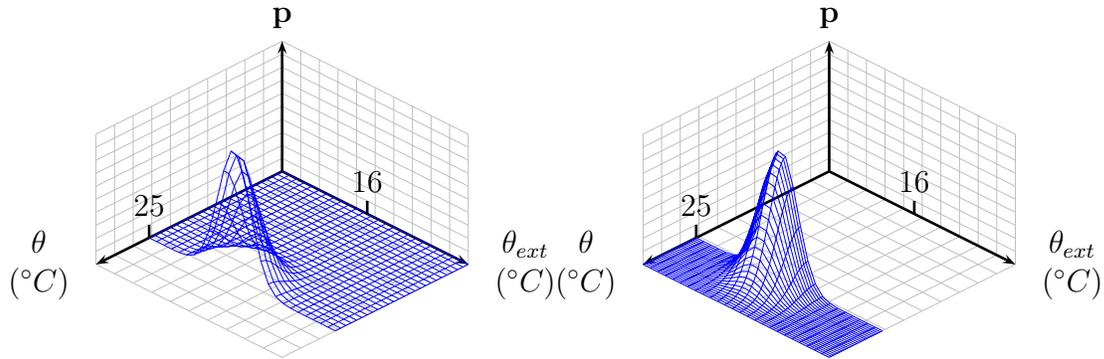
Le **pseudo-tir** de t fournit alors un marquage $\widehat{\mathcal{M}}'(k|k)$, défini par

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}'(k|k, p) = \begin{cases} \widehat{\mathcal{M}}(k|k, p) + C(p, t) & \text{si } p \in P_S \cap t^\bullet \\ \widehat{\mathcal{M}}(k|k, p) & \text{sinon} \end{cases} \quad (4.25)$$

$$\forall p \in P_N \cap \bullet t, \quad \begin{cases} \omega'_k(p) = \omega(k, p) \\ \hat{\mathbf{x}}'_k(p) = \hat{\mathbf{x}}_k(p) \\ \mathbf{C}'_k(p) = \mathbf{C}_k(p) \\ D'_k(p) = D_k(p) \cap \overline{A_c(t)} \end{cases} \quad (4.26)$$

$$\forall q \in P_N \cap t^\bullet, \quad \begin{cases} \omega'_k(q) = \omega(k, q) \\ \omega'_k(q) \hat{\mathbf{x}}'_k(q) = \omega(k, q) \hat{\mathbf{x}}_k(q) + \omega(k, q) \hat{\mathbf{x}}_k(p) \\ \omega'_k(q) \mathbf{C}'_k(q) = \omega(k, q) \mathbf{C}_k(q) + \omega(k, q) \mathbf{C}_k(p) \\ D'_k(q) = D_k(q) \cup (A_c(t) \cap D_k(p)) \end{cases} \quad (4.27)$$

Exemple 4.18 Reprenons l'exemple 4.10. Le pseudo-tir de t_0 fournit le marquage de la figure 4.3(a). Les densités des variables x^1 et x^2 marquant les places p_0 et p_1 sont représentées dans la figure 4.5.



(a) Densité de la variable x^1 marquant p_0 (b) Densité de la variable x^2 marquant p_1

FIG. 4.5: Pseudo-tir de la transition t_0

Définition 4.19 (Évolution isomarquage bruitée)

L'évolution isomarquage bruitée d'un marquage $\widehat{\mathcal{M}}(k, p)$ d'une place numérique p au

temps k fournit un marquage prédit $\widehat{\mathcal{M}}(k+1|k, p)$, caractérisé par :

$$\omega_{k+1|k}(p) = \omega_k(p) \quad (4.28)$$

$$\hat{\mathbf{x}}_{k+1|k}(p) = F_k(p) \hat{\mathbf{x}}_k(p) + U_k(p) \quad (4.29)$$

$$\mathbf{C}_{k+1|k}(p) = F_k(p) \mathbf{C}_k(p) {}^t F_k(p) + Q_k(p) \quad (4.30)$$

$$D_{k+1|k}(p) = F_k(p) D_k(p) + U_k(p) \quad (4.31)$$

Remarque 4.20

Le calcul de l'ensemble $D_{k+1|k}$ peut ne pas être simple, même si D_k est un domaine polytopique et que F_k est linéaire.

L'évolution continue du marquage d'une place numérique nécessite donc d'enrichir les équations de prédiction du filtre de Kalman avec la prédiction du domaine de la place p : le domaine prédit au temps $k+1$ correspond à l'image du domaine au temps k par l'équation associée à p .

2.1.4 Prédiction de la température

Considérons le réseau de Petri particulière de la figure 4.1, représentant le fonctionnement du thermostat ainsi que la densité de probabilité initiale de la température.

Nous procédons tout d'abord à une évolution isoparticule : cette évolution, comme nous l'avons vu, simule les changements de modes du système. Pour le marquage de la figure 4.1 :

- t_0 est $\frac{1}{2}$ -franchissable, car la variable \mathbf{x} vérifie la condition $\theta > \theta_{max}$ avec la probabilité $\frac{1}{2}$, et le poids de p_0 est $\omega_0(p_0) = 1$;
- t_1 et t_2 ne sont pas franchissables, puisque la place p_1 est vide ;
- t_3 est $\frac{1}{2}$ -franchissable, car ses places amont sont marquées, et son franchissement dépend de l'événement σ_{off} .

On a dans ce cas un conflit entre les transitions t_0 et t_3 . Le franchissement de t_0 fournit le marquage de la figure 4.3(a), dont les densités des places p_0 et p_1 sont les densités tronquées (figure 4.5). Le franchissement de t_3 fournit le marquage de la figure 4.4, où les places p_0 et p_2 ont la même densité de probabilité.

La figure 4.6 est une représentation du graphe des marquages accessibles du réseau du thermostat. Dans ce graphe, la variable \mathbf{x} suit la densité de la figure 4.7(d), les variables \mathbf{x}^1 et \mathbf{x}^2 suivent les lois tronquées (figures 4.7(b) et 4.7(c)), et la notation $\frac{1}{2} \cdot \mathbf{x}^i$ signifie que le marquage suit la même distribution que \mathbf{x}^i avec un poids $\frac{1}{2}$.

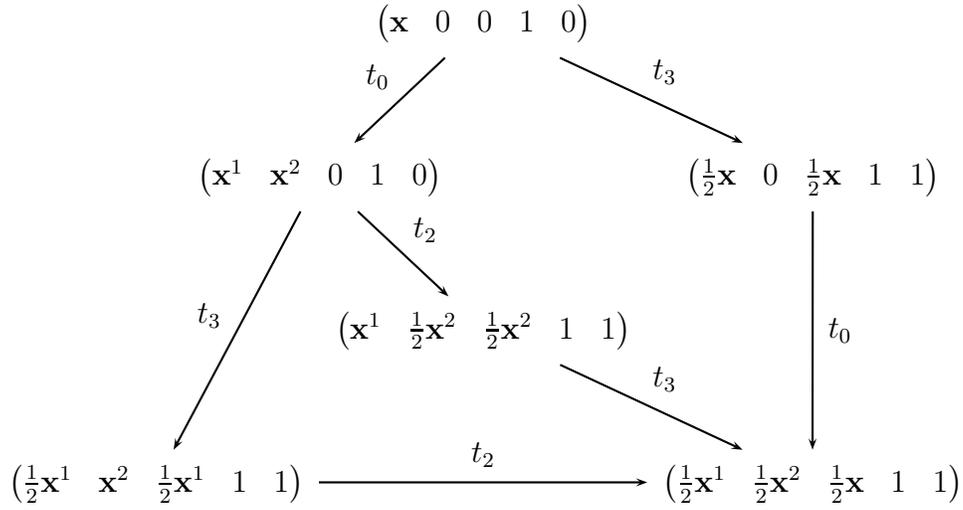


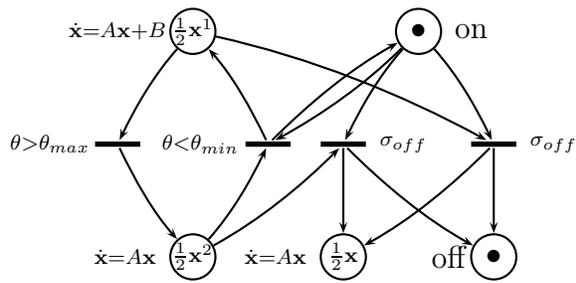
FIG. 4.6: Marquages accessibles du réseau du thermostat à partir du marquage estimé $\widehat{\mathcal{M}}(0|0)$

Remarque 4.21

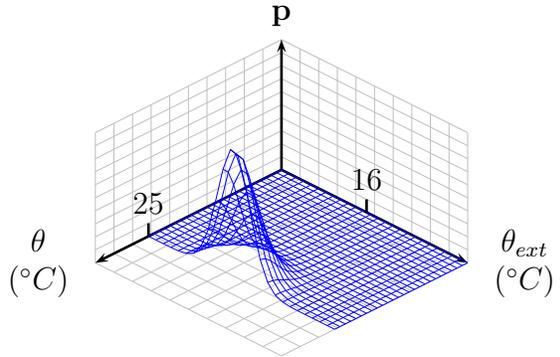
Lors de l'évolution isoparticule, l'arrivée des événements est simulée pour prédire les différentes configurations possibles. Cette simulation est donc faite une seule fois : dans chaque branche du graphe des marquages, les transitions t_2 et t_3 ne peuvent être tirées qu'une seule fois.

L'évolution isoparticule amène donc au marquage de la figure 4.7, qui est terminal.

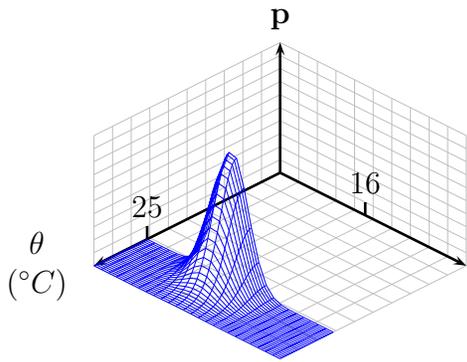
L'évolution isomarquage prédit la densité des variables de l'instant 0 à l'instant 1 (figure 4.8).



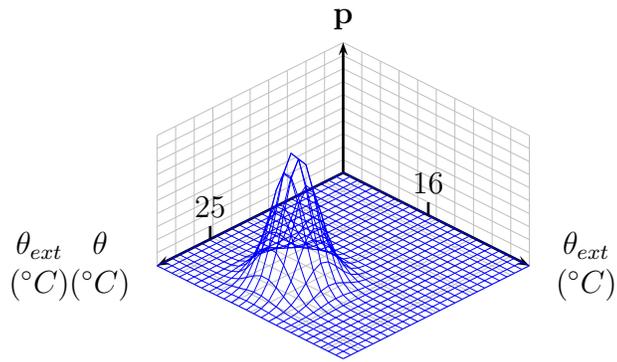
(a) Marquage



(b) Densité de la variable x^1 marquant p_0



(c) Densité de la variable x^2 marquant p_1



(d) Densité de la variable x marquant p_2

FIG. 4.7: Marquage final $\widehat{\mathcal{M}}'(0|0)$ de l'évolution isoparticule.

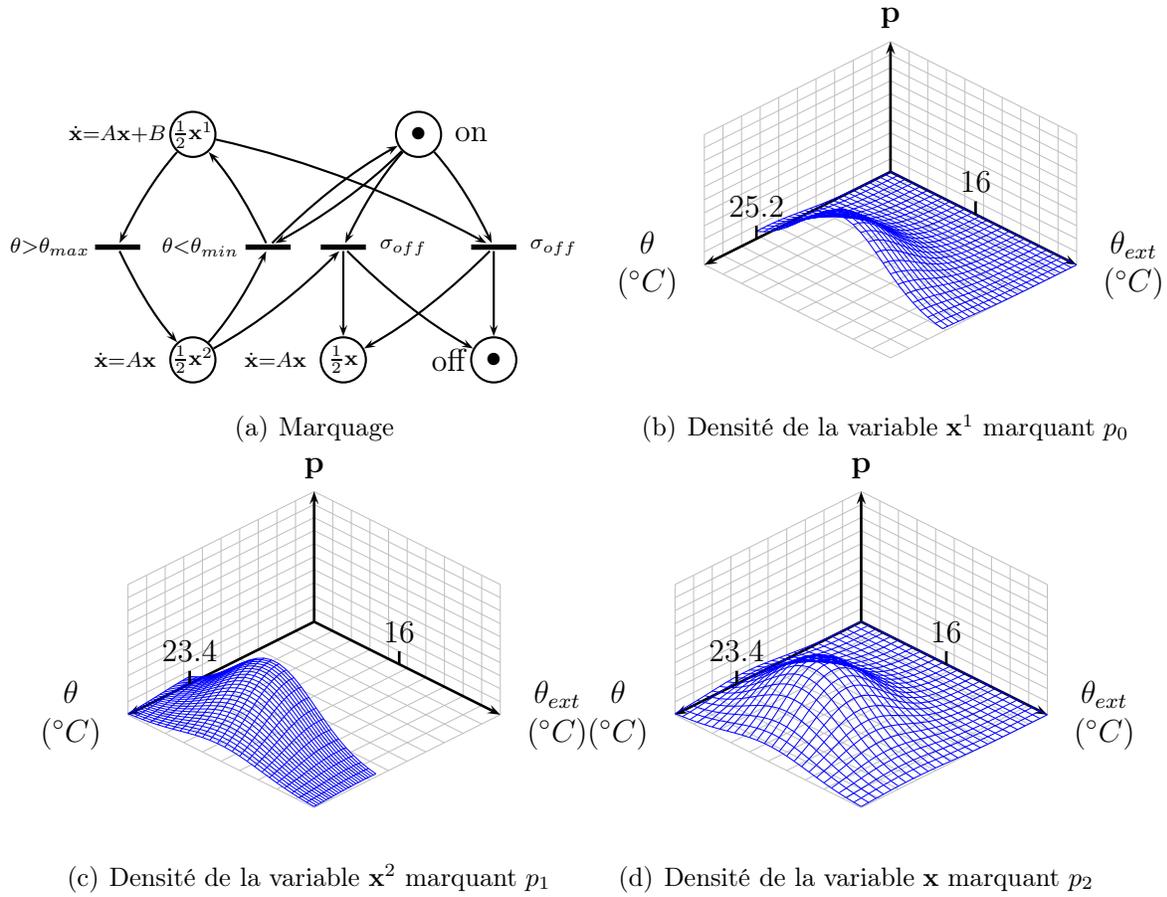


FIG. 4.8: Calcul du marquage $\widehat{\mathcal{M}}(1|0)$ à l'instant 1 à partir du marquage $\widehat{\mathcal{M}}'(0|0)$.

2.2 Recalage

L'étape de prédiction fournit un marquage $\widehat{\mathcal{M}}(k+1|k)$ représentant l'ensemble des situations possibles à l'instant $k+1$. L'étape de recalage a pour but de mettre à jour cette estimation à partir d'une observation du système au temps $k+1$. Cette observation est hybride : elle contient l'observation des paramètres numériques \mathbf{z}_{k+1} et l'observation du mode de fonctionnement du système $\boldsymbol{\mu}_{k+1}$.

2.2.1 Recalage numérique

Le recalage du marquage des places numériques suit le processus de recalage du filtre de Kalman. À chaque place numérique p est associée une équation d'observation, qui relie l'état du système à l'observation :

$$\mathbf{z}_{k+1} = H_{k+1}(p) \mathbf{x}_{k+1} + v_{k+1}(p) \quad (4.32)$$

où $H_{k+1}(p)$ est la fonction d'observation de la place p au temps $k+1$ et $v_{k+1}(p)$ le bruit d'observation associé à la place p au temps $k+1$.

Définition 4.22 (Recalage numérique)

Le recalage numérique du marquage $\widehat{\mathcal{M}}(k+1|k)$ de la place p au temps $k+1$ fournit un marquage $\widehat{\mathcal{M}}(k+1|k+1, p)$, défini par :

$$\omega_{k+1|k+1}(p) = \omega_{k+1|k}(p) \quad (4.33)$$

$$\hat{\mathbf{x}}_{k+1|k+1}(p) = \hat{\mathbf{x}}_{k+1|k}(p) + K_{k+1} (\mathbf{z}_{k+1} - H_{k+1}(p) \hat{\mathbf{x}}_{k+1|k}(p)) \quad (4.34)$$

$$\mathbf{C}_{k+1|k+1}(p) = (I - K_{k+1} H_{k+1}(p)) \mathbf{C}_{k+1|k}(p) \quad (4.35)$$

$$D_{k+1|k+1}(p) = D_{k+1|k}(p) \quad (4.36)$$

avec

$$K_{k+1} = \mathbf{C}_{k+1|k} {}^t H_{k+1}(p) (H_{k+1}(p) \mathbf{C}_{k+1|k}(p) {}^t H_{k+1}(p) + R_{k+1}(p))^{-1} \quad (4.37)$$

où $R_{k+1}(p)$ est la covariance du bruit d'observation $v_{k+1}(p)$ associé à la place p .

Le recalage numérique met à jour la partie gaussienne de la distribution, mais ne modifie ni le poids ni le domaine de la place.

Exemple 4.23 (Recalage de la température) *La prédiction du fonctionnement du thermostat a fourni le marquage de la figure 4.8. L'observation n'est faite que sur la température de la pièce. La température extérieure n'est pas observée. Toutes les places numériques*

ont la même équation d'observation, avec

$$H_k = (1 \ 0), \quad R_k = 0.3 \quad (4.38)$$

L'observation au temps 1 est $\mathbf{z}_1 = 23$. Le recalage numérique du marquage $\widehat{\mathcal{M}}(1|0)$ est :

$$\forall p, K_1(p) = \frac{1}{0.78} \begin{pmatrix} 0.48 \\ 0.06 \end{pmatrix} \quad (4.39)$$

$$\forall p, \mathbf{C}_{1|1}(p) = \frac{1}{78} \begin{pmatrix} 14.4 & 1.8 \\ 1.8 & 18 \end{pmatrix} \simeq \begin{pmatrix} 0.18 & 0.02 \\ 0.02 & 0.23 \end{pmatrix} \quad (4.40)$$

$$\hat{\mathbf{x}}_{1|1}(p_0) = \frac{1}{78} \begin{pmatrix} 1964.25 \\ 1247.83 \end{pmatrix} \simeq \begin{pmatrix} 23.85 \\ 15.83 \end{pmatrix} \quad (4.41)$$

$$\hat{\mathbf{x}}_{1|1}(p_1) = \hat{\mathbf{x}}_{1|1}(p_2) = \frac{1}{78} \begin{pmatrix} 1806 \\ 1245.6 \end{pmatrix} \simeq \begin{pmatrix} 23.15 \\ 15.97 \end{pmatrix} \quad (4.42)$$

$$(4.43)$$

Le marquage recalé $\widehat{\mathcal{M}}(1|1)$ est illustré par la figure 4.9.

2.2.2 Recalage symbolique

L'observation du mode du système permet de mettre en œuvre un principe de recalage symbolique, qui met à jour le marquage symbolique du réseau à partir de cette observation. On compare le mode observé $\boldsymbol{\mu}_{k+1}$ au temps $k + 1$ avec l'ensemble des configurations du marquage prédit $\widehat{\mathcal{M}}(k + 1|k)$.

Définition 4.24 (Recalage symbolique)

Le recalage symbolique du marquage prédit $\widehat{\mathcal{M}}(k + 1|k)$ à partir de l'observation $\boldsymbol{\mu}_{k+1}$ fournit un marquage recalé $\widehat{\mathcal{M}}(k + 1|k + 1)$, défini par :

$$\forall p \in P_S, \widehat{\mathcal{M}}(k + 1|k + 1, p) = \begin{cases} \widehat{\mathcal{M}}(k + 1|k, p) & \text{si } \boldsymbol{\mu}_{k+1} \in \delta(p) \\ 0 & \text{sinon} \end{cases} \quad (4.44)$$

Exemple 4.25 (Recalage des modes du thermostat) *Considérons le marquage numériquement recalé de la figure 4.9 et l'observation $\boldsymbol{\mu}_1 = on$. Le marquage recalé est alors celui de la figure 4.10.*

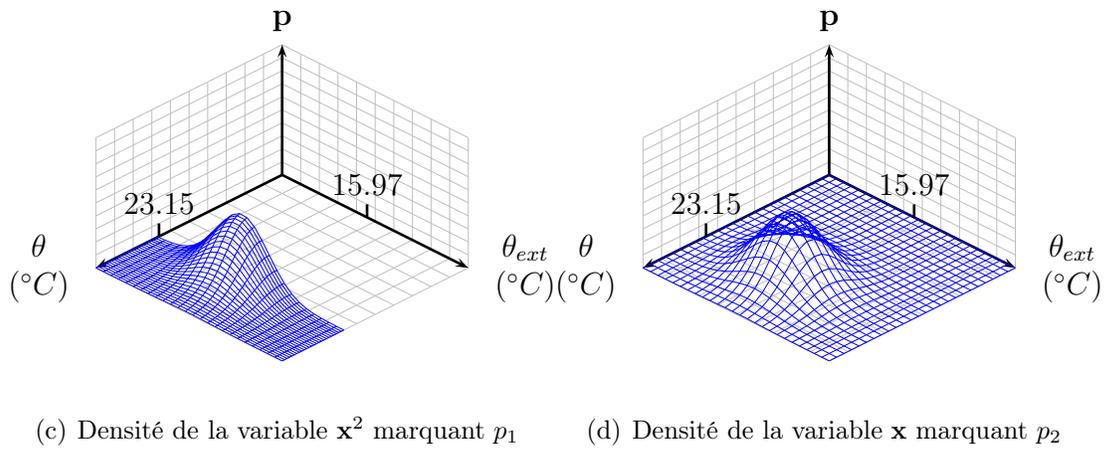
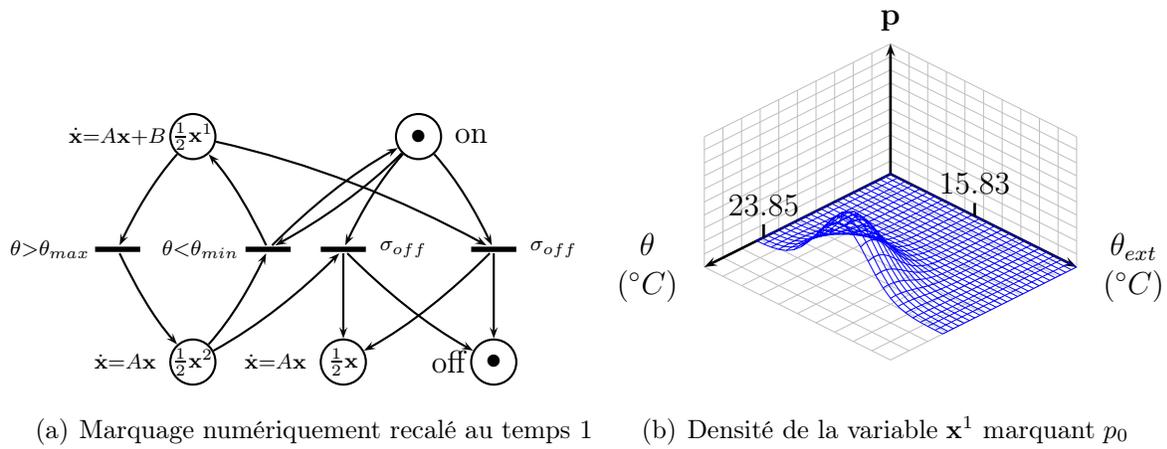


FIG. 4.9: Marquage numérique recalé à l'instant 1 à partir du marquage $\widehat{\mathcal{M}}'(1|0)$ et de l'observation numérique $z_1 = 23$.

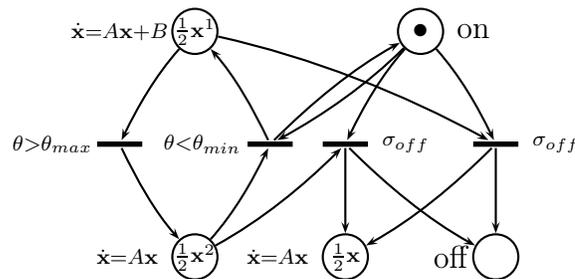


FIG. 4.10: Marquage recalé au temps 1 : le jeton de la place p_4 a été supprimé car $on \notin \{off\}$.

2.2.3 Recalage

Le marquage recalé (numériquement et symboliquement) est le marquage $\widehat{\mathcal{M}}(k+1|k+1)$ estimé au temps $k+1$ sachant l'observation au temps $k+1$ et l'estimation au temps k . Ce marquage est utilisé pour la prochaine étape de prédiction, qui calcule le marquage $\widehat{\mathcal{M}}(k+2|k+1)$, puis le met à jour à partir de l'observation $k+2$, et ainsi de suite...

3 Conclusions

Nous avons défini dans ce chapitre un principe d'estimation du marquage d'un réseau de Petri particulière inspiré du filtre de Kalman. Ce principe construit un marquage $\widehat{\mathcal{M}}(k+1|k+1)$ à partir de l'observation au temps $k+1$.

Après plusieurs tentatives, nous avons défini le marquage d'une place numérique comme une variable aléatoire de loi normale tronquée. La prédiction du marquage au temps $k+1$ à partir du marquage au temps k se fait selon deux évolutions :

1. une évolution isoparticule bruitée, qui correspond au pseudo-tir des transitions, de sorte à prédire les différents changements de modes possibles du système, en fonction des valeurs probables du vecteur d'état, et de l'arrivée potentielle d'événements ;
2. une évolution isomarquage bruitée, qui reprend les équations de prédiction du filtre de Kalman pour calculer les valeurs prédites du vecteur d'état.

Cette prédiction est ensuite recalée à partir d'une nouvelle observation. Ce recalage :

1. met à jour les distributions de probabilité des marquages numériques en utilisant les équations de recalage du filtre de Kalman ;
2. met à jour le marquage symbolique en conservant les configurations qui contiennent l'observation.

Finalement, le marquage recalé est impliqué dans la prochaine étape de prédiction, et ainsi de suite.

L'estimateur hybride proposé dans ce chapitre présente de nombreux inconvénients :

- les équations du filtre de Kalman ont été modifiées par l'ajout d'un domaine $D_k(p)$ pour chaque place p au temps k ; cette modification peut remettre en cause l'optimalité de l'estimateur ;
- cette modification ne permet pas, en l'état, de garantir que le marquage recalé vérifie l'équation (4.3), car les densités sont recalées de façon indépendantes, et le domaine n'est pas mis à jour ;
- le calcul de la distribution aval à une transition est "approximative" : la construction d'une probabilité conditionnelle, mathématiquement fondée, empêcherait la représentation gaussienne du marquage ;

- la troncature des distributions mène à un grand nombre de pseudo-tirs : à chaque instant, on peut se retrouver avec une probabilité très faible de franchir une transition, ce qui va fortement découper la distribution ;
- le pseudo-tir de transitions événementielles nécessite l'introduction de probabilités de franchissement $\mathbf{P}(\sigma)$, ce que l'on voulait justement éviter et que l'on reprochait aux méthodes existantes ; instaurer une probabilité d'arrivée de $\frac{1}{2}$ lorsque l'on n'a aucune information sur l'événement est totalement injustifié (et faux).

En conclusion, le principe d'estimation proposé ici est peu convaincant : la construction des distributions aval est arbitraire ; l'attribution de probabilités aux événements est injustifiée ; on risque de perdre le caractère optimal du filtre, ce qui réduit fortement son intérêt, car il limite toujours la modélisation à des équations linéaires et des bruits blancs gaussiens.

Finalement, l'estimateur basé sur des filtres de Kalman est abandonné. Nous proposons dans le chapitre 5 un estimateur inspiré du filtrage particulaire. Nous définirons dans ce chapitre les principes de l'estimation et pourrons constater que les problèmes posés par l'estimateur à base de filtres de Kalman sont résolus.

Chapitre 5

Estimation du marquage d'un réseau de Petri particulière par filtrage particulière

La tentative d'utiliser une méthode fondée sur le filtre de Kalman pour l'estimation du marquage d'un réseau de Petri particulière a mis en avant un certain nombre de défauts. Dans ce chapitre, nous proposons d'utiliser une méthode inspirée du filtrage particulière. Ainsi l'estimateur mis en œuvre pourra traiter des équations non linéaires et des bruits non gaussiens. Mais surtout, la modélisation discrétisée de l'incertitude numérique semble plus adaptée au modèle de réseau de Petri. Nous présentons dans ce chapitre le principe d'estimation et l'illustrons à travers l'exemple du thermostat.

1 Marquage particulière

Dans le cadre du filtrage particulière, la distribution de probabilité sur la valeur de l'état du système est approchée par un ensemble discret de particules. Ainsi :

$$\mathbf{p}(\mathbf{x}) \sim \sum_{i=1}^N \omega^i \delta_{\pi^i} \quad (5.1)$$

où π^i est la i -ème particule, ω^i son poids, δ la fonction de Kronecker et N le nombre de particules du modèle.

La représentation de l'incertitude sur la valeur du marquage d'un réseau de Petri particulière suit une modélisation similaire : chaque place numérique est marquée par un

certain nombre de particules, chaque particule représentant une valeur possible du vecteur d'état numérique du système.

En ce qui concerne la représentation de l'incertitude symbolique, nous conservons l'utilisation d'un macro-marquage pour modéliser les différents modes possibles du système. Chaque place symbolique marquée porte une configuration possible du système.

2 Principe d'estimation

L'estimation du marquage $\widehat{\mathcal{M}}(k|k)$ du réseau particulaire à l'instant k à partir de l'observation à l'instant k est assurée par un processus en deux étapes : prédiction et recalage.

2.1 Prédiction

La prédiction, comme nous l'avons vu dans le cadre du filtre de Kalman, calcule le marquage $\widehat{\mathcal{M}}(k+1|k)$ qui représente l'ensemble des situations possibles à l'instant suivant.

Cette prédiction est hybride : une prédiction discrète (évolution isoparticule bruitée) qui calcule les changements de modes possibles, et une prédiction continue (évolution isomarquage bruitée) qui calcule les valeurs suivantes des particules.

2.1.1 Évolution isoparticule bruitée

L'évolution isoparticule bruitée consiste à jouer le réseau jusqu'à obtenir un marquage terminal. Ce jeu représente les changements de modes de fonctionnement du système et est basé sur la satisfaction de conditions et l'arrivée d'événements.

Nous avons déjà discuté dans le chapitre précédent (remarque 4.12) de la simulation de l'arrivée d'événements pour pouvoir prédire les comportements possibles du système. Nous avons également critiqué l'association d'une probabilité d'arrivée aux événements, qui limite l'utilisation et le bien-fondé de l'estimateur.

Nous avons également discuté (remarque 4.21) de cette simulation en établissant que chaque événement ne devait être simulé (prédit) qu'une fois lors de l'étape de prédiction.

Par conséquent, nous définissons l'évolution isoparticule par :

Définition 5.1 (Transition pseudo-franchissable)

Une transition t d'un réseau de Petri particulière \mathcal{R} est **pseudo-franchissable** pour le marquage estimé $\widehat{\mathcal{M}}(k|k)$ si et seulement si :

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}(k|k, p) \geq Pre(p, t) \quad (5.2)$$

$$\forall p \in P_N, \quad |\{\pi \in \widehat{\mathcal{M}}(k|k, p) / A_c(t)(\pi)\}| \geq Pre(p, t) \quad (5.3)$$

$$\exists \sigma \in A_c(t) \Rightarrow t \text{ non tirée à l'instant } k \quad (5.4)$$

Une transition t est pseudo-franchissable si :

- ses places symboliques aval ont suffisamment de jetons ;
- ses places numériques aval possèdent au moins une¹ particule satisfaisant la condition associée à t ;
- si le franchissement de t dépend de l'arrivée d'un événement σ , t ne doit pas avoir été tirée à l'instant k .

Définition 5.2 (Pseudo-tir)

Le **pseudo-tir** d'une transition t pseudo-franchissable pour un marquage $\widehat{\mathcal{M}}(k|k)$ fournit un marquage $\widehat{\mathcal{M}}'(k|k)$ défini par :

$$\forall p \notin \bullet t \cup t \bullet, \quad \widehat{\mathcal{M}}'(k|k, p) = \widehat{\mathcal{M}}(k|k, p) \quad (5.5)$$

$$\forall p \in P_S \cap \bullet t, \quad \widehat{\mathcal{M}}'(k|k, p) = \widehat{\mathcal{M}}(k|k, p) \quad (5.6)$$

$$\forall p \in P_S \cap t \bullet, \quad \widehat{\mathcal{M}}'(k|k, p) = \widehat{\mathcal{M}}(k|k, p) + C(p, t) \quad (5.7)$$

$$\forall p \in P_N \cap \bullet t, \quad \widehat{\mathcal{M}}'(k|k, p) = \begin{cases} \widehat{\mathcal{M}}(k|k, p) & \text{si } \exists \sigma \in A_c(t) \\ \{\pi \in \widehat{\mathcal{M}}(k|k, p) / \overline{A_c(t)(\pi)}\} & \text{sinon} \end{cases} \quad (5.8)$$

$$\forall p \in P_N \cap t \bullet, \quad \widehat{\mathcal{M}}'(k|k, p) = \widehat{\mathcal{M}}(k|k, p) \cup \bigcup_{q \in P_N \cap \bullet t} \{\pi \in \widehat{\mathcal{M}}(k|k, q) / A_c(t)(\pi)\} \quad (5.9)$$

Le pseudo-tir d'une transition t :

- ne modifie pas le marquage des places non connectées à la transition ;
- ne modifie pas le marquage des places symboliques amont, de façon à conserver le mode de fonctionnement avant franchissement ;
- ajoute des jetons dans les places symboliques aval, de façon à prédire le nouveau mode de fonctionnement ;

¹Si $p \in \bullet t$ alors $Pre(p, t) = 1$ car \mathcal{R} est un graphe d'état numérique.

- si la transition attend un événement, les particules amont ne sont pas supprimées, de façon à prédire la situation où l'événement n'arrive pas ; les particules satisfaisant la condition sont copiées vers les places numériques aval pour prédire l'arrivée de l'événement ;
- si la transition n'attend pas d'événement, les particules amont qui satisfont la condition sont déplacées vers les places numériques aval.

2.1.2 Évolution isomarquage bruitée

L'évolution isomarquage consiste à calculer les valeurs possibles du vecteur d'état à l'instant suivant. Le filtrage particulaire se distingue ici par la simplicité de cette évolution : chaque particule évolue, de façon indépendante, selon l'équation qui régit le système.

Définition 5.3 (Évolution isomarquage bruitée)

L'évolution isomarquage d'un marquage $\widehat{\mathcal{M}}(k|k)$ fournit un marquage prédit $\widehat{\mathcal{M}}(k+1|k)$, défini par :

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}(k+1|k, p) = \widehat{\mathcal{M}}(k|k, p) \quad (5.10)$$

$$\forall p \in P_N, \quad \widehat{\mathcal{M}}(k+1|k, p) = \{\pi_{k+1|k} = F(k, p, \pi_{k|k}, b_k(p)) / \pi_{k|k} \in \widehat{\mathcal{M}}(k|k, p)\} \quad (5.11)$$

où $F(k, p)$ est la fonction associée à la place p à l'instant k et fournissant la valeur du vecteur d'état à l'instant $k+1$ à partir de sa valeur à l'instant k et d'un bruit $b_k(p)$.

Le marquage d'une place numérique p à l'instant $k+1$ consiste donc en l'ensemble de ses particules qui ont évolué selon l'équation et le bruit associés à p .

2.1.3 Synthèse

La prédiction se fait donc en deux étapes :

1. une *évolution isoparticule*, lors de laquelle on joue le réseau particulaire en pseudo-franchissant les transitions, jusqu'à obtenir un marquage terminal $\widehat{\mathcal{M}}^T(k|k)$;
2. une *évolution isomarquage*, qui consiste en une prédiction continue, et qui fournit le marquage prédit $\widehat{\mathcal{M}}(k+1|k)$.

2.1.4 Exemple : prédiction de la température

Considérons le réseau de Petri de la figure 5.1(a) modélisant le fonctionnement du thermostat et son marquage initial. L'incertitude initiale sur le vecteur d'état du système est représentée par 10 particules, qui ont été tirées de façon aléatoire selon la loi

$\mathcal{N}\left(\begin{pmatrix} 25 \\ 16 \end{pmatrix}, \begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}\right)$, et dont les valeurs sont représentées sur la figure 5.1(b). Le mode de fonctionnement initial du thermostat est *on*.

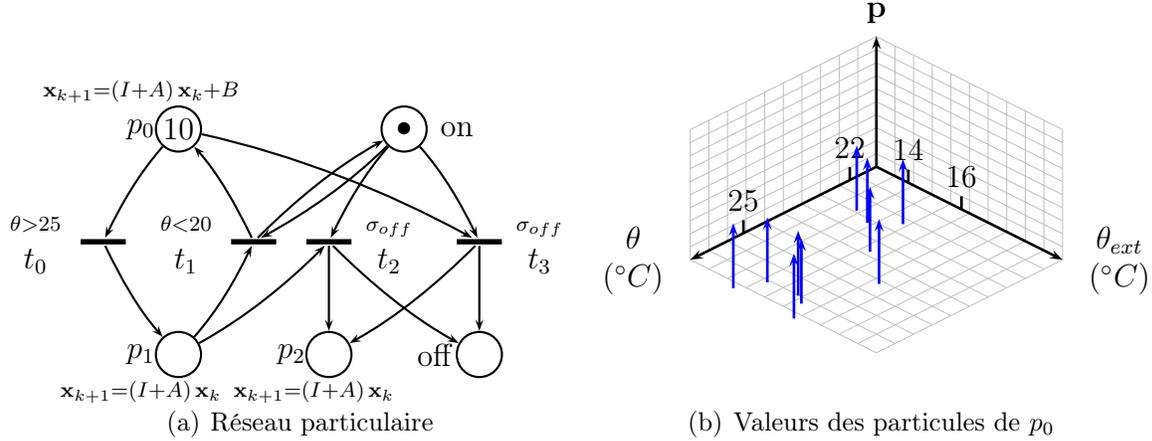


FIG. 5.1: Marquage initial $\widehat{\mathcal{M}}(0|0)$ du thermostat

La prédiction du fonctionnement du thermostat calcule l'ensemble des états possibles du système à l'instant 1.

Evolution isoparticule Les transitions t_0 et t_3 sont franchissables, la première car au moins une particule satisfait la condition $\theta > 25$, la seconde car on simule l'événement σ_{off} .

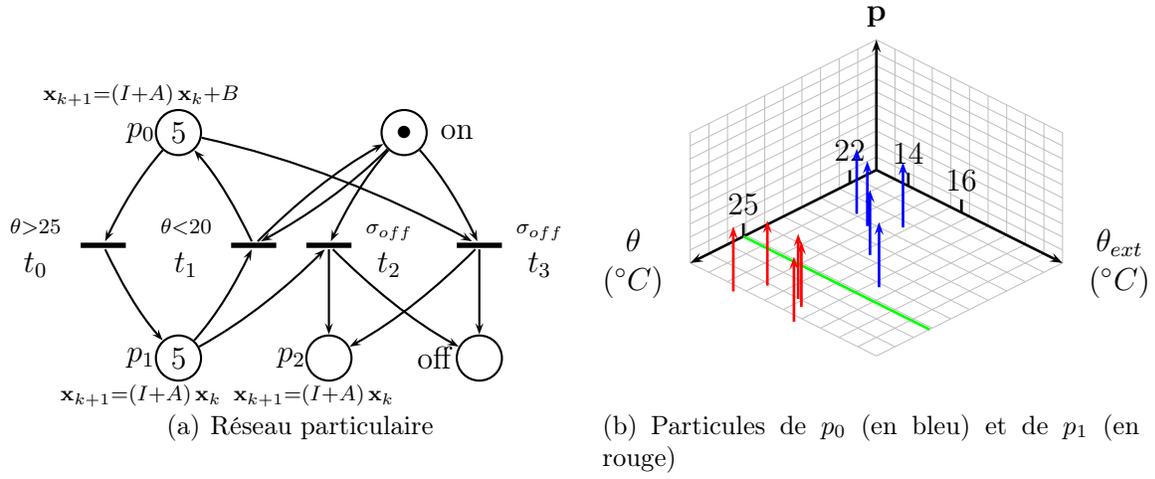
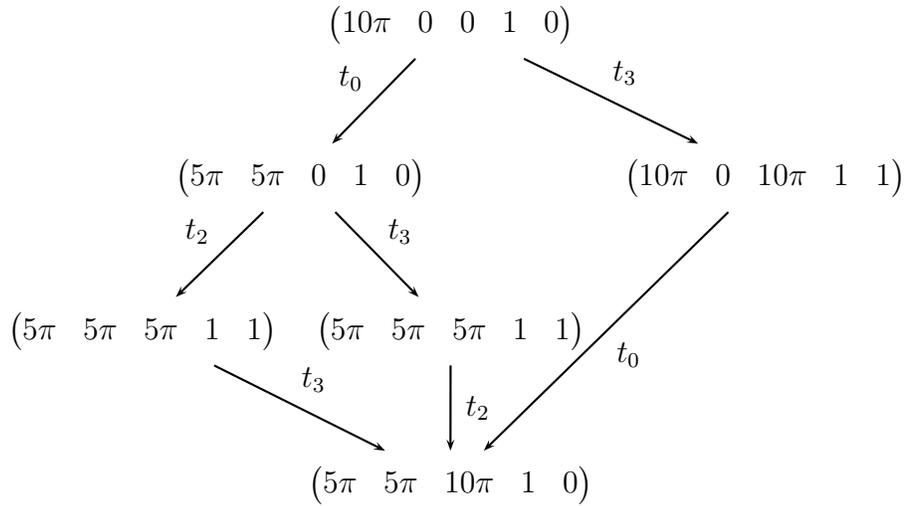
Le franchissement de t_0 à partir de $\widehat{\mathcal{M}}(0|0)$ fournit le marquage $\widehat{\mathcal{M}}'(0|0)$ de la figure 5.2 : 5 particules vérifient $\theta > 25$ et sont donc déplacées vers la place p_1 .

À partir de ce marquage, t_0 n'est plus franchissable (aucune particule ne satisfait la condition $\theta > 25$), t_1 n'est pas franchissable (aucune particule ne satisfait $\theta < 20$), t_2 et t_3 sont franchissables.

Pseudo-tirer t_3 à partir de $\widehat{\mathcal{M}}(0|0)$ copie toutes les particules de p_0 vers p_2 . Pseudo-tirer t_2 et t_3 à partir de $\widehat{\mathcal{M}}'(0|0)$ copie toutes les particules de p_0 vers p_2 et de p_1 vers p_2 (figure 5.3). On arrive donc dans les deux cas au même marquage terminal (figure 5.4).

Evolution isomarquage Nous procédons maintenant à une évolution isomarquage : les particules évoluent selon l'équation associée à leur place. Nous considérons un bruit de modèle identique pour toutes les places :

$$\forall k, \forall p \in P_N, b_k(p) \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}\right) \quad (5.12)$$

FIG. 5.2: $\widehat{\mathcal{M}}'(0|0)$ issu du pseudo-tir de t_0 à partir de $\widehat{\mathcal{M}}(0|0)$ FIG. 5.3: Marquages accessibles du réseau du thermostat à partir du marquage estimé $\widehat{\mathcal{M}}(0|0)$

Les équations associées aux places du réseau sont :

$$\mathbf{x}_{k+1} = \begin{pmatrix} 0.8 & 0.2 \\ 0 & 1 \end{pmatrix} \mathbf{x}_k + \begin{pmatrix} 2 \\ 0 \end{pmatrix} + b_k \quad (5.13)$$

$$\mathbf{x}_{k+1} = \begin{pmatrix} 0.8 & 0.2 \\ 0 & 1 \end{pmatrix} \mathbf{x}_k + b_k \quad (5.14)$$

L'évolution isomarquage fournit donc le marquage $\widehat{\mathcal{M}}(1|0)$ représenté sur la figure 5.5.

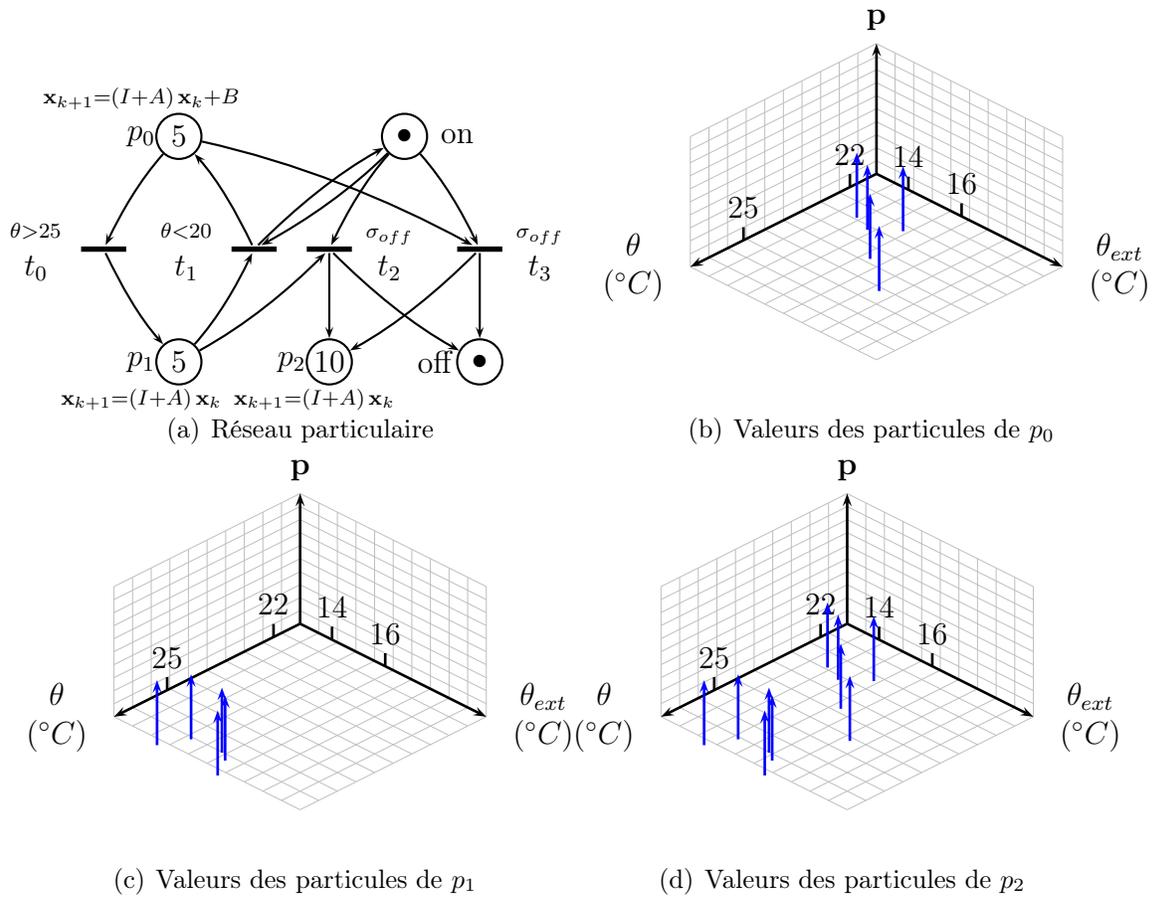
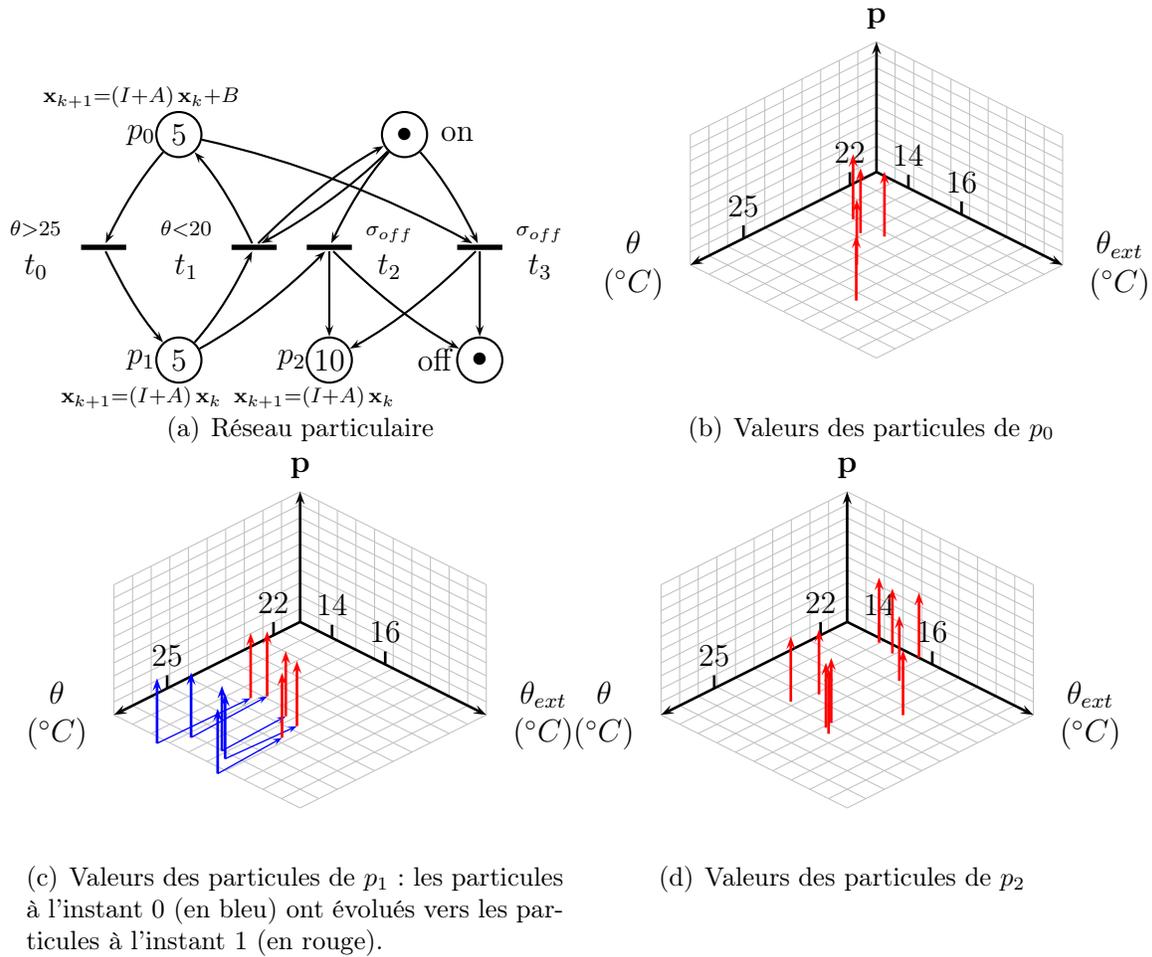


FIG. 5.4: Marquage terminal $\widehat{\mathcal{M}}^T(0|0)$

FIG. 5.5: Marquage prédit $\widehat{\mathcal{M}}(1|0)$.

2.2 Recalage

Le recalage consiste en la mise à jour du marquage $\widehat{\mathcal{M}}(k+1|k)$ à partir de l'observation à l'instant $k+1$. Ce recalage est hybride : un recalage numérique de la valeur des particules, et un recalage symbolique du mode de fonctionnement du système.

2.2.1 Recalage numérique

Le recalage numérique met à jour les valeurs des particules, qui représentent la distribution de probabilité du vecteur d'état, à partir de l'observation numérique \mathbf{z}_{k+1} .

Ce recalage s'inspire du filtrage particulaire :

- les particules sont pondérées à partir d'une valeur de vraisemblance ;
- les meilleures particules sont sélectionnées.

Définition 5.4 (Recalage numérique : pondération)

Soit $\widehat{\mathcal{M}}(k+1|k)$ le marquage prédit et \mathbf{z}_{k+1} l'observation à l'instant $k+1$. La pondération consiste à associer à chaque particule de $\widehat{\mathcal{M}}(k+1|k)$ un poids ω_{k+1}^i défini par :

$$\forall \pi_{k+1|k}^i \in \widehat{\mathcal{M}}(k+1|k), \quad \tilde{\omega}_{k+1}^i = g(\pi_{k+1|k}^i, \mathbf{z}_{k+1}) \quad (5.15)$$

$$\forall \pi_{k+1|k}^i \in \widehat{\mathcal{M}}(k+1|k), \quad \omega_{k+1}^i = \frac{\tilde{\omega}_{k+1}^i}{\sum_{\pi_{k+1|k}^j \in \widehat{\mathcal{M}}(k+1|k)} \tilde{\omega}_{k+1}^j} \quad (5.16)$$

où g est la fonction de vraisemblance, $\tilde{\omega}_{k+1}^i$ la valeur de vraisemblance de la particule i et ω_{k+1}^i son poids, normalisé, de sorte que

$$\sum_{i=1}^N \omega_{k+1}^i = 1$$

Algorithme 5.1 Rééchantillonnage

ENTRÉES: $N, \widehat{\mathcal{M}}(k+1|k)$

SORTIES: $\widehat{\mathcal{M}}(k+1|k+1)$

$\widehat{\mathcal{M}}(k+1|k+1) \leftarrow \emptyset$

pour $i \in \llbracket 1; N \rrbracket$ **faire**

$$\pi_{k+1|k+1}^i \sim \sum_{\pi_{k+1|k}^j \in \widehat{\mathcal{M}}(k+1|k)} \omega_{k+1}^j \delta_{\pi_{k+1|k}^j}$$

$$\omega_{k+1}^i \leftarrow \frac{1}{N}$$

5: $p \leftarrow \widehat{\mathcal{M}}^{-1}(k+1|k, \pi_{k+1|k+1}^i)$

$$\widehat{\mathcal{M}}(k+1|k+1, p) \leftarrow \pi_{k+1|k+1}^i$$

fin pour

pour $p \in P_S$ **faire**

$$\widehat{\mathcal{M}}(k+1|k+1, p) \leftarrow \widehat{\mathcal{M}}(k+1|k, p)$$

10: **fin pour**

Le rééchantillonnage consiste à tirer N particules parmi l'ensemble des particules pondérées (lignes 2 et 3). Ces N particules ont un poids de $\frac{1}{N}$ (ligne 4). Les particules sélectionnées

tionnées sont conservées dans leurs places (lignes 5 et 6) alors que les autres particules sont supprimées. Le marquage des places symboliques ne change pas (lignes 8 et 9).

2.2.2 Recalage symbolique

À partir de l'observation $\boldsymbol{\mu}_{k+1}$ du mode du système à l'instant $k + 1$, le recalage symbolique met à jour le marquage prédit $\widehat{\mathcal{M}}(k + 1|k)$.

De même que pour le recalage numérique, le recalage symbolique :

- évalue les jetons en fonction du mode observé ;
- sélectionne les meilleurs jetons.

Cette évaluation se fait par la mise à jour d'une relation de préférence, qui détermine quelles sont les places qui correspondent le mieux au mode observé. La mise à jour de relations de préférence, sous forme de préordres partiels, est proposée dans (Benferhat *et al.*, 2005). À partir de ces travaux, nous proposons la mise à jour d'une relation de préférence \succeq_{k+1}^Δ à l'instant $k + 1$ à partir de la relation \succeq_k^Δ à l'instant k et de l'observation du mode $\boldsymbol{\mu}_{k+1}$.

Définition 5.5 (Recalage symbolique : classement)

Soit $\widehat{\mathcal{M}}(k + 1|k)$ le marquage prédit et $\boldsymbol{\mu}_{k+1}$ le mode observé à l'instant $k + 1$. La mise à jour de la relation \succeq_k^Δ sur les places symboliques fournit une relation \succeq_{k+1}^Δ définie par :

$$\forall p, q \in P_S, \quad \widehat{\mathcal{M}}(k + 1|k, p) \neq 0, \widehat{\mathcal{M}}(k + 1|k, q) \neq 0, \quad (5.17)$$

$$\boldsymbol{\mu}_{k+1} \subset \delta(p), \boldsymbol{\mu}_{k+1} \not\subset \delta(q) \Rightarrow p \succeq_{k+1}^\Delta q \quad (5.17)$$

$$\boldsymbol{\mu}_{k+1} \subset \delta(p), \boldsymbol{\mu}_{k+1} \subset \delta(q) \Rightarrow (p \succeq_{k+1}^\Delta q \Leftrightarrow p \succeq_k^\Delta q) \quad (5.18)$$

$$\boldsymbol{\mu}_{k+1} \not\subset \delta(p), \boldsymbol{\mu}_{k+1} \not\subset \delta(q) \Rightarrow (p \succeq_{k+1}^\Delta q \Leftrightarrow p \succeq_k^\Delta q) \quad (5.19)$$

Les places symboliques marquées dont les configurations contiennent l'observation sont préférées à celles ne contenant pas l'observation. Dans les autres cas, la relation de préférence à l'instant $k + 1$ est identique à celle à l'instant k . La relation \succeq_{k+1}^Δ ne porte que sur les places symboliques marquées à l'instant $k + 1$ ($\widehat{\mathcal{M}}(k + 1|k, p) \neq 0$).

Définition 5.6 (Recalage symbolique : sélection)

Le recalage symbolique du marquage $\widehat{\mathcal{M}}(k + 1|k)$ à partir de la relation \succeq_{k+1}^Δ fournit

le marquage recalé $\widehat{\mathcal{M}}(k+1|k+1)$, défini par :

$$\forall p \in P_N, \quad \widehat{\mathcal{M}}(k+1|k+1, p) = \widehat{\mathcal{M}}(k+1|k, p) \quad (5.20)$$

$$\forall p \in P_S, \quad \widehat{\mathcal{M}}(k+1|k+1, p) = \begin{cases} 0 & \text{si } p \in \min_{P_S} \geq_{k+1}^{\Delta} \\ \widehat{\mathcal{M}}(k+1|k, p) & \text{sinon} \end{cases} \quad (5.21)$$

Lors de la sélection, on supprime le marquage des places symboliques les moins préférées :

$$p \in \min_{P_S} \geq_{k+1}^{\Delta} \Leftrightarrow \begin{cases} \forall q \in P_S, q \geq_{k+1}^{\Delta} p \text{ ou } p, q \text{ incomparables} \\ \exists q \in P_S, q >_{k+1}^{\Delta} p \end{cases} \quad (5.22)$$

C'est-à-dire que p n'est strictement préférée à aucune autre place, et il existe une place strictement préférée à p .

2.2.3 Synthèse

Le recalage du marquage $\widehat{\mathcal{M}}(k+1|k)$ prédit à l'instant $k+1$ à partir de l'observation $(\mathbf{z}_{k+1}, \boldsymbol{\mu}_{k+1})$ à l'instant $k+1$ consiste en :

1. une pondération des particules du marquage prédit,
2. un rééchantillonnage de ces particules,
3. la mise à jour de la relation de préférence sur les places symboliques marquées,
4. la suppression des jetons symboliques les moins préférés.

Ce processus fournit un marquage $\widehat{\mathcal{M}}(k+1|k+1)$ qui est alors prédit au temps $k+2$, et ainsi de suite...

2.2.4 Exemple : recalage de la température

La prédiction du fonctionnement du thermostat a fourni le marquage prédit $\widehat{\mathcal{M}}(1|0)$ (figure 5.5). L'observation à l'instant 1 est :

- $\mathbf{z}_1 = 23 \circ C$,
- $\boldsymbol{\mu}_1 = on$.

Nous considérons un bruit de mesure blanc gaussien, de variance 0.6 :

$$\mathbf{P}(v_{k+1} \in X) = \int_X \mathcal{N}(0, 0.6)(x) dx \quad (5.23)$$

La fonction de vraisemblance est alors

$$g(\mathbf{x}, \mathbf{z}_{k+1}) = \mathcal{N}(\mathbf{z}_{k+1}, 0.6)(H \mathbf{x}) \quad (5.24)$$

où $H = \begin{pmatrix} 1 & 0 \end{pmatrix}$ est la fonction d'observation.

Le marquage pondéré est représenté sur la figure 5.6, où le poids de chaque particule est représenté par sa hauteur. L'observation numérique bruitée est représentée en vert.

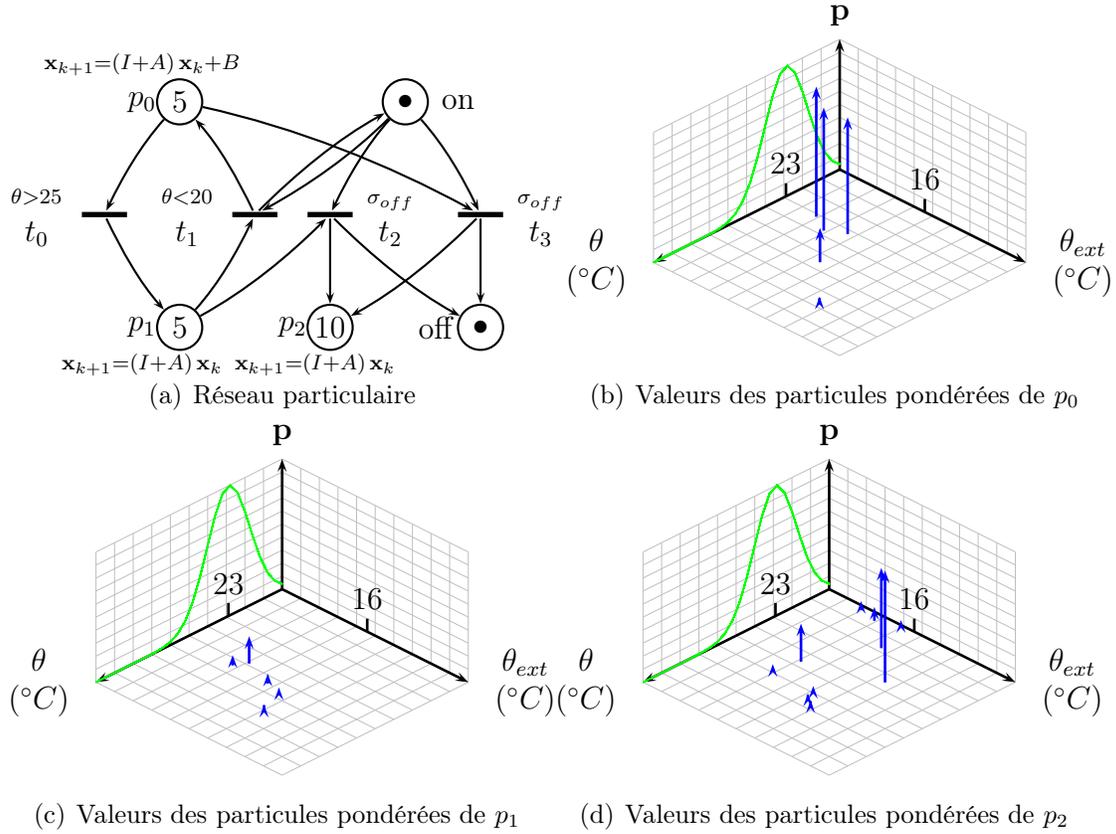


FIG. 5.6: Marquage pondéré $\widehat{\mathcal{M}}(1|0)$. La fonction de vraisemblance est représentée en vert sur le plan (θ, \mathbf{p}) .

Le rééchantillonnage du marquage pondéré fournit le marquage de la figure 5.7 : 10 particules ont été conservées, 7 dans p_0 et 3 dans p_2 .

En l'absence d'information à l'instant 0, la relation de préférence \geq_0^Δ est vide. Les places symboliques prédites sont p_3 et p_4 , avec :

- $\delta(p_3) = on$,
- $\delta(p_4) = off$.

Donc $\boldsymbol{\mu}_1 \subset \delta(p_3)$ et $\boldsymbol{\mu}_1 \not\subset \delta(p_4)$ et la relation de préférence à l'instant 1 est :

$$p_3 \geq_1^\Delta p_4 \quad (5.25)$$

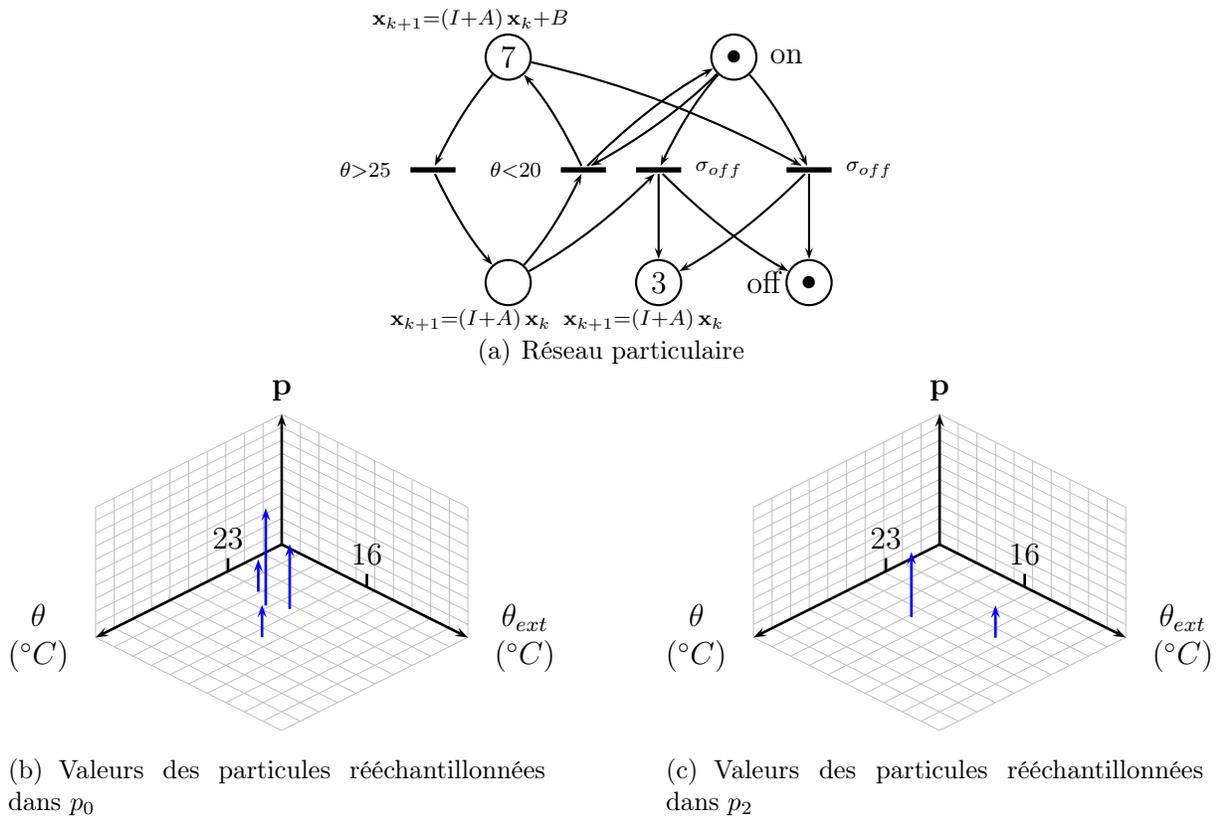


FIG. 5.7: Marquage rééchantillonné. La hauteur des flèches représente le nombre de particules ayant la même valeur (une pour les plus petites, deux pour les moyennes, trois pour les grandes).

p_4 est alors l'élément le moins préféré de la relation, et par conséquent le marquage recalé est le marquage de la figure 5.8 (p_4 est vide).

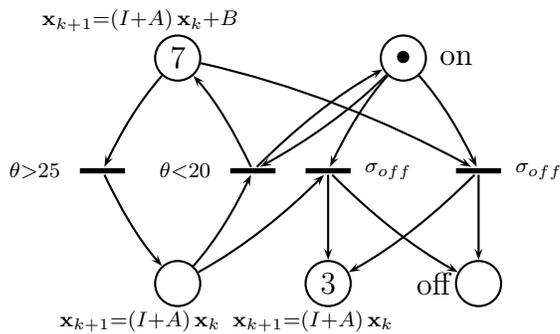


FIG. 5.8: Marquage recalé $\widehat{\mathcal{M}}(1|1)$, dont les valeurs des particules sont représentées sur la figure 5.7.

3 Conclusions

Nous avons proposé dans ce chapitre un principe d'estimation inspiré du filtrage particulaire. L'incertitude sur la valeur de l'état numérique du système est modélisée par un ensemble de particules qui représentent des valeurs possibles de l'état. De même, l'incertitude symbolique est modélisée par un macro-marquage : les places symboliques marquées représentent plusieurs modes possibles du système.

Le principe d'estimation est récursif. La phase de prédiction calcule les évolutions possibles du système. Une évolution isoparticule, basée sur le pseudo-tir des transitions, prédit les changements de modes du système. Ces changements de modes sont dépendants de la satisfaction de conditions ou de la réception d'événements.

L'évolution isomarquage, basée sur les équations d'évolution associées aux places numériques, prédit l'évolution continue du système.

Puis, à partir de l'observation à l'instant suivant, le marquage prédit est recalé. Ce recalage :

- pondère les particules prédites à partir d'une fonction de vraisemblance ;
- rééchantillonne ces particules de sorte à conserver les meilleures ;
- met à jour une relation de préférence sur les places symboliques ;
- démarque les places symboliques les moins préférées.

Ainsi, le marquage recalé correspond à une mise à jour du marquage prédit à partir de l'observation. Ce marquage est le point de départ pour la prédiction à l'instant suivant, et ainsi de suite.

Contrairement à l'estimateur basé sur le filtre de Kalman, l'estimateur proposé dans ce chapitre a de nombreux avantages :

- la discrétisation de l'incertitude facilite la prédiction, puisque chaque particule évolue de façon indépendante dans le réseau ;
- le pseudo-tir des transitions crée des distributions qui correspondent à des probabilités conditionnelles (les particules satisfaisant la condition sont déplacées) ;
- la prédiction de l'arrivée d'événements n'impose pas de fixer une valeur de probabilité à chaque événement ; il est donc possible de représenter une ignorance complète sur leur arrivée.

Nous avons pu appréhender le mécanisme de ce principe d'estimation à travers l'exemple du thermostat.

La chapitre 6 présente une synthèse du principe d'estimation que nous avons vu dans ce chapitre, et définit les étapes de l'estimation sous forme algorithmique. Ces algorithmes sont ensuite étudiés et évalués, de façon théorique et expérimentale.

Nous verrons ensuite (chapitre 7) une méthode de diagnostic d'états défaillants à partir de cet estimateur.

Chapitre 6

Algorithmique de l'estimateur

Le principe d'estimation du marquage d'un réseau de Petri particulière par filtrage particulière apporte une solution intéressante au problème de l'estimation des systèmes hybrides. Les réseaux de Petri particulières permettent de modéliser le comportement et l'état hybrides de ces systèmes. L'estimateur prédit et met à jour leur marquage en prenant en compte des observations hybrides (état numérique et mode), des dynamiques et des bruits complexes, des événements non stochastiques. Dans ce chapitre, nous étudions les aspects algorithmiques de cet estimateur.

1 Synthèse

1.1 Les réseaux de Petri particulières

Un réseau de Petri particulière modélise le comportement hybride du système : l'évolution symbolique est représentée par le déplacement des jetons dans le réseau, l'évolution numérique est représentée par l'évolution des valeurs des particules à partir d'équations différentielles.

La propriété de *graphe d'état numérique* (définition 3.5) permet de caractériser les réseaux assurant une modélisation "raisonnable" du système.

1.2 Estimation de l'état d'un système hybride

L'estimation de l'état hybride du système à l'instant k est représentée par un marquage $\widehat{\mathcal{M}}(k|k)$. Ce marquage contient :

- des jetons, qui activent les configurations associées aux places qu'ils marquent. Ces configurations représentent l'ensemble des modes de fonctionnement possibles du système ;
- des particules, dont les valeurs représentent l'ensemble des vecteurs d'état numériques possibles du système.

L'estimation de ce marquage est fondée sur un principe (chapitre 5) dont nous étudions ici l'algorithmique.

2 Prédiction

La prédiction est le calcul du marquage estimé à l'instant suivant $\widehat{\mathcal{M}}^T(k+1|k)$ à partir du marquage estimé courant $\widehat{\mathcal{M}}(k|k)$. Ce calcul est composé d'une évolution isoparticule bruitée et d'une évolution isomarquage bruitée.

2.1 Évolution isoparticule

L'évolution isoparticule bruitée consiste à pseudo-tirer les transitions pseudo-franchissables du réseau particulière marqué (algorithme 6.1).

Le pseudo-tir d'une transition t se fait en plusieurs étapes :

1. on fait évoluer le marquage des places symboliques (lignes 5 à 11) : on ajoute des jetons dans les places aval sans modifier le marquage des autres places ;
2. pour chaque place numérique amont (lignes 12 et 13), on récupère les particules qui satisfont la condition de t (lignes 14 à 15) et on met à jour le marquage de la place (lignes 17 à 21) ;
3. pour chaque place numérique aval (lignes 26 et 27), on ajoute les particules issues des places amont à son marquage (ligne 28) ;
4. on note finalement $\widehat{\mathcal{M}}$ le marquage obtenu, $\widehat{\mathcal{M}}'$ étant le marquage en construction lors du pseudo-tir de t (lignes 31 et 32).

Ce principe est mis en œuvre tant qu'une transition du réseau est pseudo-franchissable (ligne 3). Cette condition est évaluée par un autre algorithme (6.2) qui fournit une transition pseudo-franchissable du réseau, ou rien si aucune transition n'est pseudo-franchissable.

Cet algorithme parcourt l'ensemble des transitions (ligne 1). Si la transition attend un événement et qu'elle a déjà été tirée (contraposée de la ligne 2), alors on la passe : elle n'est pas franchissable.

Sinon, on vérifie que pour chaque place symbolique (ligne 3), cette place contient plus de jetons que requis par la transition (ligne 5). Et pour chaque place numérique (ligne 8), que le nombre de particules satisfaisant la condition de la transition (lignes 10 à 15) est suffisant pour franchir la transition (ligne 16).

Algorithme 6.1 Evolution isoparticule

ENTRÉES: $\mathcal{R}, \widehat{\mathcal{M}}(k|k)$ **SORTIES:** $\widehat{\mathcal{M}}^T(k|k)$

```

1:  $\widehat{\mathcal{M}} \leftarrow \widehat{\mathcal{M}}(k|k)$ 
2:  $\widehat{\mathcal{M}}' \leftarrow \emptyset$ 
3: tantque  $\exists t \in T$  pseudo-franchissable pour  $\widehat{\mathcal{M}}$  faire
4:    $\Pi \leftarrow \emptyset$  { $\Pi$  est l'ensemble utilisé pour stocker les particules franchissant la transition}
5:   pour tout  $p \in P_S$  faire
6:     si  $C(p, t) > 0$  alors
7:        $\widehat{\mathcal{M}}'(p) \leftarrow \widehat{\mathcal{M}}(p) + C(p, t)$  { $C$  est la matrice d'incidence du réseau}
8:     sinon
9:        $\widehat{\mathcal{M}}'(p) \leftarrow \widehat{\mathcal{M}}(p)$ 
10:    finsi
11:  fin pour
12:  pour tout  $p \in P_N$  faire
13:    si  $Pre(p, t) > 0$  alors
14:      pour tout  $\pi \in \widehat{\mathcal{M}}(p)$  faire
15:        si  $A_c(t)(\pi)$  alors
16:           $\Pi \leftarrow \Pi \cup \{\pi\}$ 
17:          si  $\exists \sigma \in A_c(t)$  alors
18:             $\widehat{\mathcal{M}}'(p) \leftarrow \widehat{\mathcal{M}}'(p) \cup \{\pi\}$  { $\sigma$  est un événement attendu par la transition}
19:          finsi
20:          sinon
21:             $\widehat{\mathcal{M}}'(p) \leftarrow \widehat{\mathcal{M}}'(p) \cup \{\pi\}$ 
22:          finsi
23:        fin pour
24:      finsi
25:    fin pour
26:    pour tout  $p \in P_N$  faire
27:      si  $Post(p, t) > 0$  alors
28:         $\widehat{\mathcal{M}}'(p) \leftarrow \widehat{\mathcal{M}}(p) \cup \Pi$ 
29:      finsi
30:    fin pour
31:     $\widehat{\mathcal{M}} \leftarrow \widehat{\mathcal{M}}'$ 
32:     $\widehat{\mathcal{M}}' \leftarrow \emptyset$ 
33:  fin tantque
34: return  $\widehat{\mathcal{M}}$ 

```

Algorithme 6.2 Pseudo-franchissable

ENTRÉES: $\mathcal{R}, \widehat{\mathcal{M}}, A$ $\{A$ est l'ensemble des transitions déjà franchies lors de cette évolution}

SORTIES: t pseudo-franchissable

```

1: pour tout  $t \in T$  faire
2:    $f \leftarrow (\exists \sigma \in A_c(t) \text{ ou } t \notin A)$ 
3:   pour tout  $p \in P_S$  faire
4:     si  $f$  alors
5:        $f \leftarrow f \wedge (\widehat{\mathcal{M}}(p) \geq Pre(p, t))$ 
6:     finsi
7:   fin pour
8:   pour tout  $p \in P_N$  faire
9:     si  $f$  alors
10:       $i \leftarrow 0$ 
11:      pour tout  $\pi \in \widehat{\mathcal{M}}(p)$  faire
12:        si  $A_c(t)(\pi)$  alors
13:           $i \leftarrow i + 1$ 
14:        finsi
15:      fin pour
16:       $f \leftarrow f \wedge (i \geq Pre(p, t))$ 
17:    finsi
18:  fin pour
19:  si  $f$  alors
20:     $A \leftarrow A \cup \{t\}$ 
21:  return  $t$ 
22: finsi
23: fin pour
24: return  $\emptyset$ 

```

Si toutes ces conditions sont satisfaites, alors la transition est pseudo-franchissable (lignes 19 à 21).

L'algorithme 6.2 **termine**, chaque boucle étant effectuée sur un nombre fini d'éléments. Sa complexité temporelle au pire cas est :

$$\mathcal{C}(6.2) = \mathcal{O}(|T| \cdot |P| \cdot N) \quad (6.1)$$

Cet algorithme est donc linéaire en le nombre de transitions ($|T|$), en le nombre de places ($|P|$) et en le nombre de jetons du marquage (N).

Remarque 6.1 (Complexité)

Nous donnons, dans ce chapitre, les complexité des algorithmes en fonction de trois paramètres : $|T|$ et $|P|$, qui représentent la taille du réseau, et donc la taille du système que l'on veut estimer ; N , le nombre de particules, qui constitue le paramètre essentiel de l'estimateur et permettra d'approcher au mieux l'état du système. Si on peut régler le nombre de particules représentant l'estimation numérique de l'état du système, le nombre de jetons "classiques" n'est pas paramétrable et dépend de la dynamique du système. On peut donc considérer que la taille du marquage suit une évolution équivalente à N .

En revanche, l'algorithme 6.2 n'est *a priori* **pas déterministe** : il renvoie la première transition pseudo-franchissable trouvée, et n'est donc déterministe que si les transitions sont systématiquement parcourues dans le même ordre.

La terminaison de l'algorithme 6.1 est plus délicate : dans cet algorithme on trouve une boucle *tantque* (ligne 3) basée sur une condition. Pour assurer que l'algorithme termine, il faut donc vérifier qu'il existe un moment à partir duquel la condition ne sera plus satisfaite, c'est-à-dire que le nombre de transitions pseudo-franchissables est fini.

La propriété de réseau non-Zénon (définition ??) n'est malheureusement pas suffisante pour assurer que l'algorithme 6.1 termine. En effet, considérons l'exemple de la figure 6.1(a).

Le réseau de la figure 6.1(a) est non-Zénon. En effet quel que soit le marquage initial \mathcal{M}_0 , la transition t_1 est franchissable au plus $\mathcal{M}_0(p_1)$ fois. Et donc, quel que soit le marquage accessible \mathcal{M} , le marquage de la place p_2 sera au plus de $\mathcal{M}_0(p_2) + \mathcal{M}_0(p_1)$. La transition t_0 est donc franchissable au plus $\mathcal{M}_0(p_2) + \mathcal{M}_0(p_1)$ fois à partir du marquage initial \mathcal{M}_0 . On ne peut donc pas tirer un nombre infini de fois les transitions t_0 et t_1 , et ce, quel que soit le marquage initial : le réseau est non-Zénon.

Essayons maintenant d'évaluer l'algorithme 6.1. Supposons que $P_N = \{p_0, p_2\}$ et $P_S = \{p_1, p_3\}$, et que le jeton de p_0 satisfait les conditions des transitions t_0 et t_1 . La transition t_1 est pseudo-franchissable, et son pseudo-tir engendre le marquage de la figure 6.1(b) : la particule s'est déplacée de p_0 vers p_2 , un jeton a été ajouté à la place p_3 , et

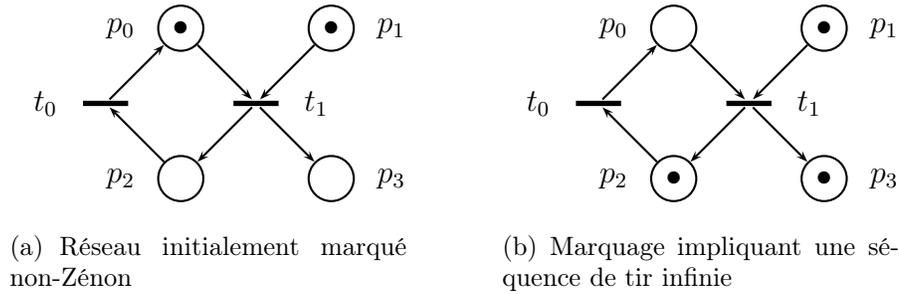


FIG. 6.1: Réseau non-Zénon pour lequel l'algorithme 6.1 ne termine pas.

le marquage de p_1 est resté identique. Dans ce marquage, la transition t_0 est pseudo-franchissable, et son pseudo-tir déplace à nouveau la particule de p_2 vers p_0 . Le marquage ainsi obtenu recouvre le marquage initial, et la séquence (t_1, t_0) est à nouveau franchissable. Le pseudo-tir de ces deux transitions peut donc se faire indéfiniment : l'algorithme 6.1 ne termine pas.

En conclusion, la propriété de réseau **non-Zénon** que l'on peut démontrer pour un réseau particulière n'est plus définie pour un pseudo-tir des transitions.

Pour évaluer la complexité de cet algorithme, nous supposons donc que, pour un réseau "bien conçu", l'algorithme termine. Une étude expérimentale (section 4) nous permettra de vérifier cette terminaison sur des instances des problèmes de navigation et du thermostat. On peut présumer que, dans la plupart des cas, il sera assez rare de tirer un grand nombre de transitions. Le pire cas auquel on peut être confronté est par exemple le cas du réseau de la figure 6.2, qui correspond à un marquage recalé $\widehat{\mathcal{M}}(k|k)$ d'un réseau de Petri particulière.

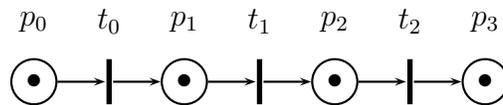


FIG. 6.2: Un exemple de pire cas pour la prédiction.

Dans cet exemple, le pire cas (le plus grand nombre de tirs) d'évolution isoparticule consiste à franchir d'abord la transition t_2 , puis les transitions t_1 et t_2 , puis les transitions t_0 , t_1 et t_2 . Le pire cas correspond donc au franchissement de $1 + 2 + \dots + |T|$ transitions,

soit $\frac{|T|(|T|-1)}{2}$ transitions. La complexité de l'algorithme est alors :

$$\mathcal{C}(6.1) = \mathcal{O}(N \cdot |P| \cdot |T|^3) \quad (6.2)$$

Pour un réseau donné ($|P|$ et $|T|$ fixés), l'algorithme est linéaire en le nombre de jetons dans le réseau (donc en le nombre de particules modélisant l'incertitude sur l'état du système).

Comme dans le cas de l'algorithme 6.2, l'algorithme 6.1 est **non-déterministe**, du fait du non-déterminisme de l'ordre de tir des transitions.

2.2 Évolution isomarquage

L'évolution isomarquage au temps k fournit le marquage prédit au temps $k + 1$ en faisant évoluer les particules selon les équations différentielles associées aux places numériques du réseau.

Algorithme 6.3 Évolution isomarquage

ENTRÉES: $\mathcal{R}, \widehat{\mathcal{M}}^T(k|k)$

SORTIES: $\widehat{\mathcal{M}}^T(k+1|k)$

- 1: $\widehat{\mathcal{M}}^T(k+1|k) \leftarrow \emptyset$
 - 2: **pour** $p \in P_S$ **faire**
 - 3: $\widehat{\mathcal{M}}^T(k+1|k, p) \leftarrow \widehat{\mathcal{M}}^T(k|k, p)$
 - 4: **fin pour**
 - 5: **pour** $p \in P_N$ **faire**
 - 6: **pour** $\pi \in \widehat{\mathcal{M}}^T(k|k, p)$ **faire**
 - 7: $b \leftarrow \text{Rand}(b_k)$ $\{b$ est une valeur tirée selon la loi b_k représentant le bruit de modèle à l'instant $k\}$
 - 8: $\widehat{\mathcal{M}}^T(k+1|k, p) \leftarrow \widehat{\mathcal{M}}^T(k+1|k, p) + \{F(k, p, \pi, b)\}$ $\{F(k, p, \cdot, \cdot)$ est la fonction d'évolution associée à la place p à l'instant $k\}$
 - 9: **fin pour**
 - 10: **fin pour**
 - 11: **return** $\widehat{\mathcal{M}}^T(k+1|k)$
-

L'algorithme **termine**, puisqu'il parcourt une et une seule fois chaque place et chaque particule, et sa complexité est :

$$\mathcal{C}(6.3) = \mathcal{O}(|P| \cdot N) \quad (6.3)$$

Cet algorithme est également **non déterministe** : le bruit de prédiction est tiré de façon aléatoire pour chaque particule.

2.3 Prédiction : synthèse

L'étape de prédiction consiste en une évolution isoparticule puis une évolution isomarquage. Sa complexité est donc finalement :

$$\mathcal{C}(\text{Prédiction}) = \mathcal{O}(N \cdot |P| \cdot |T|^3) \quad (6.4)$$

3 Recalage

Le recalage du marquage prédit à partir d'une observation se fait en plusieurs étapes :

1. une pondération des particules prédites,
2. un rééchantillonnage de ces particules,
3. un classement des configurations prédites,
4. une sélection de ces configurations, en éliminant les "dernières".

3.1 Recalage numérique

Algorithme 6.4 Pondération

ENTRÉES: $\widehat{\mathcal{M}}(k+1|k), \mathbf{z}_{k+1}$

SORTIES: ω_{k+1}^i

- 1: $s \leftarrow 0$ { s sert à calculer la somme des poids}
 - 2: **pour tout** $p \in P_N$ **faire**
 - 3: **pour tout** $\pi_{k+1|k}^i \in \widehat{\mathcal{M}}(k+1|k, p)$ **faire**
 - 4: $\tilde{\omega}_{k+1|k}^i \leftarrow g(\pi_{k+1|k}^i, \mathbf{z}_{k+1})$ { g est la fonction de vraisemblance}
 - 5: $s \leftarrow \tilde{\omega}_{k+1|k}^i + s$
 - 6: **fin pour**
 - 7: **fin pour**
 - 8: **pour tout** $p \in P_N$ **faire**
 - 9: **pour tout** $\pi_{k+1|k}^i \in \widehat{\mathcal{M}}(k+1|k)$ **faire**
 - 10: $\omega_{k+1|k}^i \leftarrow \frac{\tilde{\omega}_{k+1|k}^i}{s}$
 - 11: **fin pour**
 - 12: **fin pour**
-

L'algorithme de pondération (6.4) affecte à chaque particule (lignes 2 et 3) un poids correspondant à l'évaluation de cette particule par la fonction de vraisemblance (ligne 4). Puis ces poids sont normalisés, de sorte que leur somme vaille 1 (ligne 10).

L'algorithme **termine**, puisqu'il parcourt l'ensemble des places et des particules, et sa complexité est :

$$\mathcal{C}(6.4) = \mathcal{O}(N \cdot |P|) \quad (6.5)$$

Il est également **déterministe** puisqu'il affecte à chaque particule un poids issu d'une fonction déterministe.

Algorithme 6.5 Rééchantillonnage

ENTRÉES: $N, \widehat{\mathcal{M}}(k+1|k)$

SORTIES: $\widehat{\mathcal{M}}(k+1|k+1)$

```

1:  $\widehat{\mathcal{M}}(k+1|k+1) \leftarrow \emptyset$ 
2:  $\tilde{W} \leftarrow []$  { $\tilde{W}$  est un tableau contenant les valeurs de vraisemblance des particules}
3:  $W \leftarrow []$  { $W$  est un tableau contenant les poids additifs normalisés des particules
   (lignes 11 à 17)}
4:  $Pl \leftarrow []$  { $Pl$  est un tableau contenant la place marquée par chaque particule}
5: pour tout  $p \in P_N$  faire
6:   pour tout  $\pi_{k+1|k}^i \in \widehat{\mathcal{M}}(k+1|k, p)$  faire
7:      $Pl[i] \leftarrow p$ 
8:      $\tilde{W}[i] \leftarrow \omega_{k+1}^i$ 
9:   fin pour
10: fin pour
11:  $s \leftarrow 0$ 
12: pour  $i \in [1, |\tilde{W}|]$  faire
13:    $s \leftarrow s + \tilde{W}[i]$ 
14: fin pour
15:  $W[0] \leftarrow \tilde{W}[0]$ 
16: pour  $i \in [2, |\tilde{W}|]$  faire
17:    $W[i] \leftarrow W[i-1] + \frac{\tilde{W}[i]}{s}$ 
18: fin pour
19: pour  $i \in [1, N]$  faire
20:    $r \leftarrow \text{Rand}([0, 1])$ 
21:    $j \leftarrow \text{Cherche}(r, W)$ 
22:    $\widehat{\mathcal{M}}(k+1|k+1, Pl[j]) \leftarrow \widehat{\mathcal{M}}(k+1|k+1, Pl[j]) \cup \{\pi_{k+1|k}^j\}$ 
23: fin pour

```

L'algorithme de rééchantillonnage (6.5) stocke dans un premier temps le poids des particules ainsi que leur place dans deux tableaux (lignes 5 à 10). Puis on construit un tableau des poids normalisés (lignes 11 à 17). On tire N nombres de façon aléatoire sur $[0, 1]$ (ligne 19). Pour chacun de ces nombres, on cherche ensuite sa place dans le tableau

des poids (ligne 20), et cette recherche fournit la particule sélectionnée, qui est conservée dans le marquage recalé (ligne 21).

L'algorithme est **non déterministe** puisqu'une sélection aléatoire est opérée. Il **termine**, et sa complexité est :

$$\mathcal{C}(6.5) = \mathcal{O}(N \cdot |P|) \quad (6.6)$$

Remarque 6.2 (Algorithme de recherche)

L'algorithme *Cherche*(r, W) (ligne 20) cherche la position d'un nombre au sein d'un tableau ordonné (W est construit de sorte que $W[i+1] \geq W[i]$), et sa complexité est donc linéaire en la taille du tableau.

3.2 Recalage symbolique

Algorithme 6.6 Classement

ENTRÉES: $\widehat{\mathcal{M}}(k+1|k), \mu_{k+1}, \geq_k^\Delta \{\mu_{k+1}$ est le mode observé à l'instant $k+1$ et \geq_k^Δ la relation de préférence à l'instant $k\}$

SORTIES: \geq_{k+1}^Δ

```

1: pour tout  $p \in P_S$  faire
2:   si  $\widehat{\mathcal{M}}(k+1|k, p) \neq 0$  alors
3:     pour tout  $q \in P_S$  faire
4:       si  $\widehat{\mathcal{M}}(k+1|k, q) \neq 0$  alors
5:         si  $\mu_{k+1} \subset \delta(p)$  alors
6:           si  $\mu_{k+1} \subset \delta(q)$  alors
7:              $\geq_{k+1}^\Delta(p, q) \leftarrow \geq_k^\Delta(p, q)$  {Si les deux configurations contiennent l'observation, la préférence à l'instant  $k$  est conservée à l'instant  $k+1$ }
8:           sinon
9:              $p >_{k+1}^\Delta q$  {Sinon  $p$  est préférée à  $q$ }
10:          finsi
11:        sinon
12:          si  $\mu_{k+1} \subset \delta(q)$  alors
13:             $q >_{k+1}^\Delta p$ 
14:          sinon
15:             $\geq_{k+1}^\Delta(p, q) \leftarrow \geq_k^\Delta(p, q)$ 
16:          finsi
17:        finsi
18:      finsi
19:    fin pour
20:  finsi
21: fin pour

```

Le classement (algorithme 6.6) compare deux à deux l'ensemble des places symboliques marquées (lignes 1 à 4). Puis en fonction de la valeur de leur configuration, par rapport au mode observé, la relation de préférence est mise à jour :

- Si la configuration de p contient l'observation, mais la configuration de q non, alors p est préférée à q (ligne 9, ou inversement ligne 13) ;
- Dans tous les autres cas, on conserve la préférence à l'instant k (lignes 7 et 15).

L'algorithme est **déterministe** et **termine**. Sa complexité est :

$$\mathcal{C}(6.6) = \mathcal{O}(|P|^2) \quad (6.7)$$

L'algorithme de sélection (6.7) conserve les jetons des places symboliques qui ne sont pas "dernières" : pour chaque place symbolique p (ligne 1), s'il existe une autre place à laquelle p est préférée, alors on conserve le marquage de p (lignes 2 et 3), sinon, c'est que p est "dernière", et on supprime son marquage (ligne 5).

Ce test nécessite de parcourir l'ensemble des places symboliques. La complexité de l'algorithme, qui **termine** et est **déterministe**, est :

$$\mathcal{C}(6.7) = \mathcal{O}(|P|^2) \quad (6.8)$$

Algorithme 6.7 Sélection

ENTRÉES: $\widehat{\mathcal{M}}(k+1|k), \geq_{k+1}^{\Delta}$

SORTIES: $\widehat{\mathcal{M}}(k+1|k+1)$

1: **pour tout** $p \in P_S$ **faire**

2: **si** $\exists q / p \geq_{k+1}^{\Delta} q$ **alors**

3: $\widehat{\mathcal{M}}(k+1|k+1, p) \leftarrow \widehat{\mathcal{M}}(k+1|k, p)$

4: **sinon**

5: $\widehat{\mathcal{M}}(k+1|k+1, p) \leftarrow 0$

6: **finsi**

7: **fin pour**

3.3 Recalage : synthèse

La complexité résultante de l'étape de recalage est donc :

$$\mathcal{C}(\text{Recalage}) = \mathcal{O}(N \cdot |P|^2) \quad (6.9)$$

4 Évaluation expérimentale

4.1 Exhost-PIPE

La mise en œuvre des principes présentés dans ce manuscrit a mené à la réalisation d'un outil : Exhost-PIPE¹ (Bonnet-Torrès *et al.*, 2006). Ce logiciel permet, entre autres, la modélisation de réseaux de Petri particuliers et le jeu de ces réseaux. Une description détaillée d'Exhost-PIPE est fournie en annexe C.

Nous avons également implanté le processus d'estimation et les algorithmes présentés dans ce chapitre. Les résultats expérimentaux présentés ici (et plus loin dans ce manuscrit) ont été réalisés avec cet outil.

Pour obtenir les résultats présentés ici, nous avons réalisé les séries de simulation sur un ordinateur équipé d'un processeur Intel Pentium 4 d'une fréquence de 3GHz et équipé de 512Mo de mémoire vive.

4.2 Complexité en fonction de N

Pour évaluer la complexité des algorithmes en fonction de N , nous avons travaillé sur un jeu de systèmes de taille fixe ($|P|$ et $|T|$ constants).

4.2.1 Problème de navigation

Les premiers résultats ont été effectués sur le problème de navigation pour une grille 3×3 . Nous avons donc utilisé les instances Nav01 à Nav06 (voir l'annexe D pour une description de ces instances) de ce problème. Le tableau 6.1 contient les moyennes des temps de calcul des différents algorithmes pour l'instance Nav01 en fonction de N . Ces valeurs moyennes ont été calculées, pour chaque valeur de N , sur trois lances du processus d'estimation. Ces trois expériences ont été faites à partir du même marquage initial, des mêmes observations et sur un jeu de dix observations. La différence résidait dans la définition des bruits, réglés respectivement à $0.1I$, $0.2I$ et $0.3I$.

Remarque 6.3 (Somme des temps de calcul)

Les temps de calcul mesurés pour les phases de prédiction et de recalage ne correspondent pas à leur temps de calcul théorique : le temps de calcul de la prédiction, par exemple, devrait être la somme des temps de calcul des algorithmes 6.1 et 6.3. La différence entre la somme des temps et le temps mesuré peut s'expliquer par la gestion de l'affichage et des fichiers de log, et également par l'interruption possible de la tâche d'estimation par la machine virtuelle (processeur affecté à d'autres tâches de gestion de mémoire ou autres).

¹Extended high-level oriented and standard Platform Independant Petri net Editor, <http://www.onera.fr/dcsd-en/exhost-pipe/index.php>

N	Algorithmes							
	6.1	6.3	Prédiction	6.4	6.5	6.6	6.7	Recalage
100	29 740	19 020	51 212	14 915	10 978	176	24	28 796
250	42 310	31 225	76 029	19 815	52 515	216	13	107 794
500	86 242	89 707	179 277	33 388	64 124	152	8	103 842
750	69 730	79 771	152 912	42 875	42 761	203	12	94 033
1000	96 821	118 275	217 951	93 645	53 993	243	11	157 063
1250	110 012	113 791	226 803	62 412	68 612	207	11	139 603
1500	123 380	137 322	263 604	73 026	75 186	264	10	186 859
1750	153 680	165 643	322 075	84 399	166 200	242	8	264 513
2000	174 581	238 265	415 881	94 256	124 358	229	10	241 667

TAB. 6.1: Temps de calcul en micro-secondes pour l'instance Nav01 en fonction de N .

On peut également remarquer que, même dans le cas $N = 2000$, les temps de calcul des phases de prédiction et de recalage sont inférieurs à la demi-seconde, ce qui permet de traiter les deux phases lorsqu'une observation arrive toutes les secondes.

Nous avons ensuite fait une moyenne sur les temps de calcul des six instances Nav01 à Nav06, obtenue comme expliqué précédemment. L'évolution des temps de calcul des algorithmes liés à la prédiction et au recalage en fonction de N est représentée sur les figures 6.3 et 6.4.

Les trois algorithmes 6.1 (évolution isoparticule), 6.3 (évolution isomarquage) et de prédiction ont un temps de calcul qui suit une évolution à peu près linéaire en N , comme prévu par leur complexité théorique (équations (6.2), (6.3) et (6.4)).

Les algorithmes 6.4 (pondération), 6.5 (rééchantillonnage) et de recalage également ont un temps de calcul qui suit une évolution linéaire en N , ce qui est conforme à leur complexité théorique ($\mathcal{O}(N \cdot |P|)$ et $\mathcal{O}(N \cdot |P|^2)$). Les algorithmes 6.6 (classement) et 6.7 (sélection) ont un temps de calcul à peu près constant (conforme à leur complexité en $\mathcal{O}(|P|^2)$) et de plus ce temps de calcul est assez faible (quelques micro-secondes), puisque le réseau de navigation ne contient pas de place symbolique.

Les temps de calcul mesurés pour les instances Nav01 à Nav06 permettent donc de vérifier expérimentalement les complexités théoriques des algorithmes. De plus, on peut noter que l'algorithme 6.1 (évolution isoparticule) termine.

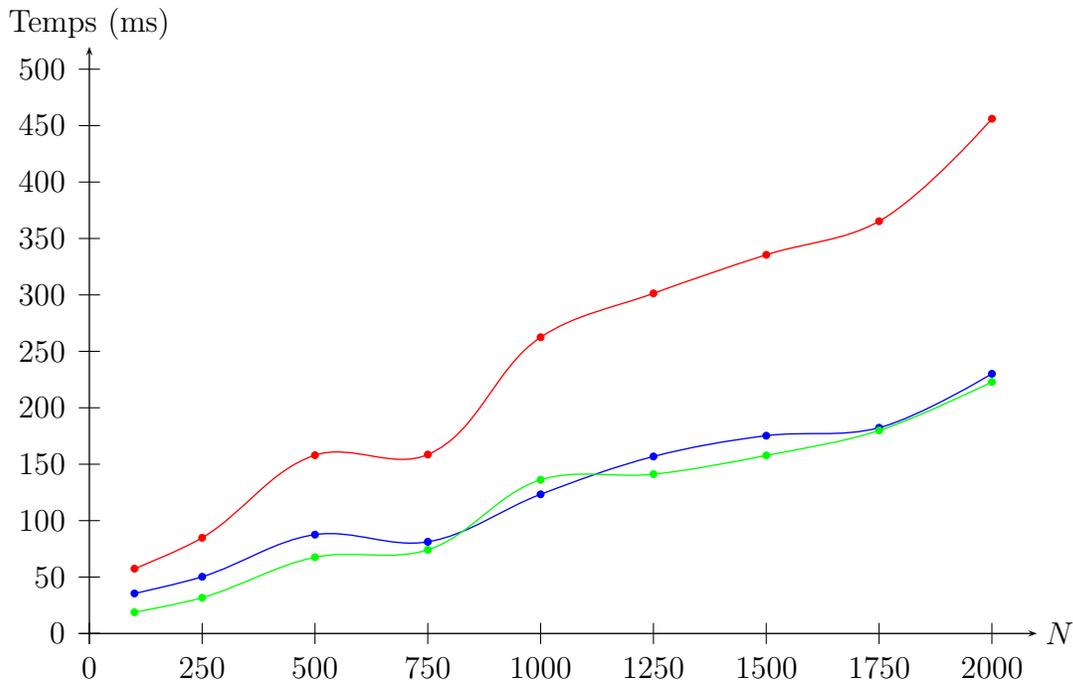


FIG. 6.3: Évolution des temps de calcul des algorithmes de prédiction en fonction de N . Les courbes bleue, verte et rouge correspondent respectivement aux algorithmes 6.1, 6.3 et de prédiction.

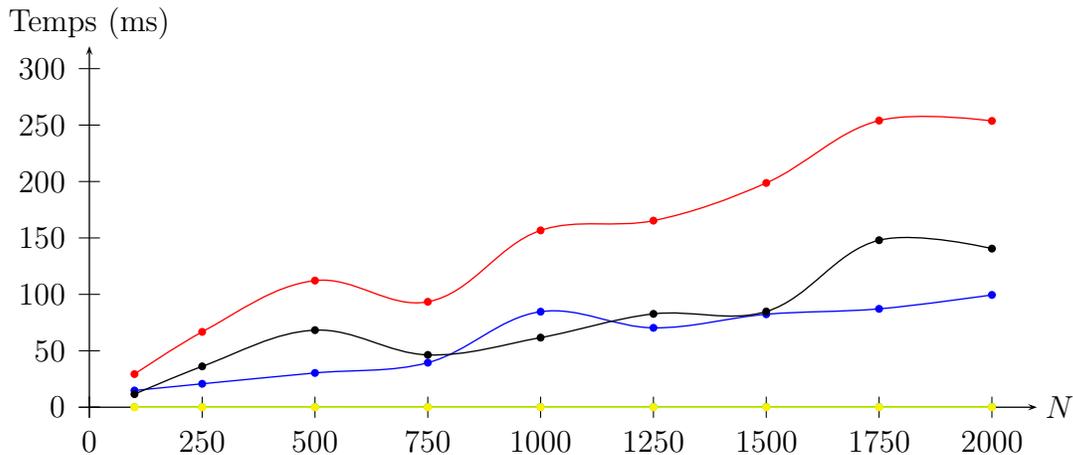


FIG. 6.4: Évolution des temps de calcul des algorithmes de recalage en fonction de N . Les courbes bleue, noire, verte (confondue avec la jaune), jaune et rouge correspondent respectivement aux algorithmes 6.4, 6.5, 6.6, 6.7 et de recalage.

4.2.2 Le thermostat

L'étude de l'influence du nombre de particules sur le problème du thermostat a été réalisée avec le réseau de la figure 6.5.

Les temps de calcul affichés sur les figures 6.6 et 6.7 correspondent à des valeurs

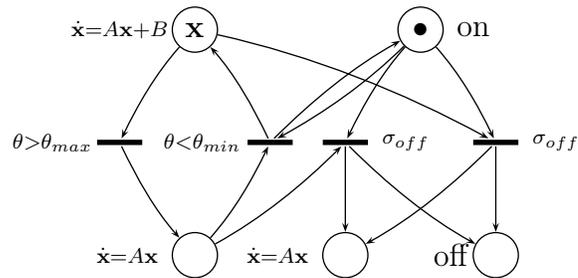


FIG. 6.5: Réseau de Petri du thermostat

moyennes calculées sur les 20 premières observations des instances Thermo01 à Thermo03, décrites en annexe E.

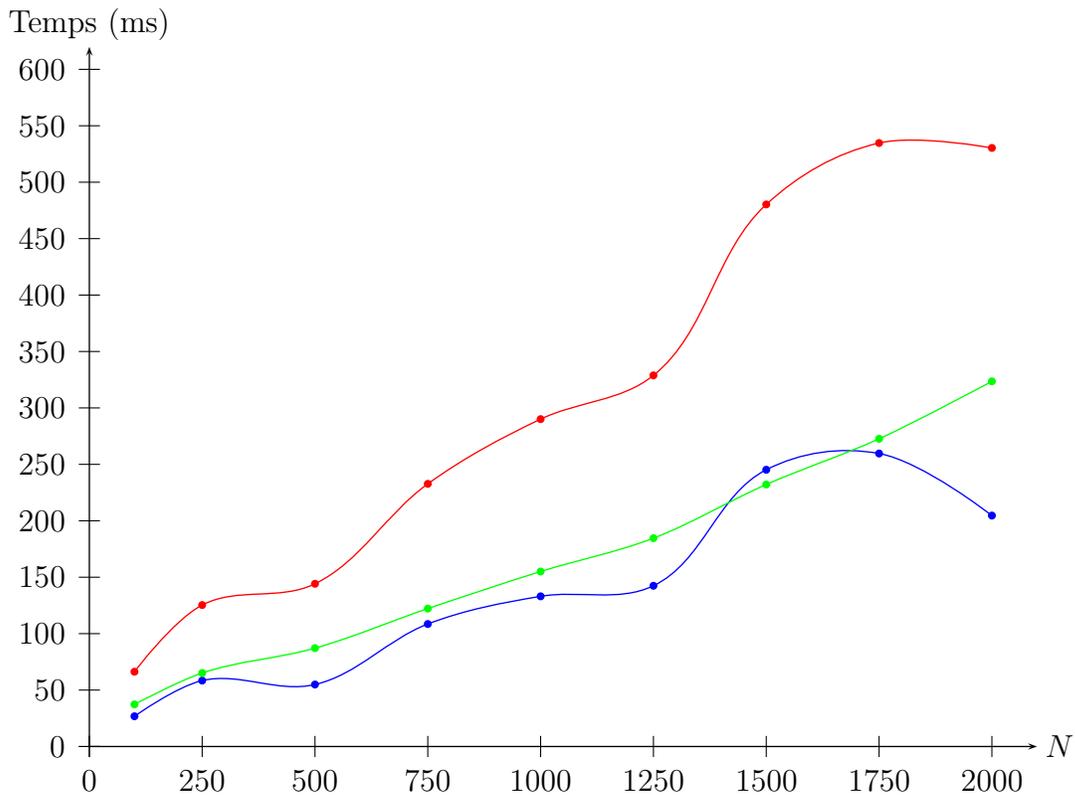


FIG. 6.6: Évolution des temps de calcul des algorithmes de prédiction en fonction de N . Les courbes bleue, verte et rouge correspondent respectivement aux algorithmes 6.1, 6.3 et de prédiction.

Comme dans les évaluations du problème de navigation, les algorithmes sont linéaires en N , excepté le classement (6.6) et la sélection (6.7) qui sont constants. Ces deux algo-

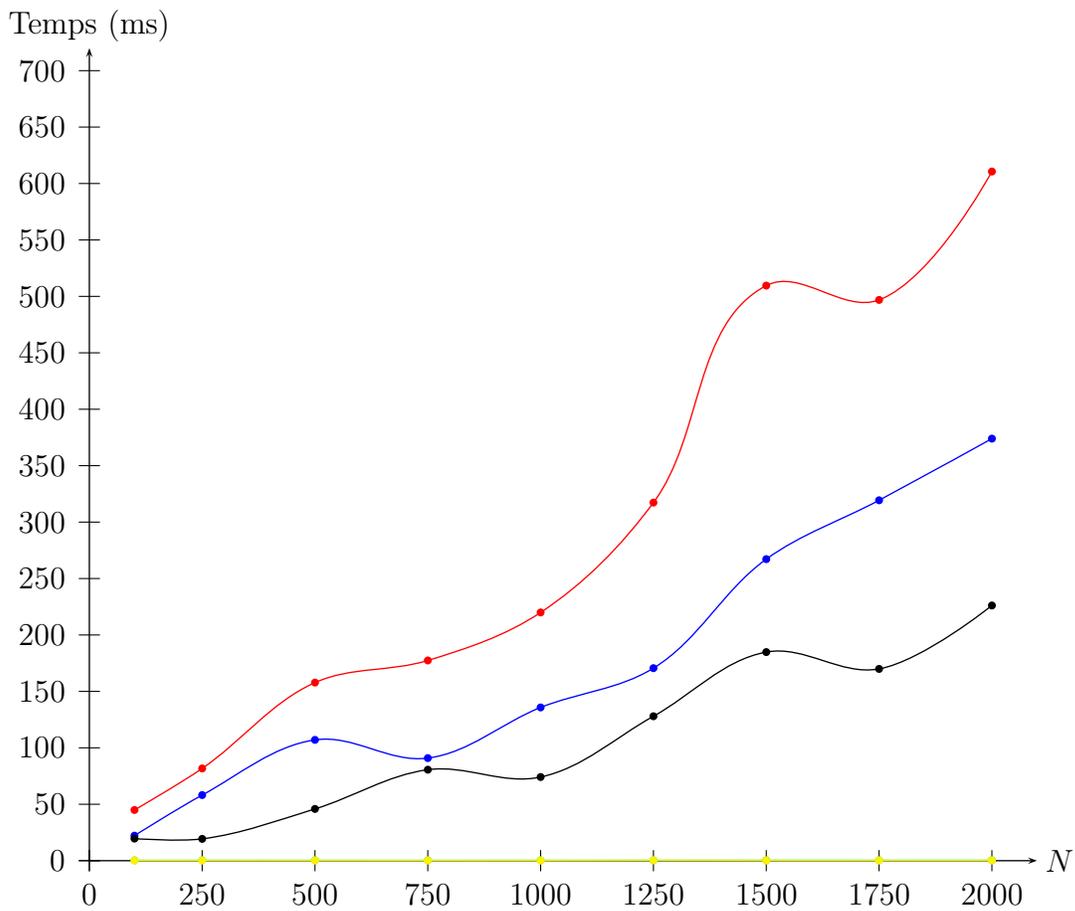


FIG. 6.7: Évolution des temps de calcul des algorithmes de recalage en fonction de N . Les courbes bleue, noire, verte (confondue avec la jaune), jaune et rouge correspondent respectivement aux algorithmes 6.4, 6.5, 6.6, 6.7 et de recalage.

rithmes ont d'ailleurs des temps de calcul très faibles, du même ordre de grandeur que dans le cas de la navigation, même si le réseau du thermostat contient deux places symboliques.

4.3 Complexité en fonction de la taille du système

Dans ce paragraphe, nous évaluons de façon expérimentale la complexité des algorithmes de l'estimation en fonction de la taille du système, et donc de la taille du réseau. Les complexités théoriques calculées sont données en fonction de $|P|$ et $|T|$. Les temps de calcul obtenus seront données en fonction de $|\mathcal{R}| = |P| + |T|$, le nombre de nœuds dans le réseau.

4.3.1 Problème de navigation

Pour évaluer ce temps de calcul, nous avons procédé à une série de tests avec un nombre de particules $N = 100$ (figures 6.8 et 6.9). Le temps de calcul donné est une moyenne des temps de calcul des différents algorithmes sur un échantillon de 20 observations, avec un bruit fixe de $0.1I$, sur les instances Nav01 à Nav18.

Les temps de calcul de l'évolution isoparticule (algorithme 6.1) et de la prédiction approchent une trajectoire polynomiale en $|\mathcal{R}|$ (leur complexité est en $|\mathcal{R}|^4$). On peut remarquer que dans tous les cas simulés, l'évolution isoparticule termine.

Le temps de calcul de l'évolution isomarquage (algorithme 6.3) semble constant en fonction de la taille du réseau, alors que sa complexité théorique est linéaire en fonction du nombre de places. L'allure de son temps de calcul peut s'expliquer par le fait que, de manière générale, les particules sont contenues dans un nombre restreint de places (pour le problème de navigation, quatre ou cinq places au maximum) et donc un grand nombre de places sont vides. Ceci rend l'algorithme indépendant, en pratique, de la taille du réseau.

Pour les mêmes raisons que précédemment, les temps de calcul de la pondération (algorithme 6.4), du rééchantillonnage (algorithme 6.5) et du recalage semblent constants : théoriquement, ces algorithmes dépendent du nombre de places dans le réseau, mais en pratique les particules sont contenues dans un petit nombre de places.

Les algorithmes de classement (6.6) et de sélection (6.7) ont eux un temps de calcul très faible du fait de l'absence de places symboliques dans le réseau.

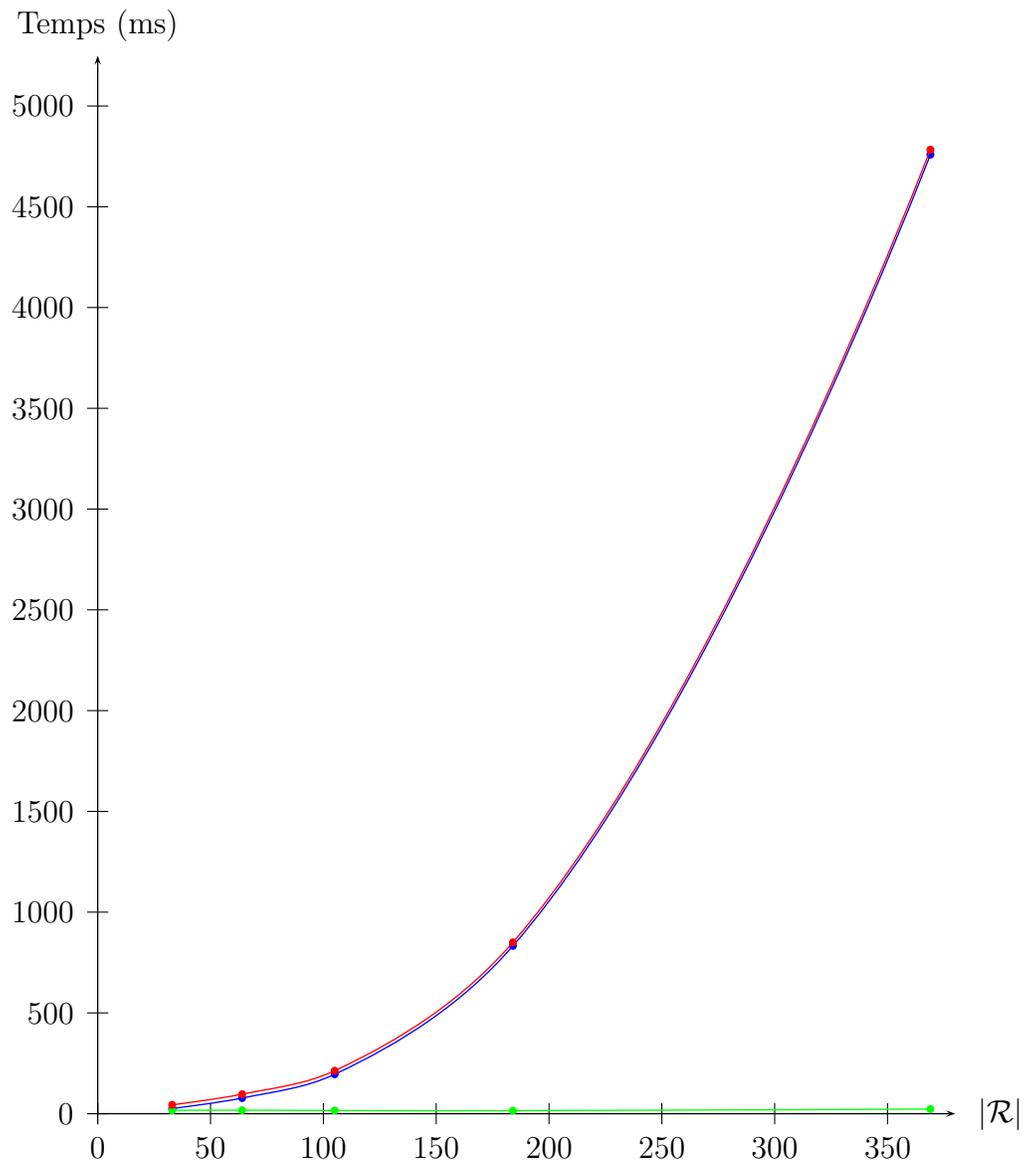


FIG. 6.8: Évolution des temps de calcul des algorithmes de prédiction en fonction de la taille du réseau, pour $N = 100$. Les courbes bleue, verte et rouge correspondent respectivement aux algorithmes 6.1, 6.3 et de prédiction.

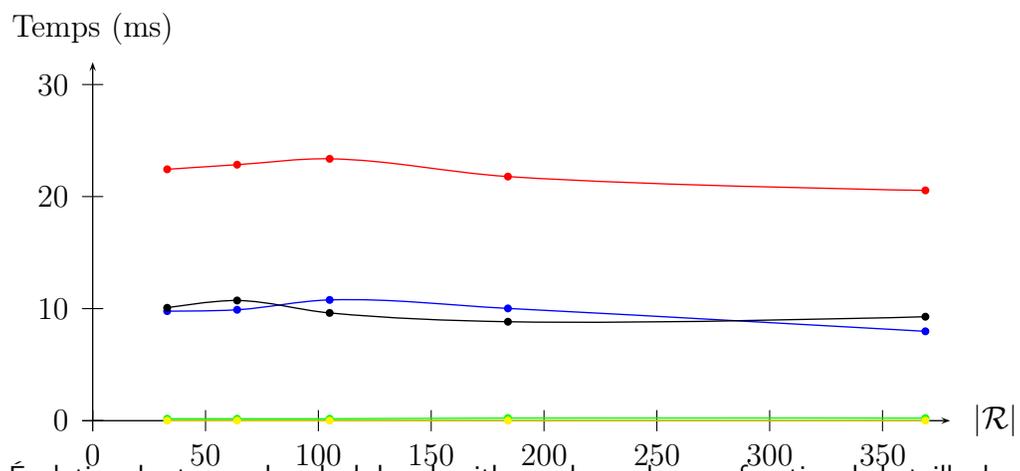


FIG. 6.9: Évolution des temps de calcul des algorithmes de recalage en fonction de la taille du réseau, pour $N = 100$. Les courbes bleue, noire, verte, jaune et rouge correspondent respectivement aux algorithmes 6.4, 6.5, 6.6, 6.7 et de recalage.

4.3.2 Le thermostat

Les instances du problème du thermostat diffèrent par la dynamique du système, mais la taille du système est la même. Afin d'évaluer le comportement des algorithmes en fonction de la taille du réseau, l'estimation est lancée sur plusieurs instances.

Ainsi, les figures 6.10, 6.11 et 6.12 représentent l'évolution des algorithmes d'estimation en fonction de la taille du réseau. L'instance *Termo01* a une taille de 9 (5 places et 4 transitions), et les différents points obtenus correspondent à l'estimation simultanée de plusieurs instances (2 instances ont une taille de 18, 3 instances de 27. . .) avec $N = 100$.

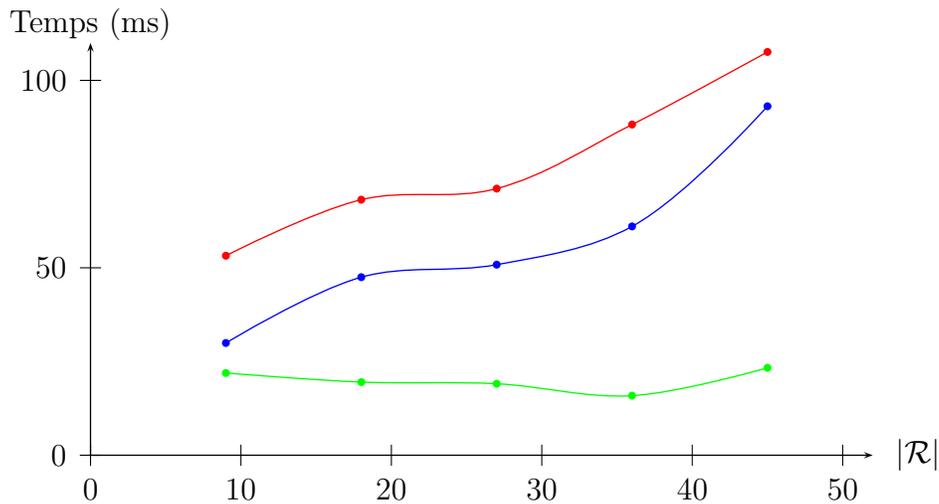


FIG. 6.10: Évolution des temps de calcul des algorithmes de prédiction en fonction de la taille du réseau, pour $N = 100$. Les courbes bleue, verte et rouge correspondent respectivement aux algorithmes 6.1, 6.3 et de prédiction.

Comme nous l'avons déjà observé dans le cas de la navigation, l'évolution isomarque a un temps de calcul constant. L'évolution isoparticule (et donc la prédiction) a une évolution linéaire en $|\mathcal{R}|$.

Les algorithmes de recalage numérique ont une évolution à peu près constante en la taille du problème, comme dans le cas du problème de navigation.

Nous pouvons voir ici, même si les temps de calcul sont très faibles, que les algorithmes de classement (6.6) et de sélection (6.7) ont une évolution linéaire en la taille du problème. Cette évolution observée est meilleure que celle espérée par leur complexité théorique (quadratique). Ceci est notamment dû aux faibles temps de calcul et à un petit nombre de places symboliques dans les instances, qui rendent difficile une réelle évaluation de la complexité. Cependant, on peut constater que la complexité pratique des algorithmes est

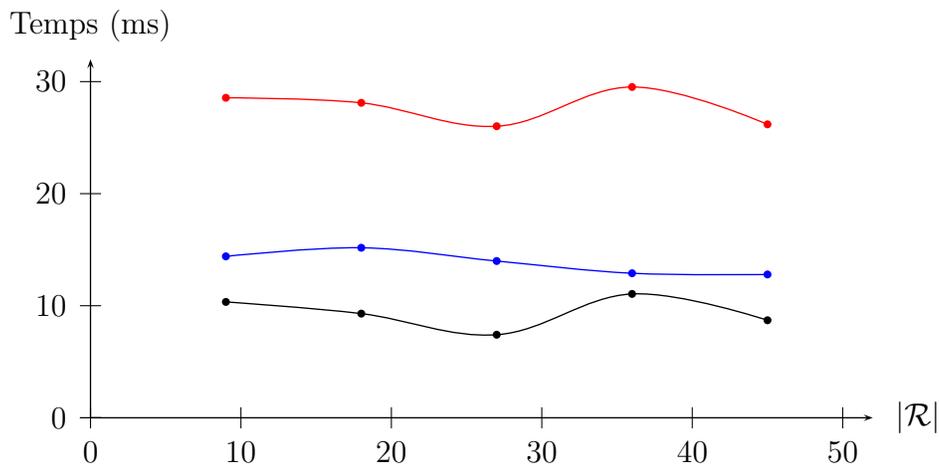


FIG. 6.11: Évolution des temps de calcul des algorithmes de recalage numérique en fonction de la taille du réseau, pour $N = 100$. Les courbes bleue, noire et rouge correspondent respectivement aux algorithmes 6.4, 6.5 et de recalage.

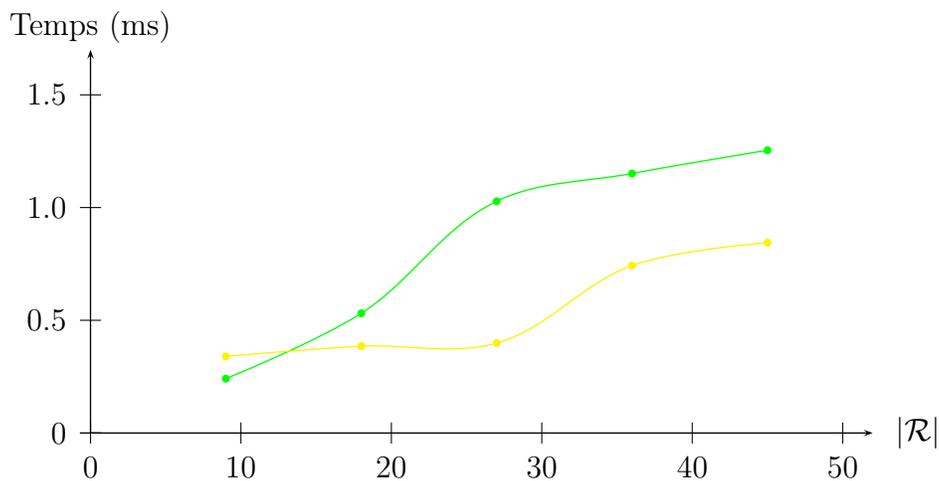


FIG. 6.12: Évolution des temps de calcul des algorithmes de recalage symbolique en fonction de la taille du réseau, pour $N = 100$. Les courbes verte et jaune correspondent respectivement aux algorithmes 6.6 et 6.7.

raisonnable (inférieure ou du même ordre de grandeur que leur complexité théorique).

5 Conclusion

Dans ce chapitre, nous avons décrit les algorithmes correspondant au principe d'estimation défini dans le chapitre 5. Nous nous sommes intéressé à la terminaison des algorithmes et avons calculé leur complexité.

Seule la terminaison de l'algorithme de l'évolution isoparticule est indécidable dans un cas général, cette évolution dépendant fortement de la structure du réseau et du marquage. Nous avons cependant observé sur les exemples de navigation et du thermostat que l'algorithme terminait dans un temps qui permet au processus d'estimation de pouvoir alterner prédiction et recalage.

Le tableau 6.2 reprend les différents algorithmes, leur complexité théorique et leur comportement mesuré.

Algorithme	Complexité théorique	Comportement en fonction	
		de N	de $ \mathcal{R} $
Isoparticule (6.1)	$\mathcal{O}(N \cdot \mathcal{R} ^4)^2$	linéaire	polynomial
Isomarquage (6.3)	$\mathcal{O}(N \cdot \mathcal{R})$	linéaire	constant
Prédiction	$\mathcal{O}(N \cdot \mathcal{R} ^4)$	linéaire	polynomial
Pondération (6.4)	$\mathcal{O}(N \cdot \mathcal{R})$	linéaire	constant
Rééchantillonnage (6.5)	$\mathcal{O}(N \cdot \mathcal{R})$	linéaire	constant
Classement (6.6)	$\mathcal{O}(\mathcal{R} ^2)$	constant	linéaire
Sélection (6.7)	$\mathcal{O}(\mathcal{R} ^2)$	constant	linéaire
Recalage	$\mathcal{O}(N \cdot \mathcal{R} ^2)$	linéaire	constant ³

TAB. 6.2: Complexités théoriques et mesurées des algorithmes d'estimation.

On peut remarquer que la structure de réseau de Petri particulière ne détériore en rien les performances des algorithmes inspirés du filtrage particulière. En effet, les algorithmes de prédiction et de recalage d'un filtre particulière ont une complexité linéaire en N et polynomiale en la taille du système (Karlsson *et al.*, 2005).

Les résultats expérimentaux obtenus dans ce chapitre sont très satisfaisants : ils permettent une modélisation d'un système de grande taille (81 places et 288 transitions pour les instances Nav16-18) et avec un nombre de particules élevé (quelques milliers) sans que les algorithmes ne deviennent trop complexes en pratique.

²La complexité donnée pour l'évolution isoparticule est une estimation faite sur les cas rencontrés et ne correspond pas à un calcul fondé puisque la terminaison de l'algorithme n'est pas démontrée.

³La complexité observée de l'algorithme de recalage est constante en fonction de la taille du problème, ce qui peut paraître aberrant, vu qu'une partie de cet algorithme (le recalage symbolique) a une complexité linéaire. Cependant, les temps de calcul très faibles du recalage symbolique face au recalage numérique absorbent cette évolution linéaire.

Chapitre 7

Diagnostic et détection d'incohérences

L'étude des systèmes hybrides et de leur comportement passe par l'élaboration d'un modèle (un réseau de Petri particulière) et d'un principe d'estimation, que nous avons décrit dans le chapitre 5. À partir de ce principe, nous pouvons fournir, après chaque observation, une estimation de l'état hybride du système observé. Cette estimation permet de connaître le comportement du système. L'analyse de cette estimation permet alors de détecter des comportements défailants ou incohérents du système. Dans ce chapitre, nous décrivons ce principe de diagnostic d'un système hybride.

1 Estimation de mode

Comme nous l'avons vu dans le chapitre précédent, le processus d'estimation permet à chaque instant d'avoir une connaissance sur l'état courant du système. Cette connaissance est représentée par un marquage estimé $\widehat{\mathcal{M}}_{k|k}$ à l'instant k du réseau de Petri particulière.

Ce marquage est hybride : il contient un ensemble de particules, qui représentent des valeurs possibles du vecteur d'état numérique du système, et un ensemble de jetons, dont le marquage représente un ensemble de modes possibles du système.

La détection d'incohérences, proposée dans ce chapitre, va essentiellement utiliser une confrontation des particules estimées et des modes possibles pour déterminer quelles situations sont conflictuelles dans le comportement du système. Cette confrontation va donc comparer les configurations des places marquées (les modes symboliques possibles), aux modes numériques possibles, déterminés à partir des particules estimées.

C'est cette **estimation de mode** que nous définissons dans ce paragraphe : à partir de l'ensemble des particules estimées à l'instant k , nous proposons d'évaluer les modes de fonctionnement numériques possibles du système.

1.1 Recalage numérique

Soit $\widehat{\mathcal{M}}_{k|k-1}$ le marquage prédit à l'instant k et \mathbf{z}_k la mesure numérique à l'instant k . Le processus de recalage numérique :

1. pondère les particules à partir de la fonction de vraisemblance ;
2. sélectionne N particules en fonction de ces poids.

L'ensemble des particules représente la distribution de probabilité du vecteur d'état numérique du système :

$$\mathbf{P}(\mathbf{x}_k = x) = \sum_{\pi_{k|k-1}^i \in \widehat{\mathcal{M}}_{k|k-1}} \omega_k^i \cdot \delta_x(\pi_{k|k-1}^i) \quad (7.1)$$

Ainsi, la pondération des particules présentes dans chaque place numérique permet d'estimer le mode numérique du système :

$$\forall p \in P_N, \mathbf{P}(\mathbf{x}_k \in \widehat{\mathcal{M}}_{k|k-1}(p)) = \sum_{\pi_{k|k-1}^i \in \widehat{\mathcal{M}}_{k|k-1}(p)} \omega_k^i \quad (7.2)$$

1.2 Estimation du mode numérique

Définition 7.1 (Estimation du mode numérique)

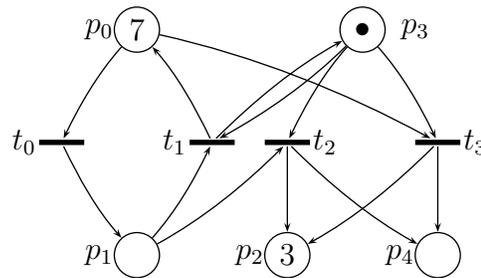
L'estimation du mode numérique du système à l'instant k est une fonction

$$E_k : P_N \mapsto [0, 1]$$

qui à chaque place numérique associe la probabilité du mode numérique associé, définie par :

$$\forall p \in P_N, E_k(p) = \mathbf{P}(\mathbf{x}_k \in \widehat{\mathcal{M}}_{k|k}(p)) \quad (7.3)$$

L'estimation de mode numérique et le classement des configurations fait lors du recalage permettent ainsi de déterminer les modes de fonctionnement du système les plus vraisemblables à chaque instant.

FIG. 7.1: Marquage recalé $\widehat{\mathcal{M}}(1|1)$.

Exemple 7.2 Reprenons l'exemple du thermostat et le marquage recalé présenté par la figure 7.1 (chapitre 5, section 2.2.4).

Dans cet exemple, l'estimation du mode numérique est :

$$\begin{aligned} E_1(p_0) &= \frac{7}{10} \\ E_1(p_1) &= 0 \\ E_1(p_2) &= \frac{3}{10} \end{aligned} \tag{7.4}$$

C'est-à-dire que le mode numérique le plus probable est p_0 (chauffage en fonctionnement) avec une probabilité de 0.7, puis le mode numérique p_2 (chauffage éteint) avec une probabilité de 0.3. Le mode numérique p_1 a une probabilité nulle.

Cette estimation du mode de fonctionnement du système permet, comme nous allons le voir par la suite, de déterminer les comportements corrects, défaillants ou conflictuels du système.

2 Diagnostic à base de modèles

Le diagnostic à base de modèles repose, comme son nom l'indique, sur la modélisation du comportement du système, y compris de son comportement défaillant. Cette technique est la plus utilisée dans le domaine du diagnostic (voir chapitre 1, section 2, et chapitre 2, section 4).

2.1 Modélisation des modes défaillants

Le diagnostic à base de modèles repose sur l'estimation du comportement du système et l'évaluation de ce comportement (ou du plus probable par exemple) relativement à des défaillances. Il est donc nécessaire de définir, *a priori*, un ensemble d'états qui correspondent à des comportements défaillants du système.

Dans le cas des réseaux de Petri particuliers, la définition d'un état du système revient à définir un marquage du réseau. Un comportement défaillant peut correspondre à une défaillance dans la dynamique continue du système (au niveau des places numériques), dans la configuration du système (au niveau des places symboliques), ou une mauvaise adéquation des deux comportements.

Définition 7.3 (Modes défaillants)

L'ensemble des modes défaillants d'un système est $D \subset P_N \cup P_S$. Cet ensemble contient les marquages (partiels) représentant des comportements défaillants du système.

2.2 Diagnostic de comportements défaillants

Dans ce paragraphe, nous proposons des méthodes de diagnostic de comportements défaillants à partir de la définition précédente. Ces méthodes sont présentées pour montrer que les réseaux de Petri particuliers peuvent être utilisés dans une approche de diagnostic à base de modèles. Cependant, l'objectif et l'apport principal de ces travaux est de développer des méthodes de détection d'incohérences, c'est-à-dire de défaillances non explicitement modélisées, qui seront exposées dans la section suivante.

Définition 7.4 (Défaillance numérique)

Soit \mathcal{R} un réseau de Petri particulière et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé à l'instant k . Soit D l'ensemble des modes défaillants du système modélisé par \mathcal{R} .

*Si $p = \operatorname{argmax}_{q \in P_N} E_k(q) \in D$ alors une **défaillance numérique** en p est diagnostiquée.*

Une défaillance numérique est détectée lorsque le mode numérique le plus probable est un mode défaillant. Dans le problème de navigation, le mode numérique défaillant est la place B , qui représente la case de la grille sur laquelle se déplace le mobile que l'on veut éviter.

Définition 7.5 (Défaillance symbolique)

Soit \mathcal{R} un réseau de Petri particulière et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé à l'instant k . Soit D l'ensemble des modes défaillants du système modélisé par \mathcal{R} .

Si $\exists p \in D$ qui soit une racine de la relation \geq_k^Δ , alors une **défaillance symbolique** en p est diagnostiquée.

Une défaillance symbolique est détectée lorsque la configuration correspondant à un mode symbolique défaillant est la configuration préférée.

Définition 7.6 (Défaillance hybride)

Soit \mathcal{R} un réseau de Petri particulière et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé à l'instant k . Soit D l'ensemble des modes défaillants du système modélisé par \mathcal{R} .

Si $\exists (p, q) \in D^2$ tels que :

- $p = \operatorname{argmax}_{r \in P_N} E_k(r)$

- $q \in P_S$ est une racine de la relation \geq_k^Δ

alors une **défaillance hybride** en (p, q) est diagnostiquée.

Une défaillance hybride correspond à un couple défaillant, dont la place numérique correspond au mode numérique le plus probable et la place symbolique à la configuration préférée.

3 Détection et prédiction d'incohérences

Le diagnostic de comportements défaillants permet de détecter des cas défectueux connus et modélisables. Cependant, dans de nombreuses applications, il est intéressant, voire indispensable, de pouvoir détecter des incohérences dans le comportement du système, même si ces incohérences n'ont pas été modélisées explicitement en tant que défaillances du système.

De plus, il peut également être intéressant de prédire les comportements défectueux, pour pouvoir anticiper leur arrivée, et éventuellement mettre en œuvre des comportements alternatifs permettant d'éviter une telle défaillance.

3.1 Incohérence numérique/symbolique

Dans le cadre de l'estimation de systèmes hybrides, le type d'incohérence le plus fréquent est une incohérence entre comportement numérique et comportement symbolique : le thermostat est éteint, mais le chauffage continue de fonctionner, ou la vitesse suivie par l'objet ne correspond pas à la zone dans laquelle il se trouve.

C'est ce genre d'incohérence que nous proposons de détecter lors de l'estimation du système. Pour détecter un comportement anormal, il suffit *a priori* de raisonner sur les comportements normaux du système.

Ces comportements normaux correspondent au modèle du système (le réseau de Petri particulière) et à sa dynamique (le jeu du réseau). Détecter une situation incohérente revient à détecter un état qui n'est pas "possible" au vu du modèle.

3.2 Graphe des marquages cohérents

Pour déterminer quels sont les comportements normaux (et en conséquence les comportements anormaux), nous proposons de construire l'ensemble des situations qui sont possibles au vu du modèle du système et de son état d'origine : il s'agit donc de construire le graphe des marquages accessibles du réseau à partir de son marquage initial.

Construire exactement le graphe du réseau particulière est complexe, puisqu'il s'agit de calculer les évolutions numériques possibles du vecteur d'état. Pour caractériser les comportements cohérents, nous réalisons une abstraction du système, en construisant *l'ensemble des modes cohérents du système*.

Rappel 7.7 (Réseau sous-jacent (définition 3.3))

Le réseau sous-jacent \mathcal{R}_{sj} d'un réseau de Petri particulière \mathcal{R} est le réseau ordinaire ayant la même structure que \mathcal{R} , c'est-à-dire privé des informations de type (places numériques/symboliques) et des événements, conditions, configurations, équations associés aux nœuds du réseau.

Définition 7.8 (Modes cohérents)

Soit \mathcal{R} un réseau de Petri particulière et \mathcal{M}_0 son marquage initial. L'ensemble des **modes cohérents** du système est le graphe \mathcal{G}_C des marquages accessibles du réseau sous-jacent \mathcal{R}_{sj} à partir de \mathcal{M}_0 .

Ainsi ce graphe nous permet d'identifier, avant de commencer l'estimation du système, tous les modes du système qui sont cohérents.

Exemple 7.9 Le graphe des marquages cohérents du thermostat, étudié au chapitre 3, section 2.2, est représenté sur la figure 7.2.

3.3 Graphe de recalage

Considérons un réseau de Petri particulière \mathcal{R} et $\widehat{\mathcal{M}}(k|k)$ son marquage estimé au temps k . L'étape de recalage à l'instant k a permis, entre autres, d'évaluer les particules

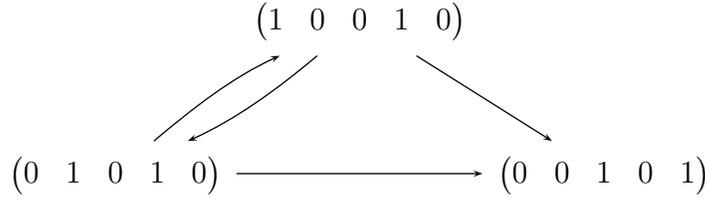


FIG. 7.2: Graphe des marquages cohérents du thermostat.

correspondant le mieux à l'observation et de classer les places symboliques selon une relation de préférence.

Ces informations permettent de construire un *graphe de recalage*, contenant un ensemble de marquages hybrides, construits à partir d'une particule et d'une place symbolique.

Définition 7.10 (Graphe de recalage)

Soit \mathcal{R} un réseau de Petri particulière, $\mathcal{M}(k|k-1)$ et $\widehat{\mathcal{M}}(k|k)$ ses marquages prédit et recalé à l'instant k , ω_k^i les poids des particules prédites et \geq_k^Δ la relation de préférence sur les places symboliques.

Le **graphe de recalage** à l'instant k est \mathcal{G}_k , dont les nœuds sont tous les couples $(\pi_{k|k-1}^i, p) \in \widehat{\mathcal{M}}(k|k-1) \times P_S$, p marquée, et les arcs du graphe sont définis par :

$$A((\pi_{k|k-1}^i, p), (\pi_{k|k-1}^j, q)) = 1 \Leftrightarrow \omega_k^i \geq \omega_k^j \text{ et } p \geq_k^\Delta q \quad (7.5)$$

où A est la matrice d'adjacence du graphe (et donc $A(a, b) = 1$ représente un arc de a vers b).

Le graphe de recalage contient donc tous les couples composés d'un état numérique possible du système (une particule) et d'une configuration possible du système¹. Dans ce graphe, un couple est préféré à un autre si sa particule a un poids supérieur et sa place symbolique est préférée.

Remarque 7.11 (Graphe de recalage)

Le graphe de recalage est construit sur la base des poids des particules prédites, et non des particules recalées. Ceci est dû au fait que les particules recalées ont toutes un poids égal ($\frac{1}{N}$), et le classement résultant n'a donc que peu d'intérêt.

¹La relation de préférence \geq_k^Δ porte sur les places symboliques du réseau de Petri particulière. Cette relation est basée sur la valeur de la configuration associée à chaque place, et, par abus de langage, nous utiliserons indifféremment une place symbolique ou sa configuration.

Exemple 7.12 *Le tableau 7.1 donne les valeurs des poids des particules prédites à l'instant 1, représentées sur la figure 5.6, au cours de l'estimation du thermostat (chapitre 5, section 2.2.4).*

Particule	π^1	π^2	π^3	π^4	π^5	π^6	π^7	π^8	π^9	π^{10}
Place	p_0					p_1				
Poids ($\cdot 10^{-3}$)	169	2	178	47	160	0	14	8	37	14
Particule	π^{11}	π^{12}	π^{13}	π^{14}	π^{15}	π^{16}	π^{17}	π^{18}	π^{19}	π^{20}
Place	p_2									
Poids ($\cdot 10^{-3}$)	1	7	2	52	6	19	153	12	110	7

TAB. 7.1: Poids et places des particules prédites à l'instant 1.

La relation de préférence donne

$$p_3 \geq_1^\Delta p_4 \quad (7.6)$$

(équation (5.25)), et par conséquent, le graphe de recalage contient tous les couples (π^i, p_j) avec $i \in \{1, \dots, 20\}$, $j \in \{3, 4\}$, soit 40 nœuds.

Il est évident qu'étudier un tel graphe devient vite laborieux, surtout pour essayer d'y repérer (éventuellement à "l'oeil") des situations anormales. Dans un tel graphe, de nombreuses particules représentent des informations très proches et la détection de situations incohérentes porte sur les *modes* de fonctionnement du système. La valeur de chaque particule n'est donc pas significative pour cette détection, mais c'est la place qu'elle marque, et donc le mode numérique qu'elle représente, qui importe.

Ainsi nous proposons de simplifier largement ce graphe de recalage, de façon à pouvoir l'étudier et appréhender sa structure de façon "allégée".

3.4 Stratégies de regroupement

3.4.1 Regroupement par place

La détection d'un comportement incohérent repose sur la comparaison entre mode numérique et mode symbolique. L'information pertinente apportée par une particule, au sein du graphe de recalage, est la place que cette particule marque, qui permet d'identifier le mode de fonctionnement numérique.

Ainsi, simplifier le graphe de recalage peut se faire en regroupant toutes les particules marquant la même place, et les nœuds correspondants du graphe de recalage. Ce regroupement va construire des nœuds $(p, q) \in P_N \times P_S$, avec un poids de la place numérique p égal à la somme des poids de ses particules.

Définition 7.13 (Graphe de recalage par places)

Soit \mathcal{R} un réseau de Petri particulière, $\widehat{\mathcal{M}}(k|k)$ son marquage recalé à l'instant k et \geq_k^Δ la relation de préférence sur les places symboliques.

Le **graphe de recalage par places** à l'instant k est $\mathcal{G}_{k/P}$, dont les nœuds sont tous les couples de places marquées $(p, q) \in P_N \times P_S$, et dont les arcs sont définis par :

$$A((p_1, q_1), (p_2, q_2)) = 1 \Leftrightarrow E_k(p_1) \geq E_k(p_2) \text{ et } q_1 \geq_k^\Delta q_2 \quad (7.7)$$

Un couple est meilleur qu'un autre si le mode de sa place numérique est plus probable et sa place symbolique est préférée.

Remarque 7.14 (Poids des particules et poids des places)

La remarque 7.11 a mis en avant l'utilisation des particules prédites, dont les poids diffèrent, alors que les poids des particules recalées sont tous égaux. Cependant, lorsque l'on regroupe les particules par places, les deux notions se confondent² et le poids d'une place numérique (l'estimation de son mode) correspond à la somme des poids des particules, c'est-à-dire au nombre de particules de cette place pour le marquage recalé.

Utiliser les particules recalées a l'avantage de présenter un graphe de recalage plus "propre", puisque les états qui ne correspondent pas à l'observation ont été supprimés par l'étape de rééchantillonnage. En revanche, l'utilisation des particules prédites fournit une information plus complète, notamment sur l'évolution possible du système, et cette information peut permettre de voir "émerger" un comportement incohérent.

Exemple 7.15 Dans le cas du thermostat, le graphe de recalage par places à l'instant 1, issu des équations (7.4) et (7.6), est présenté sur la figure 7.3.

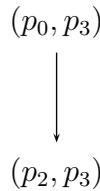


FIG. 7.3: Graphe de recalage par places du thermostat

²Le poids d'une place numérique n'est pas, dans la pratique, identique pour le marquage recalé ou le marquage prédit. Cependant, de façon asymptotique (lorsque N tend vers l'infini), la différence entre ces deux poids tend vers 0. On peut donc considérer qu'en moyenne, l'écart entre les deux poids est assez faible.

Dans ce graphe, le couple (p_0, p_3) est préféré puisque p_0 a une meilleure probabilité (0.7 contre 0.3). Les couples contenant les places p_1 et p_4 n'apparaissent pas puisque ces places ne sont pas marquées dans le marquage recalé $\widehat{\mathcal{M}}(k|k)$.

Le graphe de recalage par places est succinct. Dans le cas du thermostat par exemple, il contiendra le plus souvent un ou deux couples, et apporte de ce fait une information réduite sur le comportement du système.

Cette information réduite permet donc d'avoir un aperçu des différents modes de fonctionnement du système et de leur classement, ce qui peut permettre à un opérateur humain, par exemple, d'évaluer son comportement d'un simple coup d'oeil. Cela peut également permettre de détecter facilement des situations anormales.

Cependant, le graphe a une taille limitée (par le nombre de places du réseau). De plus, les modes anormaux, qui ne sont pas explicitement modélisés, risquent d'être difficilement identifiables.

3.4.2 Regroupement par poids

Certains cas, éventuellement incohérents, peuvent ne pas apparaître dans le graphe de recalage par places, du fait d'une trop forte agrégation des particules. Par exemple, il est possible que la particule correspondant le mieux à l'observation soit située dans une place de poids global faible (par exemple parce qu'elle ne contient qu'une particule sur un millier), et se retrouve à une feuille du graphe de recalage (la "moins préférée" de toutes), alors qu'elle correspond à une défaillance.

Pour simplifier le graphe de recalage, sans perdre l'information associée aux poids des particules, nous proposons de regrouper les particules par classes, selon la relation R :

Théorème 7.16 (Regroupement par poids)

Soit \mathcal{G}_k le graphe de recalage à l'instant k et R la relation définie par :

$$\forall \pi_{k|k-1}^i, \pi_{k|k-1}^j, \quad \pi_{k|k-1}^i R \pi_{k|k-1}^j \Leftrightarrow \begin{cases} pl(\pi_{k|k-1}^i) = pl(\pi_{k|k-1}^j) \\ \forall \pi_{k|k-1}^l / pl(\pi_{k|k-1}^l) \neq pl(\pi_{k|k-1}^j), \\ \quad \omega_k^l < \omega_k^i \Rightarrow \omega_k^l < \omega_k^j \\ \quad \omega_k^l < \omega_k^j \Rightarrow \omega_k^l < \omega_k^i \end{cases} \quad (7.8)$$

où $pl(\pi)$ est la place marquée par π .

R est une **relation d'équivalence** sur \mathcal{G}_k .

La relation R regroupe les particules en classes, de sorte que :

- les particules d'une même classe marquent la même place ;
- aucune autre particule n'a un poids strictement compris entre les poids de deux particules de cette classe.

Démonstration 7.17 (Relation d'équivalence)

Réflexivité R est clairement réflexive, puisque $pl(\pi^i) = pl(\pi^i)$ et que $w^l < w^i \Rightarrow w^l < w^i$.

Donc

$$\forall \pi_{k|k-1}^i, \pi_{k|k-1}^i R \pi_{k|k-1}^i \quad (7.9)$$

Symétrie R est clairement symétrique, puisque $pl(\pi^i) = pl(\pi^j) \Rightarrow pl(\pi^j) = pl(\pi^i)$, et que les deux implications sur les poids sont symétriques l'une de l'autre. Donc

$$\forall \pi_{k|k-1}^i, \pi_{k|k-1}^j \quad \pi_{k|k-1}^i R \pi_{k|k-1}^j \Rightarrow \pi_{k|k-1}^j R \pi_{k|k-1}^i \quad (7.10)$$

Transitivité Soient π^i, π^j, π^l telles que $\pi^i R \pi^j$ et $\pi^j R \pi^l$. Alors :

- $pl(\pi^i) = pl(\pi^j) = pl(\pi^l)$,
- Soit π^u telle que $pl(\pi^u) \neq pl(\pi^i)$, alors

$$\begin{aligned} w^u < w^i &\Rightarrow w^u < w^j \Rightarrow w^u < w^l \\ w^u < w^l &\Rightarrow w^u < w^j \Rightarrow w^u < w^i \end{aligned}$$

et donc $\pi^i R \pi^l$. Donc

$$\forall \pi_{k|k-1}^i, \pi_{k|k-1}^j, \pi_{k|k-1}^l, \pi_{k|k-1}^i R \pi_{k|k-1}^j \text{ et } \pi_{k|k-1}^j R \pi_{k|k-1}^l \Rightarrow \pi_{k|k-1}^i R \pi_{k|k-1}^l \quad (7.11)$$

R est transitive.

En conclusion, R est réflexive, symétrique et transitive, c'est une relation d'équivalence sur \mathcal{G}_k .

Remarque 7.18 (Partition)

La notion de relation d'équivalence permet de construire l'ensemble des classes d'équivalence, qui forment une partition de l'ensemble des particules : chaque particule appartient à une et une seule classe d'équivalence.

À chaque classe d'équivalence de la relation R on associe un poids égal à la somme des poids de ses particules. La table 7.2 expose les classes issues du regroupement par poids des particules de la table 7.1.

À partir de 20 particules, nous avons obtenu 12 classes. Ce regroupement permet de conserver les informations sur les poids des particules, mais ne réduit pas suffisamment le graphe de recalage.

Appliquons à nouveau la relation R sur les classes du tableau 7.2 (table 7.3).

Nous avons à nouveau réduit le nombre de classes à 9. Appliquer de nouveau la relation R ne modifie pas les classes.

Classes	Place	Poids ($\cdot 10^{-3}$)
$\{\pi^1, \pi^3, \pi^5\}$	p_0	507
$\{\pi^{14}, \pi^{17}, \pi^{19}\}$	p_2	315
$\{\pi^4\}$	p_0	47
$\{\pi^9\}$	p_1	37
$\{\pi^{16}\}$	p_2	19
$\{\pi^7, \pi^{10}\}$	p_1	28
$\{\pi^{18}\}$	p_2	12
$\{\pi^8\}$	p_1	8
$\{\pi^{12}, \pi^{13}, \pi^{15}, \pi^{20}\}$	p_2	22
$\{\pi^2\}$	p_0	2
$\{\pi^{11}\}$	p_2	1
$\{\pi^6\}$	p_1	0

TAB. 7.2: Classes d'équivalence, leurs places et leurs poids.

Classes	Place	Poids ($\cdot 10^{-3}$)
$\gamma_1 = \{\pi^1, \pi^3, \pi^5\}$	p_0	507
$\gamma_2 = \{\pi^{14}, \pi^{17}, \pi^{19}\}$	p_2	315
$\gamma_3 = \{\pi^4\}$	p_0	47
$\gamma_4 = \{\pi^9, \pi^7, \pi^{10}\}$	p_1	64
$\gamma_5 = \{\pi^8\}$	p_1	8
$\gamma_6 = \{\pi^{12}, \pi^{13}, \pi^{15}, \pi^{16}, \pi^{18}, \pi^{20}\}$	p_2	53
$\gamma_7 = \{\pi^2\}$	p_0	2
$\gamma_8 = \{\pi^{11}\}$	p_2	1
$\gamma_9 = \{\pi^6\}$	p_1	0

TAB. 7.3: Classes d'équivalence obtenues à partir des classes d'équivalence.

En appliquant successivement la relation R sur les particules puis sur les classes obtenues, il est donc possible de synthétiser l'information sur les modes numériques du système tout en disposant de plus d'éléments que dans le cas du regroupement par places. Nous pouvons alors définir le graphe de recalage par poids par :

Définition 7.19 (Graphe de recalage par poids)

Soit \mathcal{R} un réseau de Petri particulaire, Γ_k l'ensemble des classes d'équivalence obtenues par application successive de la relation R sur les particules prédites, et \geq_k^Δ la relation de préférence sur les places symboliques.

Le **graphe de recalage par poids** à l'instant k est $\mathcal{G}_{k/R}$, dont les nœuds sont les couples (γ, p) , avec γ une classe d'équivalence de R et $p \in P_S$ marquée, et dont les arcs sont définis par :

$$A((\gamma_1, p_1), (\gamma_2, p_2)) = 1 \Leftrightarrow \omega(\gamma_1) \geq \omega(\gamma_2) \text{ et } p_1 \geq_k^{\Delta} p_2 \quad (7.12)$$

où le poids d'une classe est défini par

$$\omega(\gamma) = \sum_{\pi_{k|k-1}^i \in \gamma} \omega_k^i \quad (7.13)$$

Le graphe de recalage par poids du thermostat est représenté sur la figure 7.4.

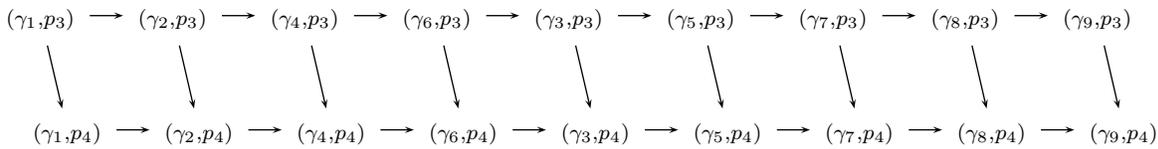


FIG. 7.4: Graphe de recalage par poids du thermostat.

3.5 Détection d'incohérences

Parmi tous les couples construits dans le graphe de recalage, certains correspondent à des états du système (plus ou moins vraisemblables selon l'observation), alors que d'autres sont impossibles. On dit que ces couples sont *incohérents*.

La notion de cohérence est définie aussi bien pour un couple $(\pi_{k|k-1}^i, q)$ du graphe de recalage, qu'un couple (p, q) du graphe de recalage par places ou un couple $(\gamma_{k|k-1}^j, q)$ du graphe de recalage par poids.

Définition 7.20 (Cohérence)

Soit \mathcal{R} un réseau de Petri particulière, \mathcal{G}_C son graphe des marquages cohérents et \mathcal{G}_k , $\mathcal{G}_{k/P}$ et $\mathcal{G}_{k/R}$ ses graphes de recalage, de recalage par places et de recalage par poids à

l'instant k . On définit la fonction de **cohérence** Ch à l'instant k par

$$\forall c = (\pi^i, q) \in \mathcal{G}_k, \quad Ch(c) = \begin{cases} 1 & \text{si } \exists \mathcal{M} \in \mathcal{G}_C \text{ tel que } \mathcal{M}(pl(\pi^i)) \geq 1 \text{ et } \mathcal{M}(q) \geq 1 \\ 0 & \text{sinon} \end{cases} \quad (7.14)$$

$$\forall c = (p, q) \in \mathcal{G}_{k/P}, \quad Ch(c) = \begin{cases} 1 & \text{si } \exists \mathcal{M} \in \mathcal{G}_C \text{ tel que } \mathcal{M}(p) \geq 1 \text{ et } \mathcal{M}(q) \geq 1 \\ 0 & \text{sinon} \end{cases} \quad (7.15)$$

$$\forall c = (\gamma^j, q) \in \mathcal{G}_{k/R}, \quad Ch(c) = \begin{cases} 1 & \text{si } \exists \mathcal{M} \in \mathcal{G}_C \text{ tel que } \mathcal{M}(pl(\gamma^j))^3 \geq 1 \text{ et } \mathcal{M}(q) \geq 1 \\ 0 & \text{sinon} \end{cases} \quad (7.16)$$

où $pl(\pi^i)$ est la place du réseau contenant la particule π^i dans le marquage $\widehat{\mathcal{M}}(k|k-1)$.

Un couple est cohérent s'il est couvert par un marquage du graphe des marquages cohérents du réseau. Autrement, l'état représenté par ce couple est inaccessible depuis l'état initial du système : il est **incohérent**.

À partir de la définition de la cohérence d'un couple, et en utilisant les graphes de recalage, nous pouvons désormais détecter des incohérences dans le comportement du système.

Rappel 7.21 (Noyau d'un graphe)

Le noyau N d'un graphe $G = (X, A)$ est un ensemble de nœuds vérifiant :

$$\forall x \in X \setminus N, \quad \exists n \in N / (n, x) \in A \quad (7.17)$$

$$\forall n, m \in N, \quad n \neq m \Rightarrow (n, m) \notin A \quad (7.18)$$

C'est-à-dire que tout nœud du graphe est précédé par un nœud du noyau, et que deux nœuds du noyau sont non comparables.

Dans le cas de relations d'ordre ou de préordre, le noyau du graphe représente les meilleurs nœuds de la relation.

³ $pl(\gamma^j)$ est la place contenant l'ensemble des particules de la classe, qui existe et est unique, par construction.

Définition 7.22 (Détection d'incohérence)

Soit \mathcal{R} un réseau de Petri particulière, Ch sa fonction de cohérence, $\mathcal{G}_{k/P}$ et $\mathcal{G}_{k/R}$ ses graphes de recalage par places et de recalage par poids à l'instant k .

Une **incohérence** est détectée à l'instant k s'il existe un couple dans le noyau de $\mathcal{G}_{k/P}$ ou de $\mathcal{G}_{k/R}$ qui soit incohérent.

Nous définissons une incohérence comme le fait que l'estimation la plus vraisemblable, d'après les estimations de modes (graphe de recalage par places) ou les pondérations des particules (graphe de recalage par poids), corresponde à un marquage inaccessible. Nous verrons par la suite, notamment à travers des exemples d'incohérences détectées, comment affiner cette détection et utiliser la structure des graphes de recalage pour prédire l'arrivée de situations incohérentes.

Exemple 7.23 Reprenons l'exemple du thermostat. Parmi les nœuds du graphe de recalage par places (figure 7.3), (p_0, p_3) est cohérent, car il correspond au marquage $[10010]$ du graphe de cohérence (figure 7.2), alors que (p_2, p_3) est incohérent.

Dans le graphe de recalage par poids, les couples incohérents sont les couples (γ_i, p_3) , $i \in \{2, 6, 8\}$ et (γ_j, p_4) , $j \in \{1, 2, 4, 5, 7, 9\}$.

Dans les deux graphes, les nœuds préférés (au sommet du graphe) sont cohérents : aucune incohérence n'est détectée.

Remarque 7.24 (Niveaux d'incohérence)

La définition de cohérence (et d'incohérence) présentée précédemment se base sur l'analyse des sommets des graphes de recalage. La détection peut a priori être affinée en étudiant la structure de ces graphes, et la position des marquages incohérents. Le fait qu'un marquage incohérent se "rapproche" du sommet du graphe au fil des observations peut ainsi conduire à prévoir une situation incohérente.

4 Conclusion

Nous avons essentiellement vu dans ce paragraphe une méthode de détection d'incohérences : cette détection permet de diagnostiquer des comportements anormaux sans les avoir modélisés explicitement.

Pour cela, suite à l'étape de recalage, nous définissons un graphe de recalage, que l'on peut condenser de deux façons différentes (par places ou par poids), et qui décrit l'ensemble des comportements hybrides possibles du système.

Le graphe de recalage par places représente les couples (mode numérique, mode symbolique) et se base sur l'estimation de mode numérique et la relation de préférence sur les configurations.

Le graphe de recalage par poids représente les couples (classes de particules, mode symbolique), et permet de représenter des cas qui peuvent être masqués par la construction du graphe de recalage par places.

Puis, à partir du graphe des marquages cohérents, qui contient l'ensemble des modes accessibles du système, nous évaluons la cohérence des nœuds de ces graphes, ce qui permet, lorsque les meilleurs nœuds sont incohérents, de détecter une incohérence dans le comportement du système.

Le chapitre 8 présente un certain nombre d'exemples illustrant le comportement du principe d'estimation et sa sensibilité à certains paramètres. Il présente également des résultats sur l'estimation de mode numérique et la détection d'incohérences dans les systèmes hybrides.

Nous verrons également par la suite comment l'étude de la structure des graphes de recalage, et non plus uniquement de leurs sommets, peut aider à la prédiction de situations incohérentes.

Chapitre 8

Simulations

Après avoir présenté le modèle des réseaux de Petri particuliers, le principe d'estimation par filtrage particulaire permettant d'estimer l'état d'un système hybride sous la forme d'un marquage du réseau, et un principe de diagnostic, basé sur une modélisation d'états défaillants et sur une détection d'incohérences, nous présentons dans ce chapitre les résultats obtenus sur les problèmes de navigation et du thermostat. Ces résultats vont nous permettre d'évaluer le comportement de l'estimateur, en fonction de différents critères.

1 Estimation de l'état d'un système

Dans ce paragraphe, nous présentons des résultats de simulations qui permettent d'évaluer le comportement de l'estimateur. Ces simulations sont réalisées sur les instances Nav01 (de taille 33) et Nav13 (de taille 184) du problème de navigation, et sur l'instance Thermo01 du thermostat.

Dans les figures ci-dessous, l'estimation représentée est la valeur moyenne des particules recalées et l'observation correspond à une version bruitée de l'état réel du système.

1.1 Navigation

Pour les instances Nav01 et Nav13, nous avons représenté l'évolution de la position de l'objet sur la grille (figures 8.1 et 8.4 respectivement), puis l'évolution de l'abscisse (figures 8.2 et 8.5 respectivement) et de l'ordonnée (figures 8.3 et 8.6 respectivement) de

l'objet.

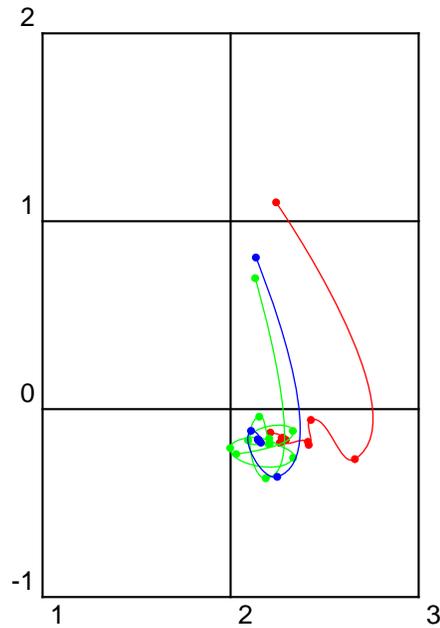


FIG. 8.1: Estimation de la position de l'objet pour Nav01, $N = 100$ et un bruit d'observation de $0.1I^1$. L'estimation est représentée en rouge, l'observation en vert et l'état réel de l'objet en bleu.

Sur les différentes figures, on peut remarquer que l'estimation approche assez bien le comportement réel du système.

¹Ce que nous appelons "bruit d'observation", par abus de langage, correspond en fait à l'écart-type du bruit d'observation, qui est un bruit blanc gaussien.

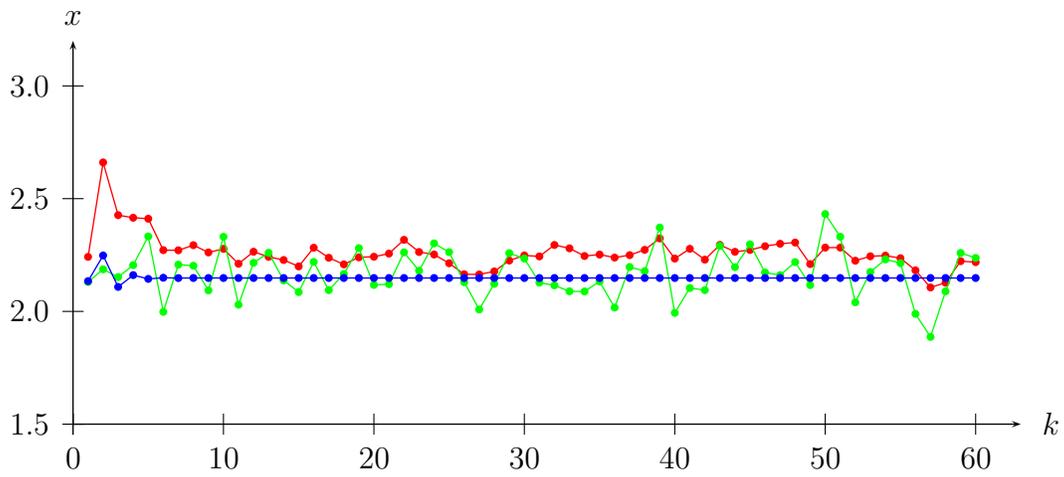


FIG. 8.2: Estimation de l'abscisse pour Nav01, $N = 100$ et un bruit d'observation de $0.1I$. L'estimation est représentée en rouge, l'observation en vert et l'état réel de l'objet en bleu.

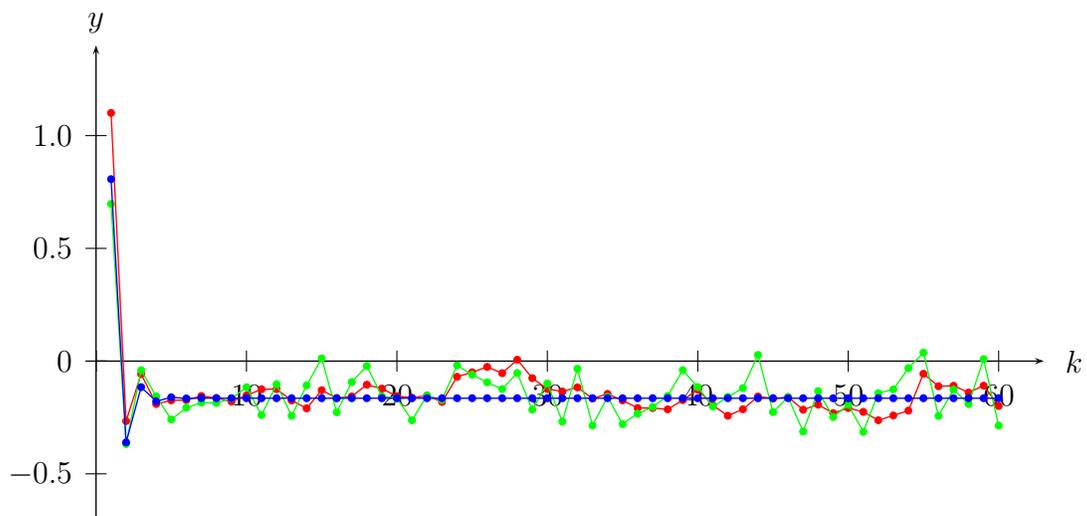


FIG. 8.3: Estimation de l'ordonnée pour Nav01, $N = 100$ et un bruit d'observation de $0.1I$. L'estimation est représentée en rouge, l'observation en vert et l'état réel de l'objet en bleu.

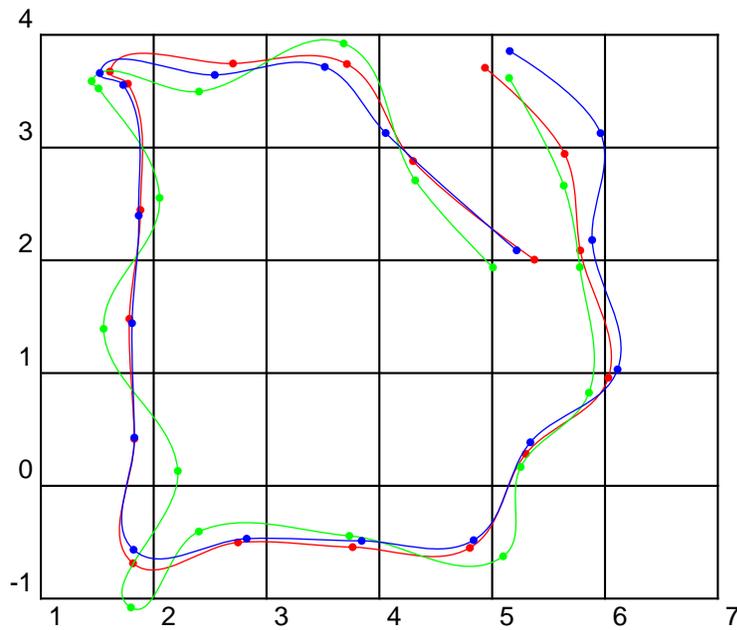


FIG. 8.4: Estimation de la position de l'objet pour Nav13, $N = 500$ et un bruit de $0.3I$. L'estimation est représentée en rouge, l'observation en vert et l'état réel de l'objet en bleu.

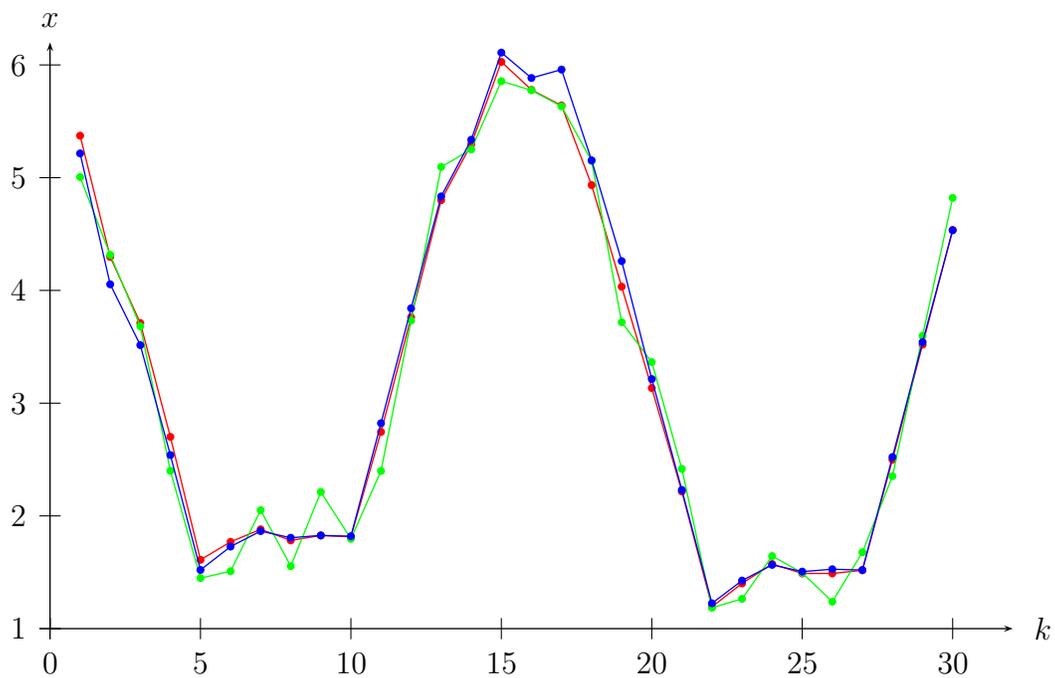


FIG. 8.5: Estimation de l'abscisse pour Nav13, $N = 500$ et un bruit de $0.3I$. L'estimation est représentée en rouge, l'observation en vert et l'état réel de l'objet en bleu.

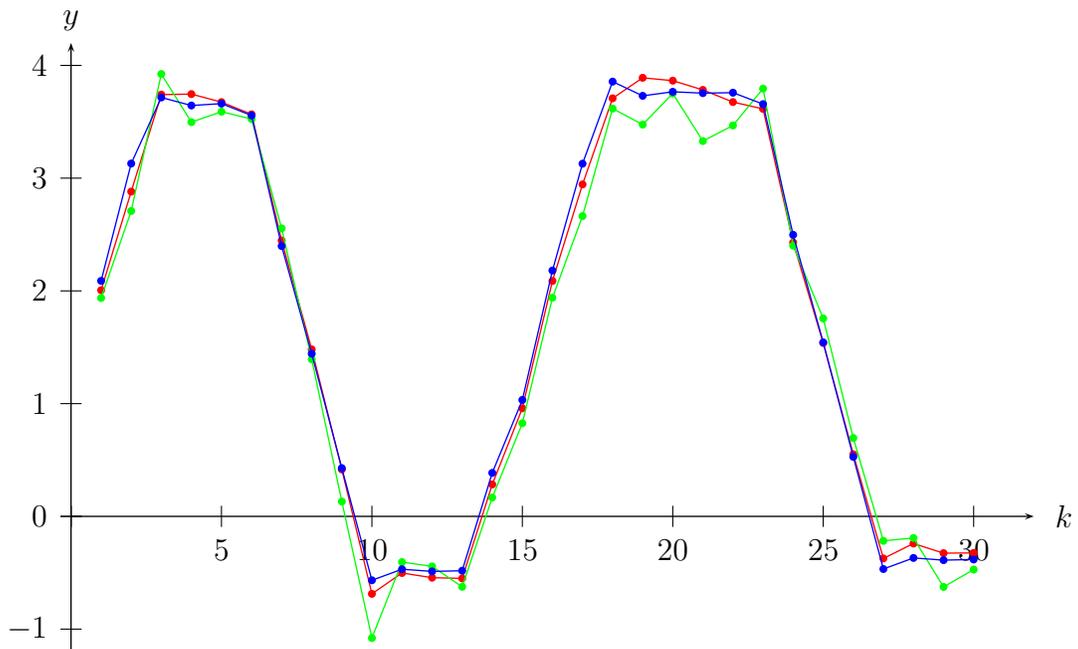


FIG. 8.6: Estimation de l'ordonnée pour Nav13, $N = 500$ et un bruit de $0.3I$. L'estimation est représentée en rouge, l'observation en vert et l'état réel de l'objet en bleu.

1.2 Le thermostat

Les figures 8.7 et 8.8 présentent respectivement l'estimation de la température de la pièce et de la température extérieure en fonction du temps pour l'instance Thermo01.

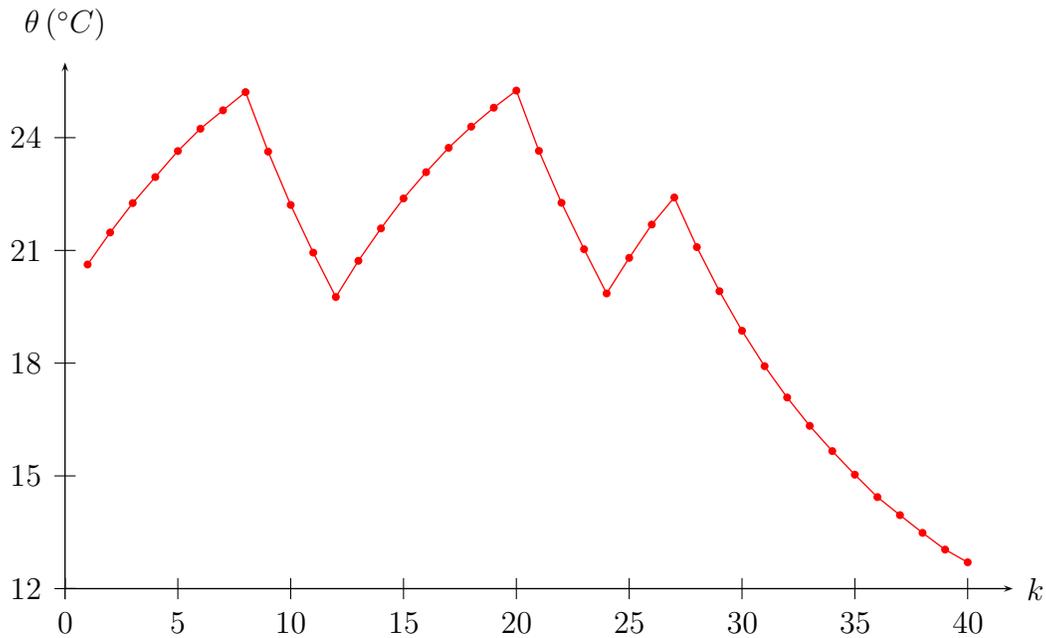


FIG. 8.7: Estimation de la température de la pièce pour l'instance Thermo01, $N = 100$ et un bruit de $0.1I$.

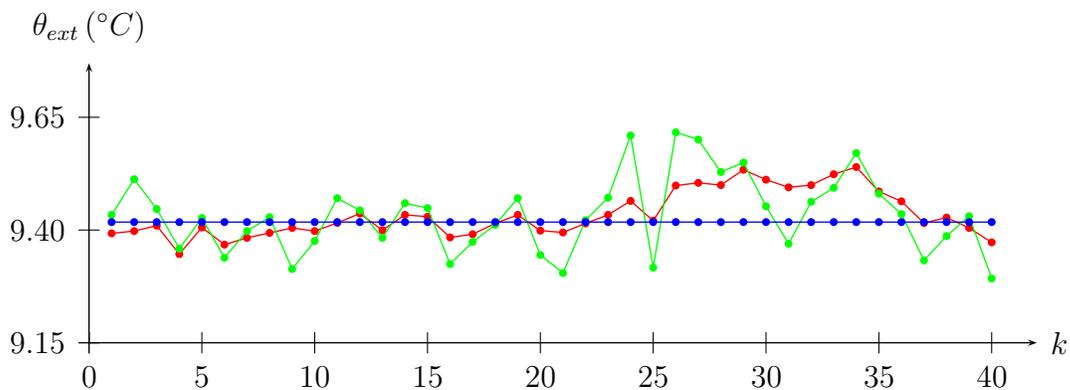


FIG. 8.8: Estimation de la température extérieure pour l'instance Thermo01, $N = 100$ et un bruit de $0.1I$. L'estimation est représentée en rouge, l'observation en vert et la température réelle en bleu.

2 Évolution de l'erreur quadratique

Les figures 8.9 et 8.10 représentent, respectivement pour Nav01 et Nav13, l'évolution de l'erreur au cours du temps. Cette erreur correspond à l'erreur en position entre l'estimation et l'état réel de l'objet (ou entre l'observation et l'état réel de l'objet pour la seconde courbe).

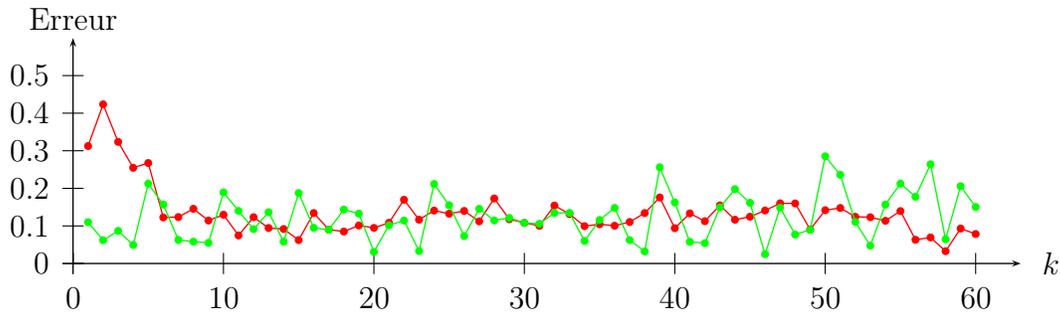


FIG. 8.9: Erreur d'estimation (en rouge) et d'observation (en vert) pour Nav01.

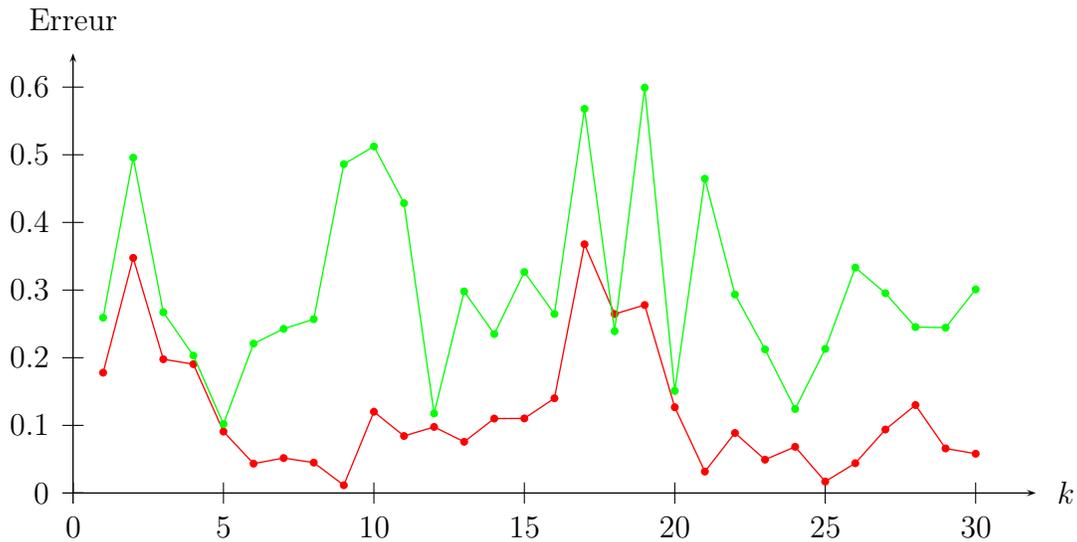


FIG. 8.10: Erreur d'estimation (en rouge) et d'observation (en vert) pour Nav13.

Dans les deux cas, on peut constater que l'erreur d'estimation reste, la plupart du temps, inférieure à l'erreur d'observation. L'estimateur joue bien son rôle : il permet d'absorber les écarts dus au bruit d'observation en donnant une information plus précise sur l'état réel du système.

Pour le cas du thermostat, la figure 8.11 représente les erreurs d'estimation sur la température de la pièce et sur la température extérieure.

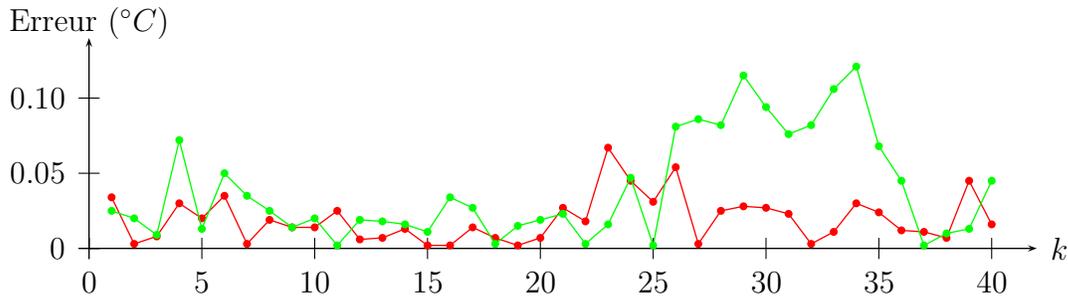


FIG. 8.11: Erreur d'estimation sur la température de la pièce (en rouge) et extérieure (en vert) pour l'instance Thermo01.

2.1 Sensibilité au nombre de particules

Pour diminuer l'erreur d'estimation et approcher au mieux l'état du système, nous augmentons le nombre de particules. Ainsi, les chances de disposer des bonnes particules sont plus grandes. Les figures 8.12, 8.13 et 8.14 présentent l'évolution de l'erreur moyenne (sur chaque simulation) en fonction du nombre de particules utilisées, pour Nav01, Nav13 et Thermo01 respectivement (avec les mêmes paramétrages que précédemment).

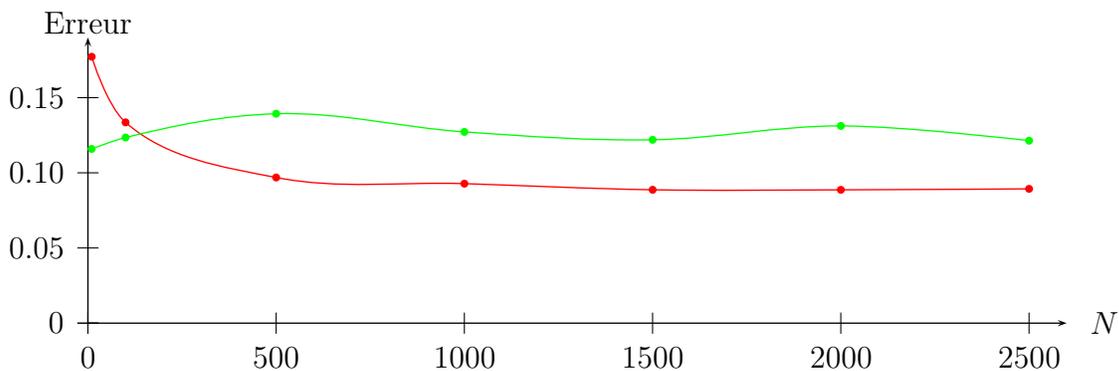


FIG. 8.12: Erreur d'estimation (en rouge) et d'observation (en vert) en fonction du nombre de particules (N) pour l'instance Nav01.

On peut essentiellement remarquer qu'augmenter le nombre de particules ne diminue pas clairement l'erreur, et donc n'améliore pas sensiblement l'efficacité de l'estimateur.

Dans les trois cas, un nombre de particules approchant le millier semble tout à fait approprié pour estimer l'état du système. Utiliser davantage de particules améliore très peu (voire détériore) la qualité de l'estimation, mais augmente les temps de calcul.

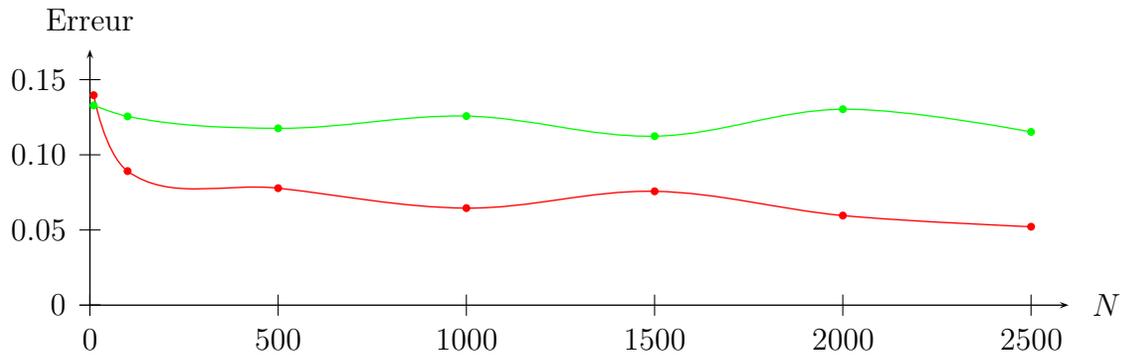


FIG. 8.13: Erreur moyenne entre estimation et état réel (en rouge) et entre observation et état réel (en vert) en fonction du nombre de particules (N) pour l'instance Nav13.

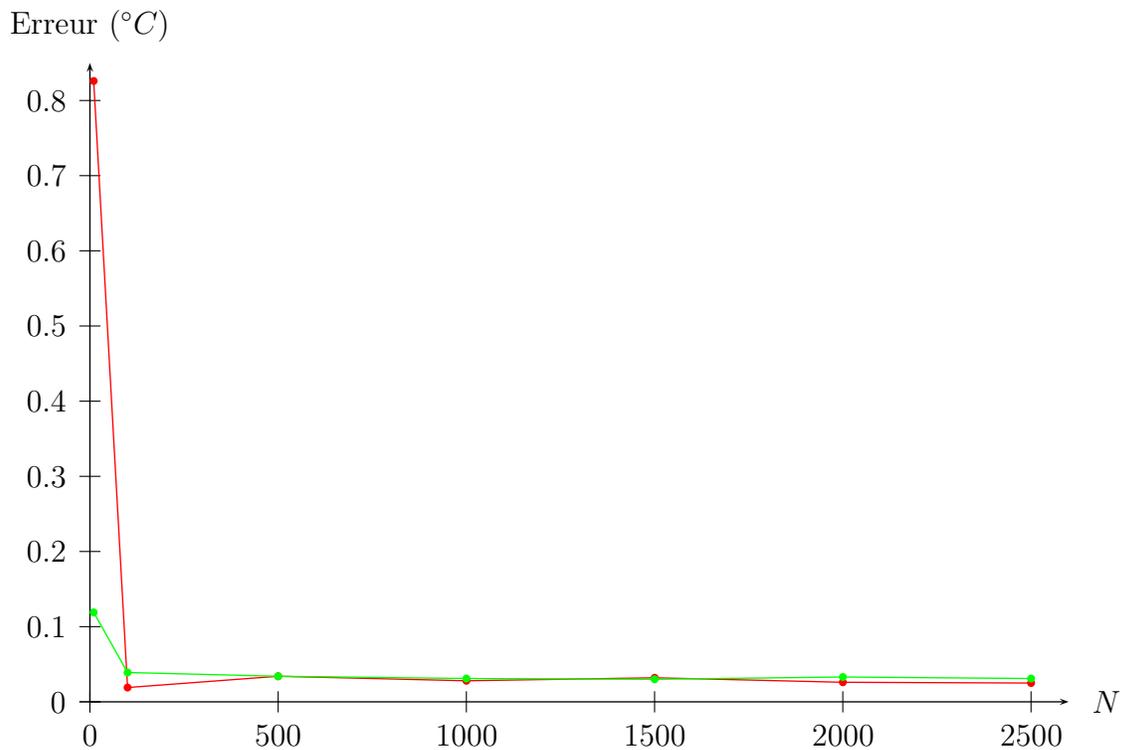


FIG. 8.14: Erreur moyenne de l'estimation de la température de la pièce (en rouge) et de la température extérieure (en vert) en fonction du nombre de particules (N) pour l'instance Thermo01.

2.2 Sensibilité au bruit

Lorsque l'on augmente le bruit d'observation, on peut constater, sur les figures 8.15 (Nav01) et 8.16 (Nav13), que l'erreur augmente de façon proportionnelle.

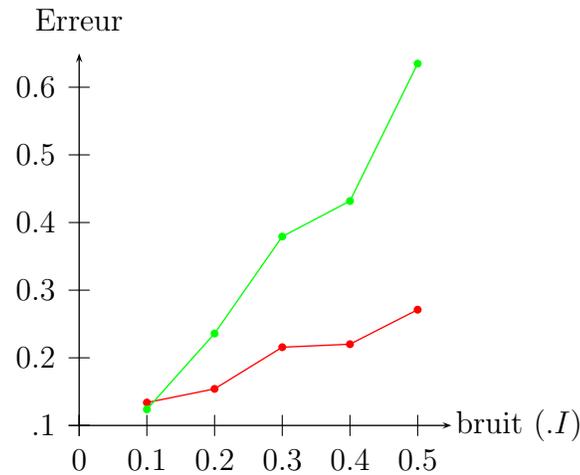


FIG. 8.15: Erreur moyenne entre estimation et état réel (en rouge) et entre observation et état réel (en vert) en fonction du bruit d'observation pour l'instance Nav01.

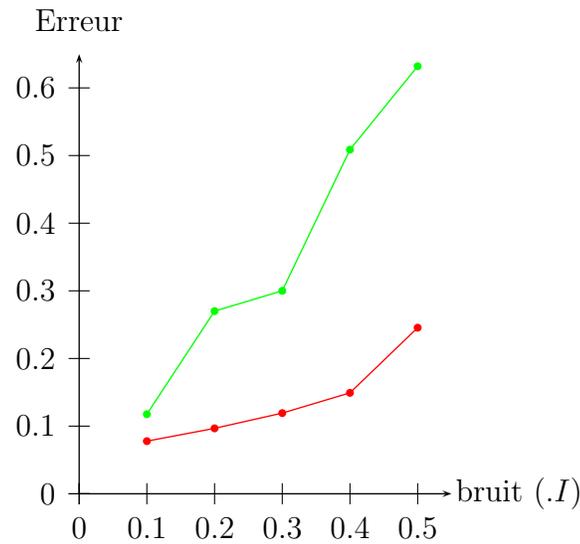


FIG. 8.16: Erreur moyenne entre estimation et état réel (en rouge) et entre observation et état réel (en vert) en fonction du bruit d'observation pour l'instance Nav13.

Cette constatation est logique, puisque l'incertitude liée à l'observation de l'état du système grandit, il devient donc plus difficile de déterminer cet état.

On ne peut cependant pas conclure sur le fait d'utiliser un bruit le plus petit possible, même si l'erreur s'en trouve diminuée. En effet, le bruit d'observation n'est pas "paramétrable", mais correspond à une information nécessaire pour assurer un suivi correct du système.

2.3 Sensibilité à l'initialisation

La particularité des instances utilisées dans ce chapitre est de proposer, en plus du modèle du système, une information sur l'état initial du système.

Cet état initial, même s'il n'est pas exact et comporte une incertitude (représentée par des intervalles de valeurs), donne suffisamment d'information pour initialiser le réseau de Petri particulière avant de lancer le processus d'estimation.

Cette initialisation se fait donc en tirant de façon uniforme les particules dans le parallépipède décrit par les intervalles.

Dans d'autres cas, il sera parfois difficile de connaître *a priori* l'état initial du système. Plusieurs solutions s'offrent alors :

- attendre la première observation, et générer ensuite les particules à partir de cette observation et du bruit d'observation ; cette méthode repose sur l'hypothèse que le bruit d'observation est suffisamment bien connu pour assurer que l'état réel du système sera représenté par les particules générées ;
- générer des particules sur un domaine très vaste, de façon à couvrir l'état initial ; cette méthode risque de noyer l'état initial sous le nombre de particules, et donc rendre difficile le suivi.

Les simulations suivantes visent à évaluer cette dernière méthode : quelle influence a le fait de générer un nombre de particules important, dans un sur-ensemble de l'état initial, sur la qualité de l'estimation ?

La figure 8.17 représente l'évolution de l'erreur en position de l'objet, pour l'instance Nav13 et différents marquages initiaux.

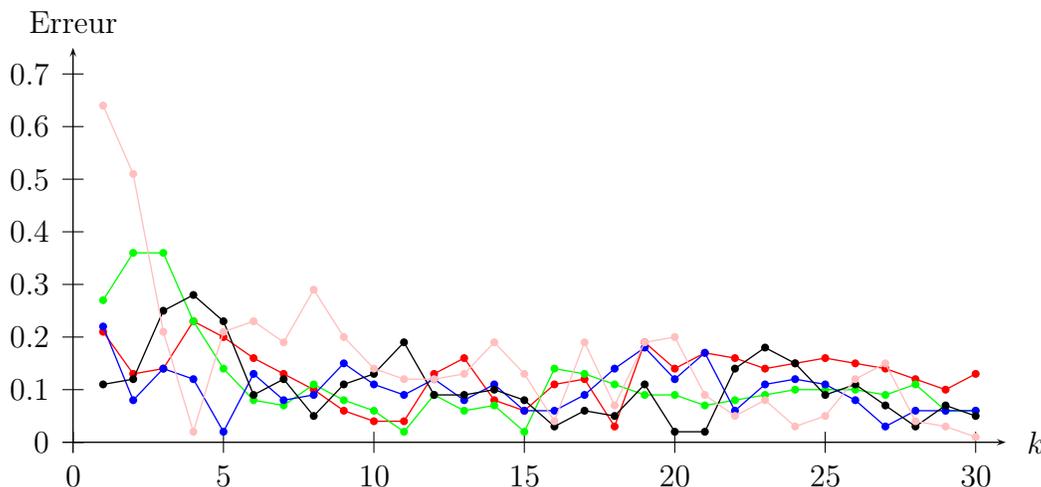


FIG. 8.17: Évolution de l'erreur d'estimation pour Nav13 en fonction du marquage initial.

Les marquages initiaux sont :

- pour la courbe rouge, le marquage généré par l'instance $((x_0, y_0) \in [4.5, 5.5] \times [1.5, 2.5])$,
- pour la courbe verte, le marquage contenant 500 fois la même valeur : $x_0 = 5$ et $y_0 = 2$ qui est le point central du cas précédent,
- pour la courbe bleue, $(x_0, y_0) \in [4, 6] \times [1, 3]$, qui contient l'état initial,
- pour la courbe noire, les points sont générés sur toute la grille $((x_0, y_0) \in [0, 7] \times [0, 6])$,
- pour la courbe rose, les points sont générés sur un intervalle adjacent à celui de l'instance : $(x_0, y_0) \in [3.5, 4.5] \times [0.5, 1.5]$.

On peut constater qu'au bout d'une dizaine d'observations, toutes les erreurs ont une évolution comparable : le marquage initial n'a donc plus aucune influence sur cette erreur, et donc sur la qualité de l'estimation.

Les deux seules courbes à avoir une erreur un peu plus importante au début de l'estimation sont la rose et la verte :

- la verte correspond à l'utilisation d'un unique point comme marquage initial ; l'état réel ne correspondant pas à ce point, mais n'en étant pas très loin, il faut un petit laps de temps pour recaler sur une valeur plus proche de la réalité ;
- la rose correspond à la case adjacente, et ne contient donc pas l'état réel initial ; quelques particules ont cependant une évolution qui n'est pas trop éloignée de l'état observé du système, ce qui permet de recaler les meilleures particules et donc d'approcher l'état réel du système.

Dans ce dernier cas, le suivi de la trajectoire de l'objet devient correct car le marquage initial contient des particules qui sont proches (relativement au bruit d'observation) de l'état initial du système.

Prenons la figure 8.18, dans laquelle sont représentées les courbes précédentes, ainsi qu'une courbe supplémentaire (bordeaux), obtenue avec un marquage initial généré dans $[0, 1] \times [0, 1]$, c'est-à-dire sur une case plus éloignée de l'état réel du système. L'estimation met une vingtaine de secondes à suivre l'état du système : l'évolution de l'objet est au début complètement fautive. Aucune particule ne correspondant à l'observation, le rééchantillonnage n'a aucun effet et le marquage initial suit son évolution indépendamment des observations. Puis, les trajectoires de l'estimation et de l'état réel se rapprochent (entre la 10^e et la 15^e observation) jusqu'à devenir suffisamment proche pour que l'observation attire des particules vers l'état réel du système.

Finalement, on peut conclure sur le fait que le marquage initial n'a aucune influence (sauf dans les premières secondes) sur la qualité de l'estimation, pour autant qu'il contienne quelques particules proches de l'état réel du système.

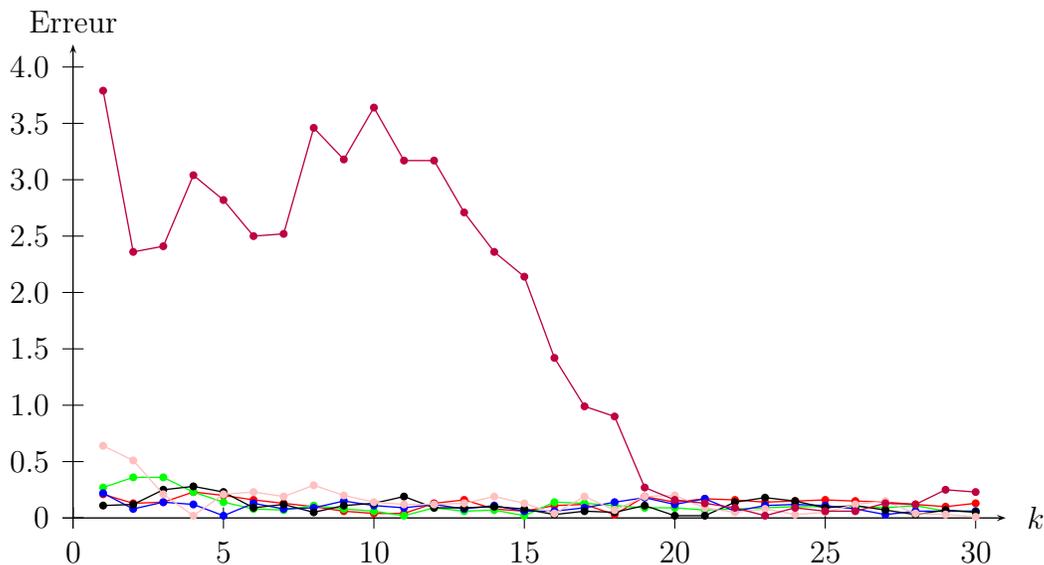


FIG. 8.18: Évolution de l'erreur d'estimation pour Nav13 en fonction du marquage initial.

3 Estimation de mode numérique

Avant d'analyser le comportement de la détection d'incohérences, nous étudions dans ce paragraphe l'estimation de mode numérique : à partir des particules recalées, la probabilité des modes numériques du système est calculée.

3.1 Problème de navigation

Dans le cas de la navigation, nous identifions la direction suivie par l'objet. Le mode est donc une valeur comprise entre 0 et 7, plus les cases "spéciales" A (associée au mode 8) et B (mode 9). La figure 8.19 représente l'évolution du mode numérique le plus probable en fonction du temps.

Dans cette simulation, sur 30 pas de temps, 2 estimations de mode numérique sont fausses, soit un taux d'erreur de 6.7%. Le tableau 8.1 expose les résultats de simulations permettant d'évaluer l'impact du nombre de particules et du bruit sur le taux d'erreur.

D'une manière générale, le taux d'erreur d'estimation de mode reste acceptable : à partir d'une centaine de particules, il passe rarement au dessus des 10%, et quand c'est le cas, le bruit est assez important.

On peut également remarquer que l'erreur d'estimation de mode numérique suit la même évolution que l'erreur d'estimation de la position de l'objet (figures 8.13 et 8.16), ce qui semble logique.

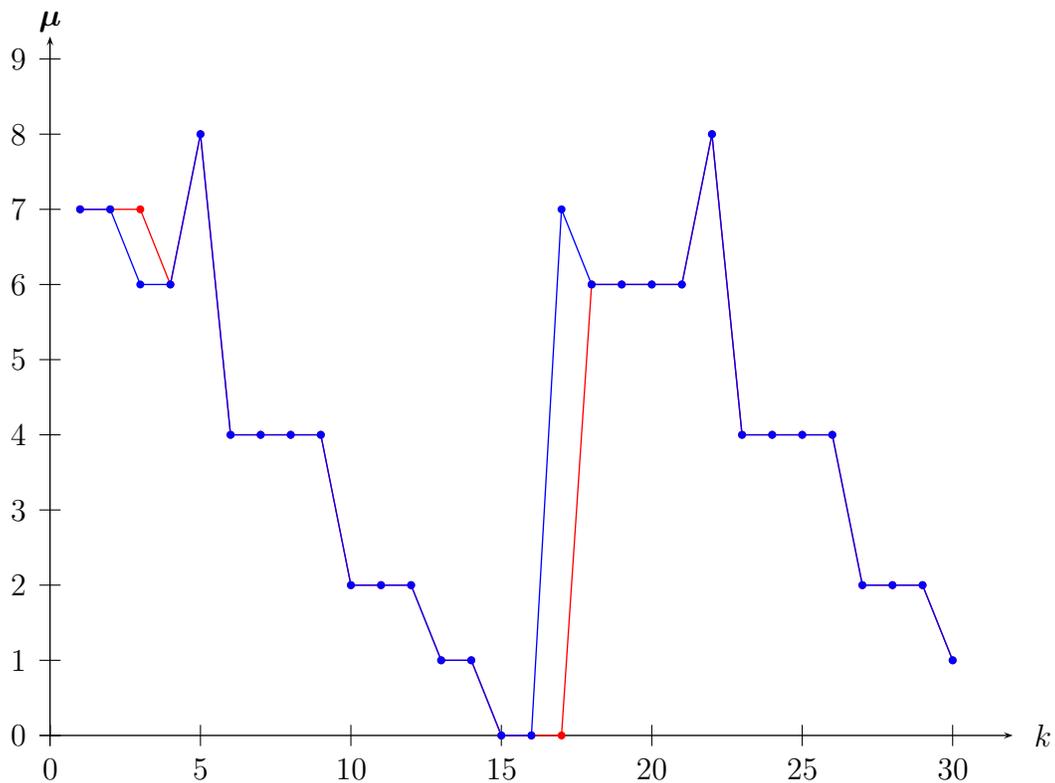


FIG. 8.19: Estimation du mode numérique pour l'instance Nav13 : le mode réel du système (en bleu) et le mode le plus probable (en rouge).

bruit \ N	10	100	500	1000	1500	2000	2500	Moyenne
0.1	0%	0%	0%	0%	0%	0%	0%	0%
0.2	3.3%	3.3%	0%	0%	3.3%	0%	0%	1.4%
0.3	6.7%	0%	6.7%	0%	0%	3.3%	3.3%	2.9%
0.4	16.7%	0%	3.3%	0%	13.3%	0%	6.7%	5.7%
0.5	10%	13.3%	13.3%	6.7%	0%	6.7%	0%	7.1%
Moyenne	7.3%	3.3%	4.7%	1.3%	3.3%	2%	2%	3.4%

TAB. 8.1: Pourcentage d'erreurs sur le mode le plus probable, en fonction de N et du bruit, pour l'instance Nav13.

En fonction de N , l'erreur décroît sensiblement entre 10 et 1000 particules, puis reste à peu près stable : quelques centaines de particules suffisent donc à assurer une bonne estimation du système.

En fonction du bruit, l'erreur augmente. L'augmentation du bruit représente une aug-

mentation de l'incertitude sur la position du système, il est donc logique que le taux d'erreur d'estimation de mode augmente de façon proportionnelle.

3.2 Le thermostat

Le modèle du thermostat comporte trois modes numériques : le mode de chauffe (p_0), le mode de veille (p_1) et le mode d'arrêt (p_2). La figure 8.20 représente l'évolution du mode numérique le plus probable au cours du temps.

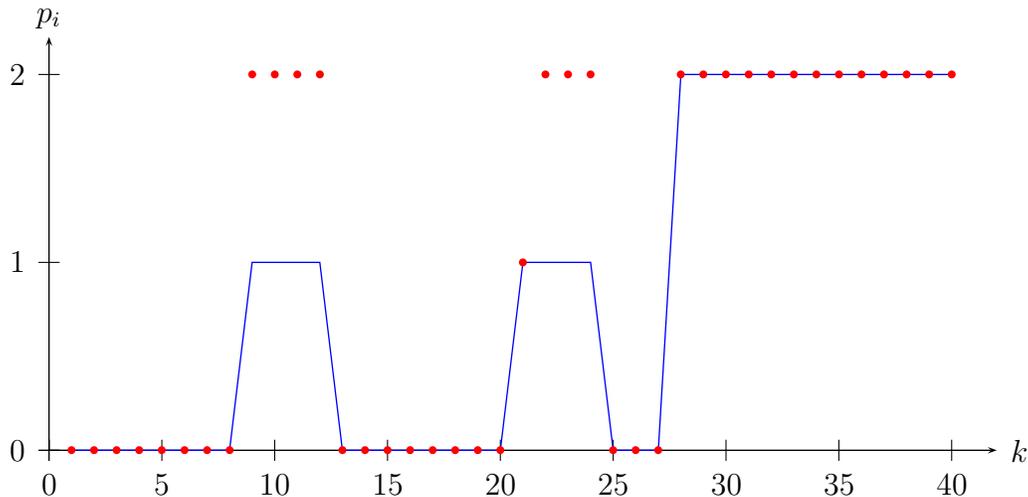


FIG. 8.20: Estimation du mode numérique le plus probable (en rouge) et mode réel (en bleu) pour Thermo01.

On peut voir que les modes p_0 (chauffe) et p_2 (arrêt) sont correctement estimés. En revanche, l'estimation est erronée 3 fois sur 4 pour le mode p_1 (veille).

Ce comportement de l'estimateur est explicable. Le modèle du thermostat a une particularité qui rend difficile l'estimation de mode : les places p_1 et p_2 ont la même équation. En effet, dans les deux modes, la température suit la même évolution. Ainsi, les particules présentes dans p_1 et p_2 ont des valeurs très proches.

Sur la figure 8.21(a), le marquage prédit $\widehat{\mathcal{M}}(k+1|k)$ contient 200 particules. 100 d'entre elles évoluent dans la place p_1 et représentent le mode de veille. Les 100 autres correspondent au mode d'arrêt, et marquent la place p_2 .

Les particules présentes dans p_2 sont issues des places p_0 et p_1 , et du pseudo-tir d'une transition simulant l'arrivée de l'événement σ_{off} . Par conséquent, les valeurs de ces particules sont sensiblement identiques (au bruit de prédiction près) aux autres particules du

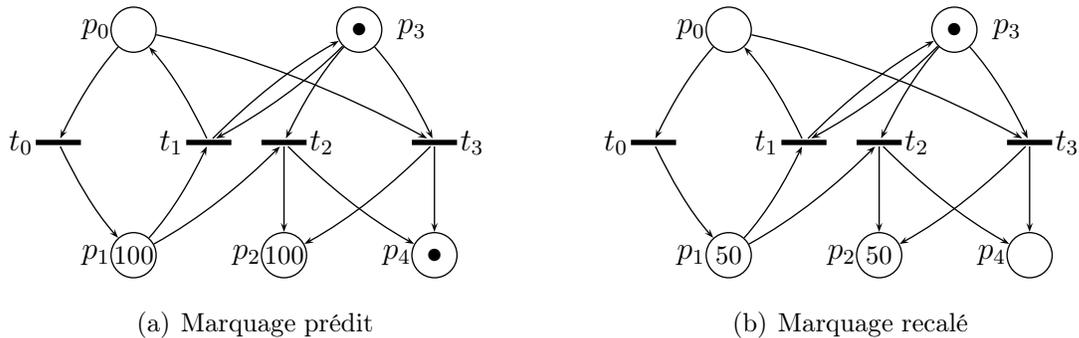


FIG. 8.21: Explication de l'erreur d'estimation de mode numérique.

réseau.

Lors du recalage, 100 particules doivent être rééchantillonnées parmi ces 200 particules. Logiquement, le marquage recalé correspondra (ou presque) à celui de la figure 8.21(b), avec une cinquantaine de particules dans chacune des places p_1 et p_2 .

L'étape de prédiction suivante va de nouveau simuler l'arrivée de l'événement σ_{off} et pseudo-tirer la transition t_2 . Le marquage résultant sera celui de la figure 8.22(a) : les 50 particules copiées de p_1 viennent s'ajouter aux 50 particules recalées dans p_2 .

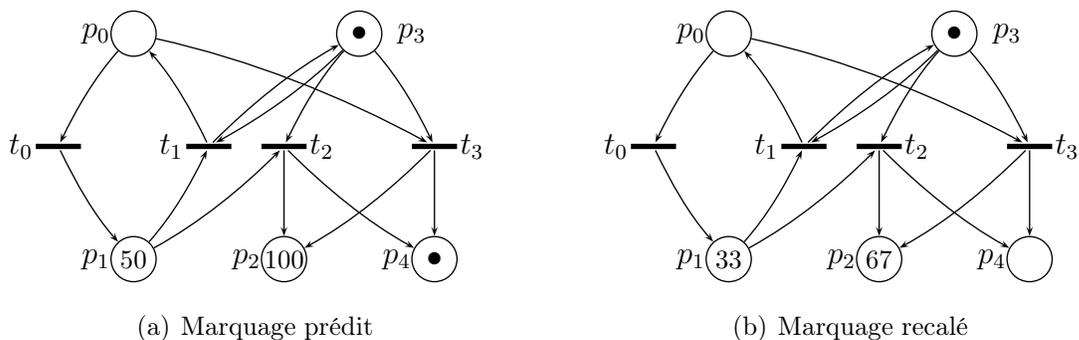


FIG. 8.22: Explication de l'erreur d'estimation de mode numérique.

Et comme précédemment, lors du recalage, les particules ont des valeurs très proches puisqu'elles suivent la même équation. On va donc sélectionner environ un tiers des par-

ticules dans p_1 , et le reste dans p_2 (figure 8.22(b)). Et ainsi de suite, au fil des recalages, les particules sont attirées dans la place p_2 .

De ce fait, le mode numérique le plus probable est p_2 alors que les valeurs des particules sont les mêmes dans p_1 et p_2 . Nous avons constaté expérimentalement ce comportement sur les différentes instances du thermostat et différentes configurations.

La figure 8.23 représente l'évolution de la probabilité de " p_0 ou p_1 " (qui vaut la somme des probabilités), c'est-à-dire la probabilité de ne pas se retrouver dans la place "absorbante" p_2 ($E_k(p_2) = 1 - (E_k(p_0) + E_k(p_1))$).

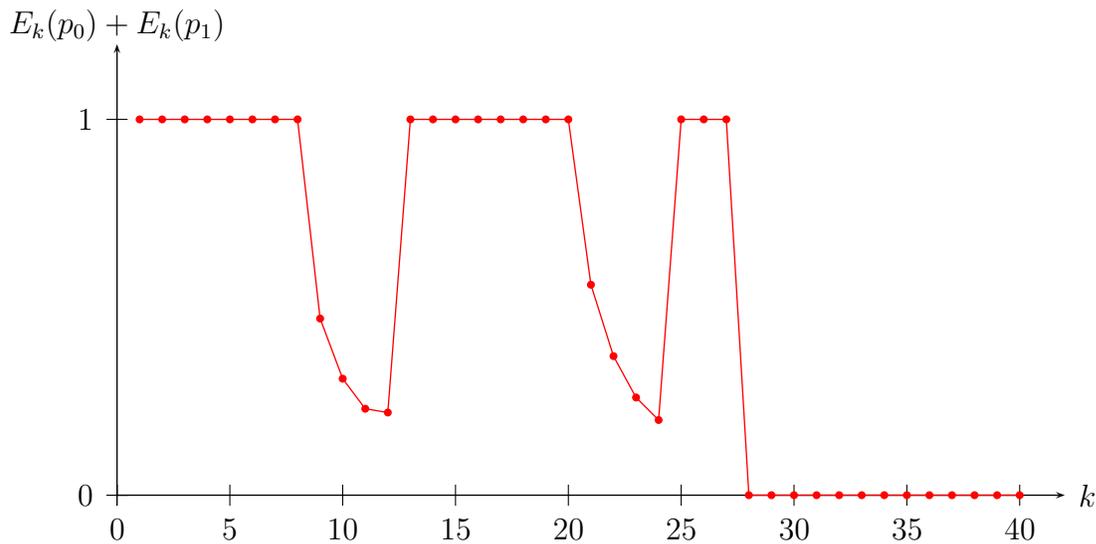


FIG. 8.23: Probabilité du mode numérique.

Nous pouvons observer que cette probabilité diminue lorsque le mode numérique du système est p_1 (figure 8.20), et même que cette évolution suit une courbe en $\frac{1}{k}$, ce qui correspond à ce que nous avons décrit ($\frac{1}{2}$ des particules au premier recalage, $\frac{1}{3}$ au second. . .)

4 Détection d'incohérences

Le problème de navigation, qui a été utilisé pour une bonne partie des simulations réalisées dans ce chapitre, n'apporte aucune information pour la détection d'incohérences : son réseau de Petri ne contient aucune place symbolique.

Nous travaillons donc sur le modèle du thermostat. Comme nous l'avons vu au chapitre 7, la détection d'incohérences repose d'abord sur la construction du graphe des marquages cohérents. Celui du réseau de Petri particulière du thermostat est représenté sur la figure 8.24.

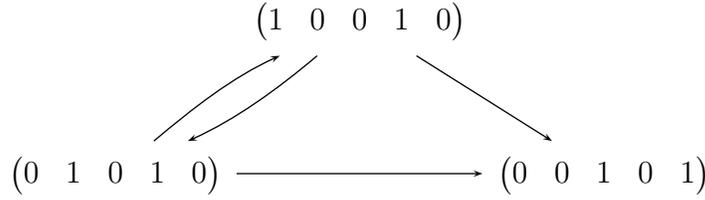


FIG. 8.24: Graphe des marquages cohérents du thermostat.

4.1 Détection d'incohérences en mode “nominal”

Nous commençons l'étude de la détection d'incohérences par l'instance Thermo01.

4.1.1 Graphe de recalage par places

L'estimation du mode numérique du système a été présentée sur la figure 8.20.

Les configurations du système sont $\delta(p_3) = \{on\}$ et $\delta(p_4) = \{off\}$. Ainsi, nous observons que la relation de préférence se limite à $\{on\}$ (qui est donc la meilleure) de l'instant 1 à l'instant 27. À partir de l'instant 28, c'est $\{off\}$ qui est la seule configuration classée.

Par conséquent, les différents graphes de recalage par places au cours de l'estimation sont² :

- des instants 1 à 8, 13 à 20 et 25 à 27 :

$$(p_0, p_3) \tag{8.1}$$

- à l'instant 21 :

$$(p_1, p_3) \longrightarrow (p_2, p_3) \tag{8.2}$$

- des instants 9 à 12 et 22 à 24 :

$$(p_2, p_3) \longrightarrow (p_1, p_3) \tag{8.3}$$

- à partir de l'instant 28 :

$$(p_2, p_4) \tag{8.4}$$

À partir de ces graphes de recalage, nous détectons des incohérences lorsque le meilleur (ou le seul) couple est inaccessible. C'est le cas uniquement pour le couple (p_2, p_3) .

Ce mode correspond à un arrêt définitif du chauffage alors que le thermostat est toujours allumé. Cette incohérence est détectée aux instants 9, 10, 11, 12, 22, 23 et 24.

Elle s'explique, tout comme l'erreur sur l'estimation du mode numérique, par l'attraction de particules dans la place p_2 .

²La représentation d'un graphe par $a \longrightarrow b$ signifie que le nœud a est préféré au nœud b .

Cette fausse détection (puisque le comportement réel du thermostat est tout à fait correct) n'est pas due à l'estimation ou au principe d'incohérence, mais au modèle du thermostat qui crée une ambiguïté sur le mode numérique suivi par le système.

4.1.2 Graphe de recalage par poids

Les graphes de recalage par poids obtenus lors de l'estimation de Thermo01 sont :

– de l'instant 1 à 7, 14 à 19 et 26 à 27 :

$$(\gamma_0, p_3) \longrightarrow (\gamma_2, p_3)^3 \quad (8.5)$$

– aux instants 8, 13, 20 et 25 :

$$(\gamma_0, p_3) \longrightarrow (\gamma_{20}, p_3) \longrightarrow (\gamma_{10}, p_3) \longrightarrow (\gamma_{21}, p_3) \longrightarrow (\gamma_{11}, p_3) \quad (8.6)$$

– aux instants 9 et 21 :

$$(\gamma_{20}, p_3) \longrightarrow (\gamma_{10}, p_3) \longrightarrow (\gamma_{21}, p_3) \longrightarrow (\gamma_{11}, p_3) \longrightarrow (\gamma_0, p_3) \quad (8.7)$$

– aux instants 10, 11, 22 et 23 :

$$(\gamma_{20}, p_3) \longrightarrow (\gamma_{10}, p_3) \longrightarrow (\gamma_{21}, p_3) \longrightarrow (\gamma_{11}, p_3) \longrightarrow (\gamma_{22}, p_3) \quad (8.8)$$

$$(\gamma_{22}, p_3) \longrightarrow (\gamma_{12}, p_3) \longrightarrow (\gamma_{23}, p_3) \longrightarrow (\gamma_{13}, p_3) \quad (8.9)$$

– aux instants 12 et 24 :

$$(\gamma_{20}, p_3) \longrightarrow (\gamma_{10}, p_3) \longrightarrow (\gamma_{21}, p_3) \longrightarrow (\gamma_{11}, p_3) \quad (8.10)$$

– à l'instant 28 :

$$(\gamma_2, p_4) \longrightarrow (\gamma_0, p_3) \quad (8.11)$$

– de l'instant 29 à 40 :

$$(\gamma_2, p_4) \quad (8.12)$$

Les nœuds incohérents sont ceux correspondant au couple (p_2, p_3) , c'est-à-dire l'ensemble des nœuds (γ_{2*}, p_3) . Il y a donc incohérence aux instants 9, 10, 11, 12, 21, 22, 23 et 24.

4.1.3 Synthèse

La figure 8.25 résume le processus d'estimation et de détection d'incohérences sur l'instance Thermo01.

³Pour faciliter l'analyse du graphe de recalage par poids, nous noterons γ_{0*} (respectivement γ_{1*} et γ_{2*}) les classes d'équivalence dont la place est p_0 (respectivement p_1 et p_2).

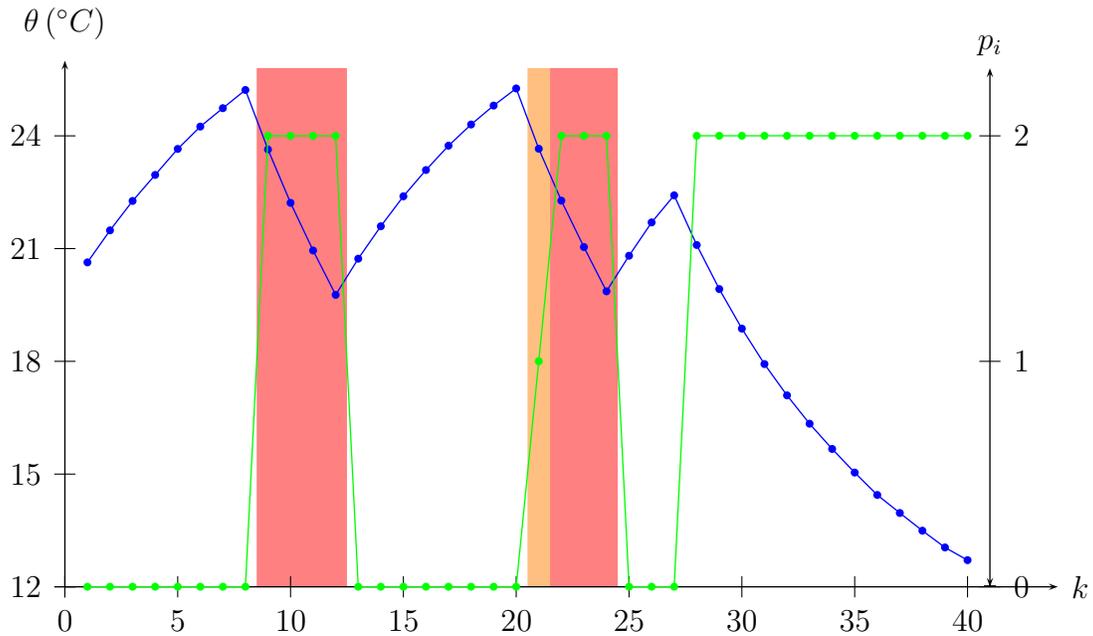


FIG. 8.25: Estimation de la température de la pièce (en bleu), estimation du mode numérique (en vert) et détection d'incohérences pour l'instance Thermo01.

La détection d'incohérences à partir du graphe de recalage par places et représentée par un fond rouge. La détection d'incohérences à partir du graphe de recalage par poids fournit les mêmes instants, plus l'instant 21, représenté par un fond orange.

Ces détections, même si elles ne correspondent pas dans la réalité à une défaillance du thermostat, mais à une défaillance du modèle, coïncident avec les erreurs d'estimation de mode numérique pour la zone rouge. La détection orange, quant à elle, ne correspond pas à une erreur d'estimation, mais on peut remarquer :

- qu'elle est due à une seule détection (par poids et non par places) ;
- qu'elle précède une zone rouge.

Pour approfondir notre étude de la détection d'incohérences sur le thermostat, nous utilisons des systèmes défaillants.

4.2 Comportements défaillants

Dans chacun des cas suivants, le modèle utilisé est toujours celui de l'instance Thermo01. En revanche, les observations sont issues d'un thermostat défaillant.

4.2.1 Thermo24

Le thermostat Thermo24 présente la défaillance suivante : au lieu d'arrêter de chauffer à $\theta_{max} = 25$, il arrête à $\theta_{max} = 24$. La figure 8.26 représente l'estimation de ce thermostat à partir du modèle Thermo01.

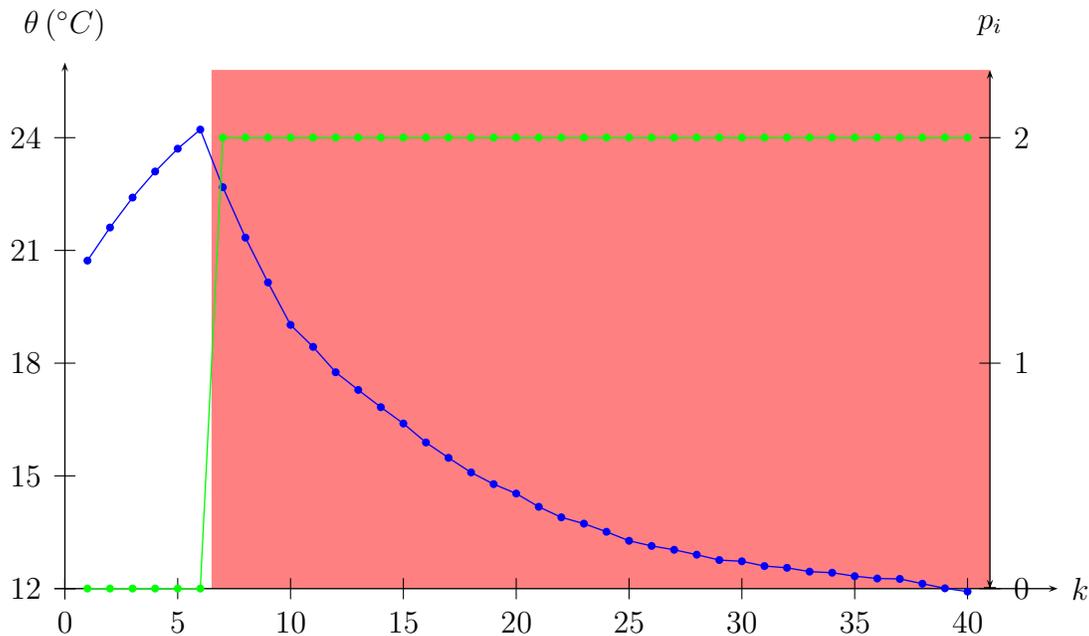


FIG. 8.26: Estimation de la température de la pièce (en bleu), estimation du mode numérique (en vert) et détection d'incohérences pour le thermostat Thermo24.

Dans ce cas, une incohérence est détectée à partir de l'instant 7, c'est-à-dire lors du premier arrêt du chauffage. Ceci s'explique par le fait que l'observation suit l'équation des places p_1 et p_2 . Etant donné qu'aucune particule ne marque la place p_1 puisque cela ne correspond pas au modèle, seule la place p_2 est estimée à partir de l'instant 7.

L'incohérence détectée correspond alors bien à une défaillance du système. Cependant, le suivi de l'état de ce système devient impossible puisque le comportement du modèle ne correspond plus au comportement du système.

4.2.2 Thermo26

Le thermostat Thermo26 présente la défaillance suivante : au lieu d'arrêter de chauffer à $\theta_{max} = 25$, il arrête à $\theta_{max} = 26$. La figure 8.27 représente l'estimation de ce thermostat à partir du modèle Thermo01.

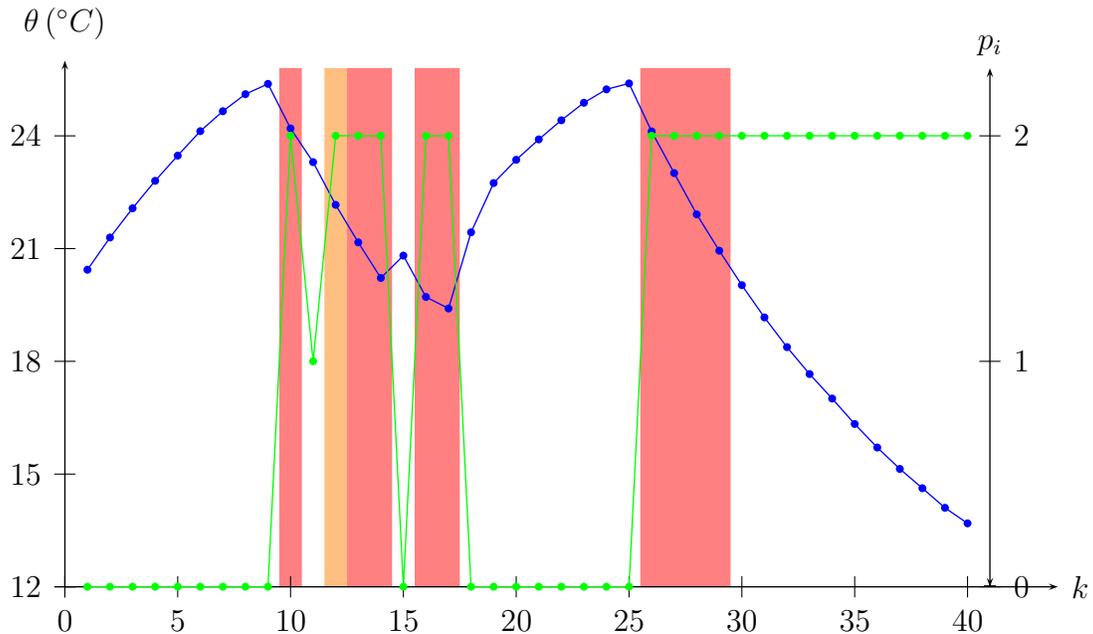


FIG. 8.27: Estimation de la température de la pièce (en bleu), estimation du mode numérique (en vert) et détection d'incohérences pour le thermostat Thermo26.

Les incohérences détectées correspondent, pour la plupart, aux incohérences détectées en mode nominal, et causées par le modèle.

Les incohérences aux instants 16 et 17 correspondent à deux observations bruitées qui font diverger l'estimation vers la place p_2 , alors que le mode réel est p_0 .

La défaillance de Thermo26 au niveau de la température maximale n'est pas détectée par le modèle. Cette défaillance est atténuée, et passe pour une observation bruitée : les particules sont recalées pour approcher au mieux l'état observé.

On remarque là aussi que la zone orange précède une zone rouge.

5 Conclusion

Dans ce chapitre, nous avons vu de nombreuses simulations qui ont permis d'illustrer les principes d'estimation et de diagnostic proposés dans ce manuscrit.

L'estimation de la position d'un objet, sur les instances Nav01 et Nav13, et de la température régulée par le thermostat Thermo01, nous ont permis de constater que le suivi de l'état numérique du système est satisfaisant.

Nous avons également constaté sur ces exemples :

- que l’erreur diminue au cours du temps, et donc que l’estimation converge, à une précision dépendant des bruits de modèle et d’observation ;
- que cette erreur diminue avec le nombre de particules, mais qu’un nombre de particules approchant le millier est amplement suffisant pour assurer un bon suivi.

Nous avons également étudié l’estimation du mode numérique, et constaté là aussi qu’un millier de particules suffisait à limiter le taux d’erreur de cette estimation. De plus, ce taux d’erreur était, pour l’instance Nav13, inférieur à 10%, ce qui est tout à fait satisfaisant.

Finalement, nous avons illustré, sur l’instance Thermo01 ainsi que deux thermostats défaillants (Thermo24 et Thermo26), la détection d’incohérences. Les incohérences détectées dans ces différents exemples ont permis de mettre en avant :

- une mauvaise modélisation, due à l’association d’une même équation à deux places distinctes ;
- la détection de la défaillance Thermo24 ;
- l’adaptation de l’estimation à la défaillance Thermo26, qui n’est pas détectée.

De plus, nous avons pu constater deux niveaux d’incohérence, selon qu’uniquement la détection d’incohérences par poids (niveau orange) ou que les deux détections (niveau rouge) avaient détecté cette incohérence. De plus, sur les différentes incohérences détectées, le niveau orange a précédé le niveau rouge.

Cette constatation nous incite à approfondir l’étude et l’expérimentation de ces niveaux d’incohérence pour pouvoir non seulement détecter des comportements incohérents, mais également les prédire.

De plus, la structure du graphe de recalage par poids, construit pour fournir davantage d’information sur le mode estimé, peut également permettre d’affiner ce diagnostic, et de pouvoir prédire, détecter et éventuellement expliquer des incohérences dans le comportement du système.

Chapitre 9

Suivi de l'activité de pilotage

Le travail de pilotage a sensiblement évolué durant les dernières décennies : d'un travail de guidage/pilotage, il est aujourd'hui devenu davantage un travail de supervision du pilote automatique. Dans ce contexte, nous proposons d'appliquer le modèle des réseaux de Petri particuliers et le principe d'estimation et de diagnostic pour mettre en place un système de suivi conjoint du comportement de l'avion et des actions de l'équipage. Ce système a pour vocation d'assister l'équipage dans sa tâche de supervision.

1 L'activité de pilotage

On peut considérer que les avions de ligne modernes sont commandés par deux agents : un agent humain, l'équipage, et un agent artificiel, composé du pilote automatique et du FMS¹. L'avion en vol est également soumis à des interactions avec d'autres agents, les contrôleurs aériens, qui peuvent demander des modifications concernant le plan de vol ou la trajectoire de l'avion : mise en attente, autorisations d'approche ou d'atterrissage. . .

L'essor des systèmes automatiques a été guidé par l'objectif d'amélioration de la sécurité et la réduction de la charge de travail de l'équipage. Ces automatismes sont des mécanismes complexes, capables de suivre un plan de vol saisi par l'équipage du décollage à l'atterrissage, sans intervention humaine. Cette gestion du vol se fait sur le plan latéral (navigation) et sur le plan vertical (niveaux de vol), de façon à optimiser les performances (temps de parcours, consommation. . .)

L'équipage a cependant la possibilité de régler, à n'importe quel moment, un certain

¹Flight Management System, système de gestion du vol qui fournit, à partir du plan de vol saisi par le pilote, les consignes correspondant au profil de vol à suivre.

nombre de paramètres (cap, vitesse. . .) pour corriger la trajectoire de l'avion ou suivre une indication du contrôle aérien. Il peut également, dans certains cas particuliers, reprendre le contrôle manuel de l'avion. Dans certaines conditions (protection du domaine de vol), le pilote automatique peut également opérer une *réversion de mode*, qui modifie sensiblement le comportement de l'avion, et a pour but d'éviter des situations dangereuses (survitesse, décrochage. . .)

La supervision de tels systèmes peut rapidement devenir complexe, et peut entraîner des conflits entre les différents agents, en particulier en situation dégradée.

2 Conflits entre l'équipage et le pilote automatique

Les cas de conflits qui nous intéressent sont les conflits entre l'équipage et le pilote automatique, puisque ce sont les deux agents principaux du pilotage, et que c'est au pilote automatique que l'équipage donne ses consignes.

Avant de présenter le scénario utilisé pour les expérimentations, nous détaillons quelques cas réels de conflits.

2.1 Déconnexion automatique non perçue

Dans ce cas, le pilote automatique rend le contrôle de l'avion à l'équipage, alors que l'équipage pense que le pilote automatique est toujours connecté.

Chaque déconnexion du pilote automatique est signalée par une alarme sonore et une alarme visuelle sur l'écran d'alarmes.

Lorsque le pilote automatique se déconnecte après une survitesse par exemple, l'alarme sonore de survitesse est prioritaire sur l'alarme de déconnexion. L'équipage, qui suit la procédure de survitesse, peut alors ne pas remarquer que le pilote automatique s'est déconnecté.

Ce cas de conflit est présenté dans un rapport du BEA² sur le *débrayage du pilote automatique*³.

2.2 Consignes incohérentes

Une consigne de l'équipage peut être appropriée dans certaines situations, mais complètement incohérente à d'autres moments. Dans le conflit présenté dans le rapport sur l'*écart de trajectoire*⁴ du BEA, le pilote automatique est en mode d'approche (APPR),

²Bureau Enquête et Analyses

³<http://www.bea-fr.org/itp/events/debrayagpa/debrayagpa.html>

⁴<http://www.bea-fr.org/itp/events/ecartdetrajectoire/ecartdetrajectoire.html>

à 1200 pieds (environ 400 m) au dessus du sol. Le copilote sélectionne alors un nouveau point de passage dans le MCDU⁵.

Cette sélection entraîne alors une réversion de mode du pilote automatique, du mode APPR aux modes NAV⁶ et V/S⁷. La vitesse verticale est donc immédiatement fixée à la dernière vitesse verticale de l'avion, dans ce cas là $-1000\text{ft}/\text{min}$. L'équipage n'a pas remarqué ce changement de mode, et l'avion est descendu à moins de 400 pieds (environ 130 m) du sol.

2.3 Buts différents de l'équipage et du pilote automatique

La sélection d'un mode de vol par l'équipage peut avoir des répercussions inattendues sur le pilote automatique. C'est le cas par exemple lorsque l'équipage entre une altitude pour la remise des gaz sur le panneau de contrôle du pilote automatique, alors que ce dernier est dans le mode Glide/Scope, c'est-à-dire qu'il est en train de suivre la pente de descente vers la piste d'atterrissage.

Si, lors de cette descente, la vitesse de l'avion devient trop importante (ce qui peut être le cas lors de la sortie des volets), le pilote automatique abandonne la consigne Glide/Scope pour essayer d'atteindre l'altitude saisie. À cet instant, l'équipage veut atterrir, mais le pilote automatique veut monter (Crow *et al.*, 2000).

2.4 Mise en palier

Un avion est en montée vers une altitude z . Cette montée est contrôlée par le FMS. Au cours de la montée, l'équipage met l'avion en palier, en demandant une vitesse verticale nulle sur le panneau de commande du pilote automatique. Cette mise en palier peut être due, par exemple, à un ordre du contrôleur aérien pour éviter une collision avec un avion situé à un niveau de vol supérieur.

Dans certaines situations, le pilote automatique, qui gère toujours le comportement de l'avion aux limites du domaine de vol, peut opérer une réversion de mode. Si la vitesse de l'avion approche la vitesse maximale autorisée (la mise en palier augmente l'accélération), le pilote automatique essaie de "casser" cette accélération en faisant cabrer l'avion : l'avion se remet donc à monter alors que la consigne du pilote est de rester en palier (figure 9.1).

Ce scénario est étudié au paragraphe suivant.

⁵Multifunction Control Display Unit, qui permet de saisir le plan de vol et contrôler ou modifier le profil du vol

⁶Navigation

⁷Vertical/Speed

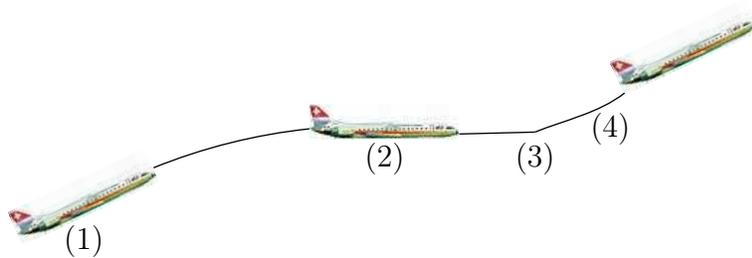


FIG. 9.1: Réversion de mode suite à une mise en palier : (1) pilote automatique en mode *Climb*, (2) mise en palier commandée par l'équipage, (3) survitesse, (4) réversion en mode *Climb*.

3 Supervision du système hybride avion–équipage

L'activité de pilotage est donc une activité de supervision du comportement de l'avion et des automatismes qui le commandent. Ce comportement est hybride : la trajectoire de l'avion est continue, et suit les règles de la dynamique du vol ; cette dynamique est conditionnée par le mode de fonctionnement (discret) de l'avion qui dépend de sa configuration (état des ailerons, des volets, de la manette des gaz. . .) et qui est commandée par le pilote automatique ou l'équipage.

Nous proposons donc d'utiliser l'approche présentée dans ce manuscrit :

- en modélisant le comportement hybride de l'avion par un réseau de Petri particulière (comportement continu de l'avion physique et actions discrètes de l'équipage),
- en utilisant le principe d'estimation pour suivre l'état du système hybride,
- en détectant des incohérences au sein de cette estimation qui représentent un potentiel conflit entre le comportement de l'avion et les commandes de l'équipage.

3.1 Modélisation du scénario de mise en palier

Le réseau de Petri particulière de la figure 9.2 représente le comportement nominal de l'avion.

Le comportement initial de l'avion est modélisé par une accélération verticale (place p_0). Puis, lorsque l'accélération est proche de 0 (transition t_0), l'avion suit une montée à vitesse verticale constante (place p_1).

Lorsque l'équipage envoie la consigne *VS* (transition t_1), le pilote automatique passe en mode *VS0* (place p_5) et l'avion décélère (place p_2) puis se met en palier (place p_3) : son altitude est constante, sa vitesse verticale est donc nulle (transition t_2).

Si l'équipage envoie la consigne *ALT* (transition t_3), le pilote automatique passe en mode *OP CLB* (place p_4) et l'avion se remet à cabrer.

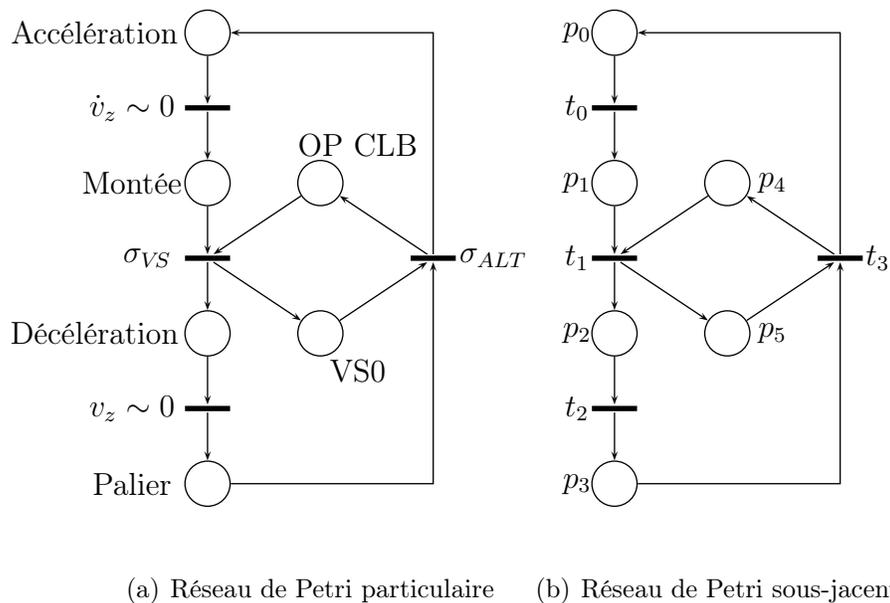


FIG. 9.2: Réseau de Petri modélisant une montée puis une mise en palier de l'avion à partir des consignes de l'équipage.

3.2 Détection de conflits

À partir de ce modèle, nous avons réalisé des expérimentations dans le simulateur de vol de Supaero. Au cours de ces simulations, nous demandions aux pilotes de se mettre en palier. Cette expérimentation a été réalisée avec trois pilotes. Nous avons utilisé un nombre de particules $N = 1000$ (figure 9.3).

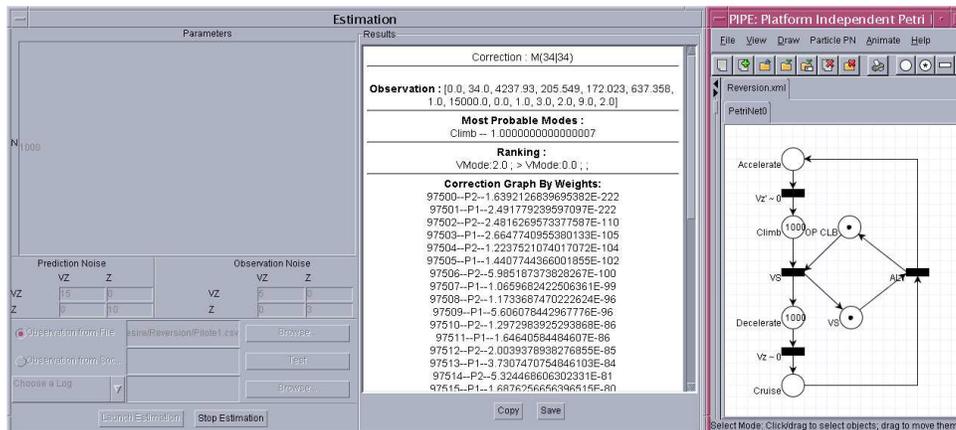


FIG. 9.3: Réseau de Petri particulière de mise en palier et estimation dans Exhost-PIPE.

3.2.1 Pilote numéro 1

Les figures 9.4 et 9.5 représentent respectivement l'évolution de l'altitude et de la vitesse verticale estimées de l'avion.

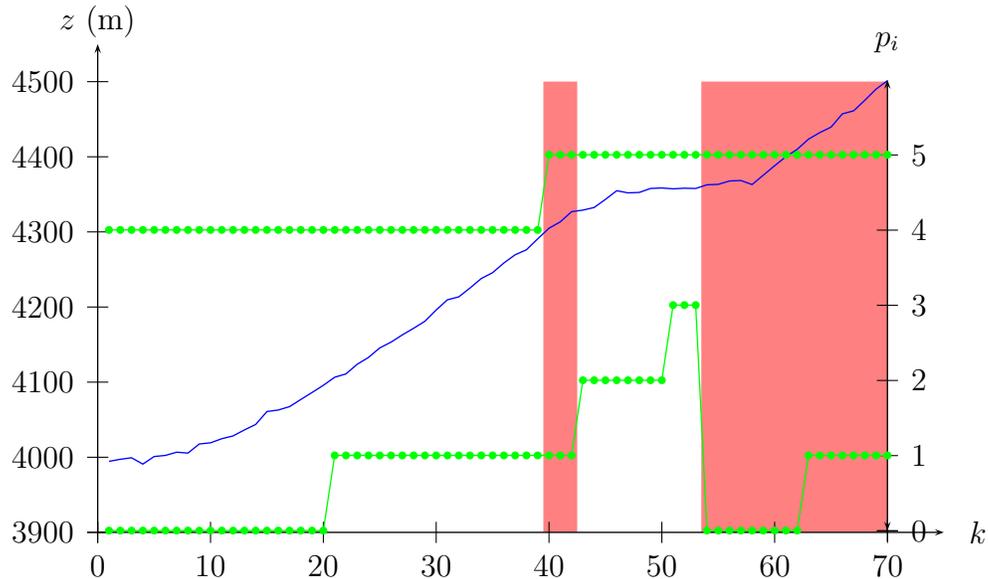


FIG. 9.4: Estimation de l'altitude de l'avion et détection d'incohérences pour le pilote 1. Les points verts représentent les modes les plus vraisemblables (la place numérique contenant le plus de particules, et la place symbolique préférée).

Les zones de détection d'incohérences sont représentées en rouge.

La première incohérence survient à l'instant 40, juste après que le pilote a pressé le bouton *VS* donnant la consigne de mise en palier. Elle est due à un léger retard entre la consigne et la réponse de l'avion.

L'incohérence qui débute à l'instant 54 représente en revanche un réel conflit dans l'activité de pilotage : l'avion s'est mis à cabrer, alors que le pilote n'a pas activé le mode *ALT*. Le marquage correspondant (p_0, p_5) est inaccessible.

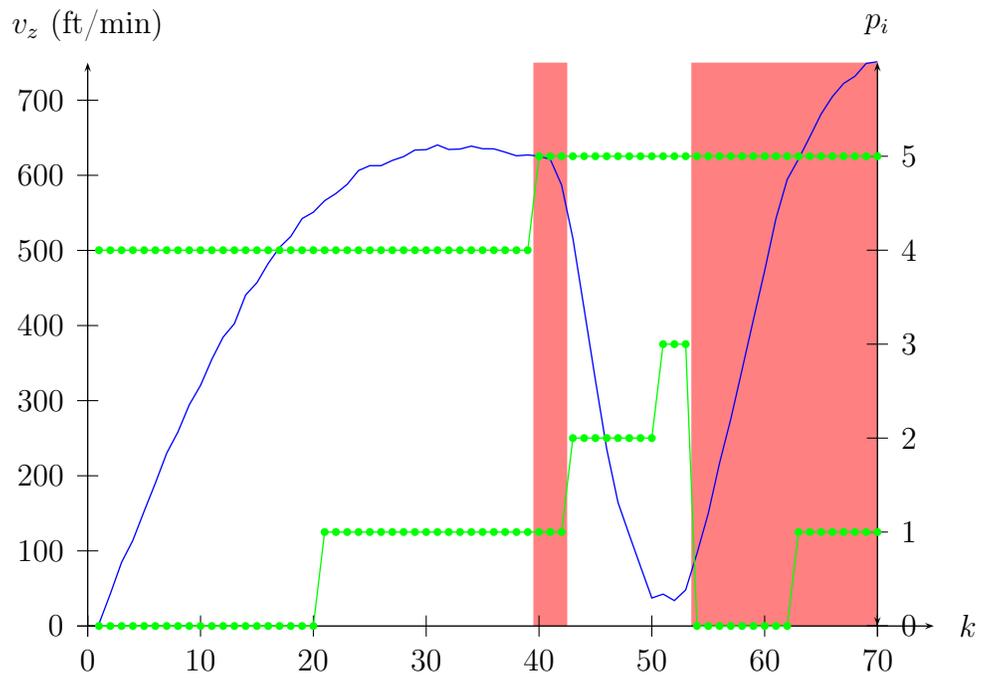


FIG. 9.5: Estimation de la vitesse verticale de l'avion et détection d'incohérences pour le pilote 1.

3.2.2 Pilote numéro 2

Les figures 9.6 et 9.7 représentent respectivement l'évolution de l'altitude et de la vitesse verticale estimées de l'ation.

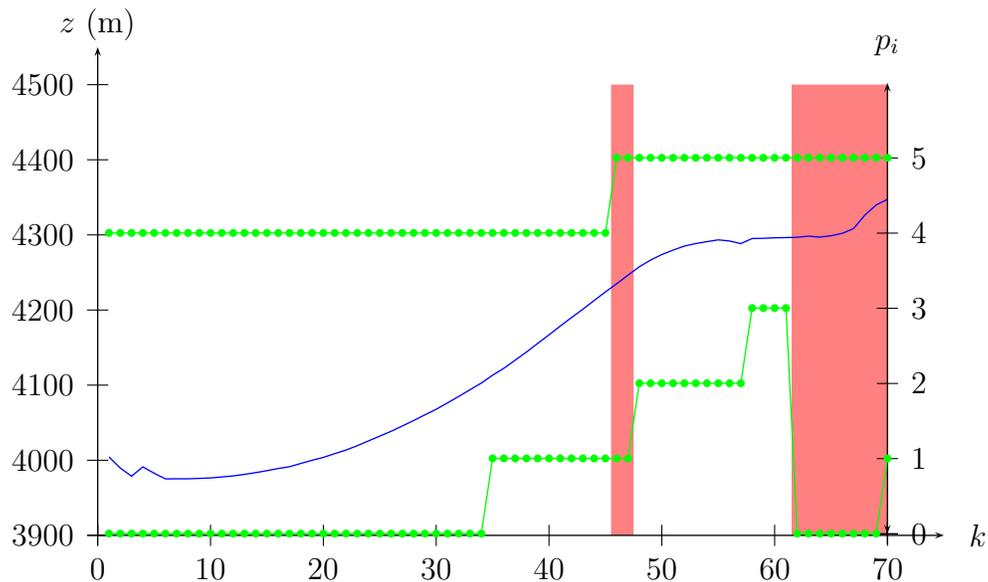


FIG. 9.6: Estimation de l'altitude de l'avion et détection d'incohérences pour le pilote 2.

Comme dans le cas du pilote 1, une incohérence est détectée à l'instant où le pilote donne la consigne de mise en palier ($k = 46$). À partir de l'instant 62, une incohérence est à nouveau détectée et signale un conflit entre la consigne de l'équipage (mise en palier) et le comportement de l'avion (montée).

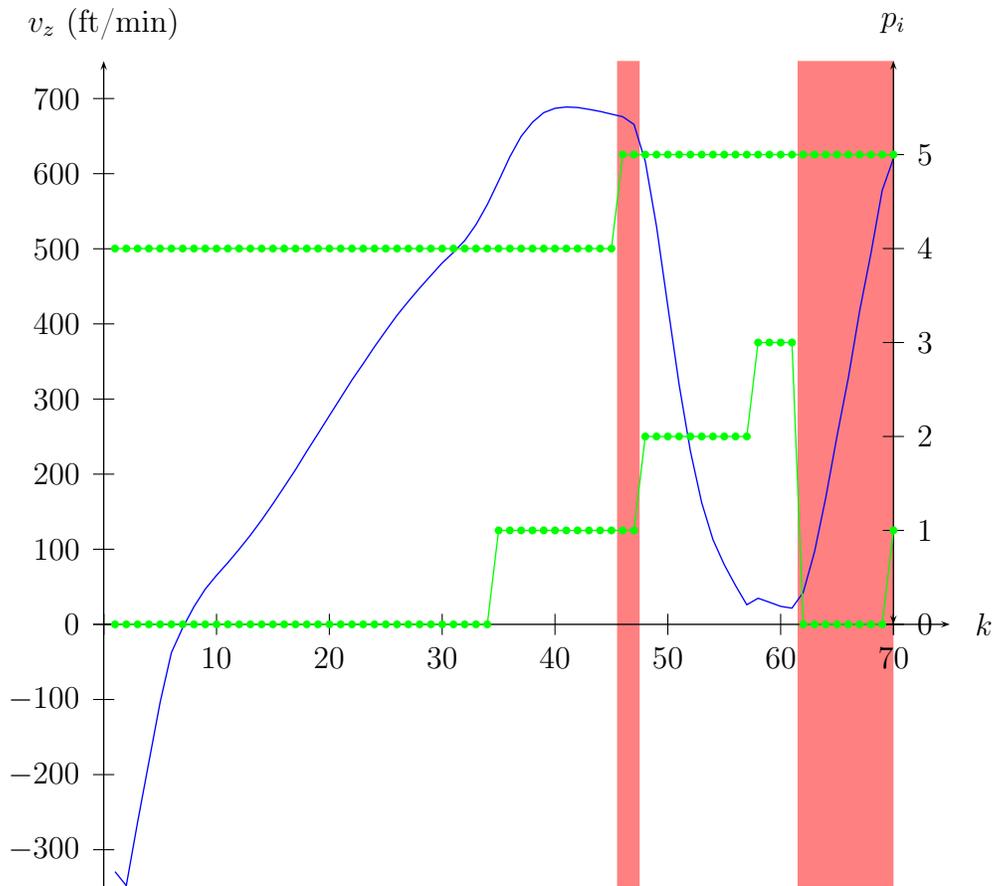


FIG. 9.7: Estimation de la vitesse verticale de l'avion et détection d'incohérences pour le pilote 2.

3.2.3 Pilote numéro 3

Les figures 9.8 et 9.9 représentent respectivement l'évolution de l'altitude et de la vitesse verticale estimées de l'avion.

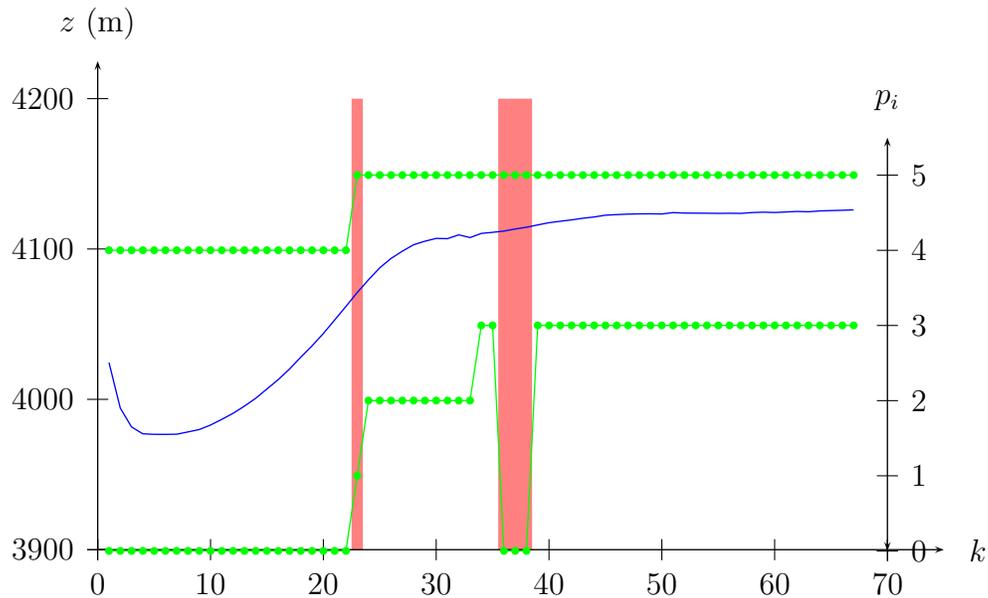


FIG. 9.8: Estimation de l'altitude de l'avion et détection d'incohérences pour le pilote 3.

Une fois de plus, l'incohérence à l'instant 23 est due au retard entre la consigne de mise en palier et le comportement de l'avion.

Dans le cas du pilote 3, le pilote automatique n'opère pas de réversion de mode, mais arrive à réguler la vitesse de l'avion pour éviter une sortie du domaine de vol par survitesse.

Cependant, une incohérence est détectée à l'instant 36, mais cesse à l'instant 39, dès que le comportement de l'avion est redevenu cohérent.

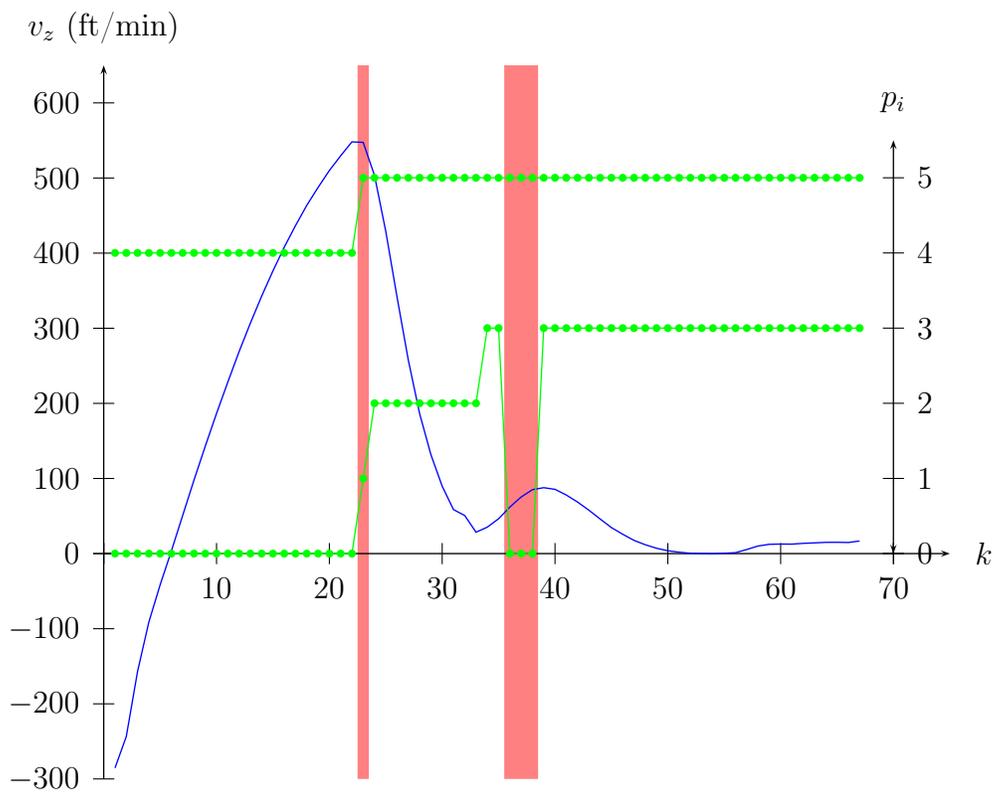


FIG. 9.9: Estimation de la vitesse verticale de l'avion et détection d'incohérences pour le pilote 3.

4 Conclusion

Nous voyons sur ces expérimentations que nous pouvons suivre l'état hybride de l'avion et détecter des incohérences entre cet état et les consignes de l'équipage. Plus particulièrement, nous avons évalué cette détection d'incohérences sur trois expérimentations sur un même scénario et :

- une incohérence “temporaire” est détectée dans chaque cas lors de la consigne de l'équipage, et provient d'un retard dans la réaction de l'avion ;
- une incohérence “permanente” est détectée dans les deux premiers cas lors de la réversion de mode du pilote automatique, qui est la cause d'un conflit entre le comportement de l'avion (montée) et le but du pilote (palier) ;
- dans le troisième cas, une incohérence est détectée, mais le pilote automatique parvient à corriger la trajectoire de l'avion et cette incohérence disparaît : le comportement de l'avion redevient cohérent.

Ces détections permettent de mettre en place un principe d'envoi de contre-mesures (Dehais *et al.*, 2005) sur les différents écrans de l'équipage pour l'aider à comprendre et résoudre cette situation (figure 9.10).

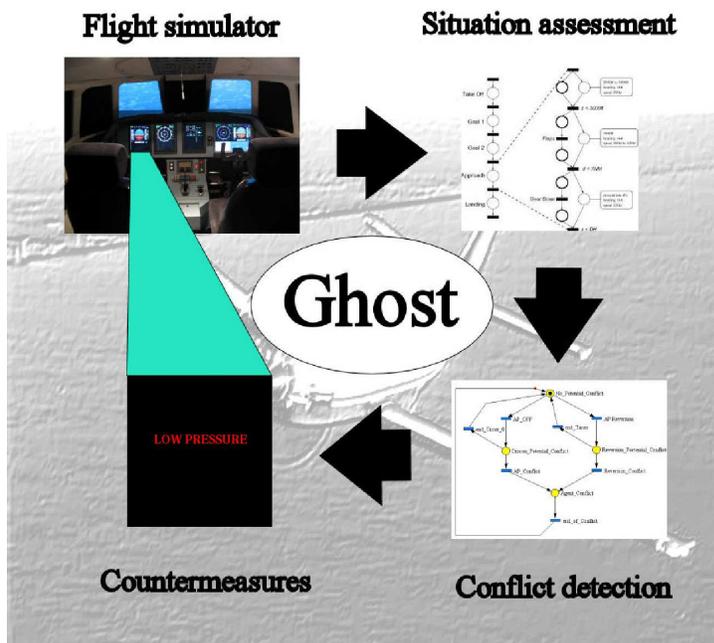


FIG. 9.10: Ghost : système de suivi de l'activité de pilotage et d'envoi de contre-mesures (de Dehais *et al.*, 2005)

D'autres expérimentations sont actuellement en cours pour évaluer ce principe sur d'autres situations de vol et d'autres réversions de mode. Ces expérimentations feront l'objet d'un document qui sera ajouté en annexe de ce manuscrit.

Conclusion

Contributions

Dans ce manuscrit, nous avons proposé et évalué une méthode d'estimation et de détection d'incohérences dans les systèmes hybrides.

Réseaux de Petri particuliers

Nous avons défini le formalisme des **réseaux de Petri particuliers** qui permet de modéliser le comportement hybride d'un système.

L'état d'un système hybride est représenté par un marquage du réseau, qui contient :

- un ensemble de **particules**, qui représentent le vecteur d'état numérique du système;
- un ensemble de jetons dont le marquage représente la **configuration** symbolique du système.

Les particules sont disposées dans des places numériques et leur évolution est décrite par des équations d'évolution associées à ces places.

La dynamique discrète est représentée par le franchissement des transitions, qui peut être dépendant de conditions ou d'événements, et modifie le mode de fonctionnement du système.

Nous avons ensuite proposé des propriétés permettant de caractériser les réseaux de Petri modélisant de façon "correcte" un système hybride, et défini la notion de **graphe d'état numérique**.

Le jeu des réseaux de Petri particuliers a également été défini, par l'alternance d'une **évolution isoparticule** et d'une **évolution isomarquage** qui représentent l'évolution hybride du système.

Ces définitions ont été illustrées par les exemples du thermostat et du problème de navigation.

Estimation par filtrage particulaire

Pour estimer l'état d'un système hybride, nous avons tout d'abord proposé un principe d'estimation du marquage d'un réseau de Petri particulaire par filtrage de Kalman. La difficulté de maintenir une distribution gaussienne lors de cette estimation nous a poussé à abandonner cette approche.

Nous avons alors proposé un principe d'**estimation par filtrage particulaire**. La représentation de l'incertitude sur l'état du système est assurée par :

- un ensemble de **particules**, qui représente les valeurs probables du système à un instant donné ;
- un **macromarquage**, qui représente l'ensemble des configurations possibles du système.

Ce principe d'estimation repose sur deux étapes :

- une **prédiction** de l'état du système, en utilisant des règles d'**évolution bruitée**, et qui permet d'obtenir le marquage $\widehat{\mathcal{M}}(k+1|k)$ prédit à l'instant $k+1$ à partir du marquage à l'instant k ;
- un **recalage** de l'état du système à partir d'une observation, qui se décompose en :
 - un recalage numérique, qui **pondère** les particules prédites à partir d'une fonction de vraisemblance, puis **rééchantillonne** ces particules en fonction de ces poids ;
 - un recalage symbolique, qui **classe** les places symboliques en fonction de la correspondance entre leurs configurations et l'observation symbolique, puis **sélectionne** les meilleures configurations ;
 et fournit ainsi, à partir de l'observation à l'instant $k+1$, le marquage estimé $\widehat{\mathcal{M}}(k+1|k+1)$.

Nous avons décrit ce principe sous forme algorithmique et étudié, pour chacun de ces algorithmes, sa terminaison et sa complexité théorique.

Nous avons vu que la terminaison de l'évolution isoparticule était délicate, fortement dépendante de la structure du réseau.

L'étude expérimentale de ces algorithmes a permis de confirmer les complexités théoriques, et nous avons pu conclure sur :

- la terminaison des algorithmes sur les instances du thermostat et de la navigation,
- une complexité linéaire des algorithmes en fonction du nombre de particules N ,
- une complexité polynomiale de la prédiction et constante du recalage en fonction de la taille du réseau $|\mathcal{R}|$.

Ces résultats sont très satisfaisants, et les complexités obtenues sont du même ordre

de grandeur que celles du filtrage particulière classique.

Détection d'incohérences

À partir de l'estimation du marquage d'un réseau de Petri particulière, nous avons proposé un principe pour la détection d'incohérences. Ce principe repose sur un **graphe des marquages cohérents** qui représente l'ensemble des marquages possibles du système étant donné son marquage initial.

Les particules estimées et les configurations sélectionnées sont ensuite utilisées pour construire deux graphes de recalage :

- un **graphe de recalage par places**, qui est constitué de couples (place numérique, place symbolique), classés à partir de la relation de préférence sur les configurations et de l'estimation de mode des places numériques ;
- un **graphe de recalage par poids**, qui est constitué de couples (classe, place symbolique) où les classes sont des classes d'équivalence de particules, obtenues à partir d'une relation de regroupement.

Pour chacun de ces graphes, les couples sont analysés pour détecter une **incohérence**, c'est-à-dire le fait qu'un couple ne puisse pas être obtenu à partir de l'état initial du système.

Lorsqu'un sommet d'un des graphes de recalage est incohérent, une incohérence dans le comportement du système est **détectée**.

Nous avons utilisé les problèmes de navigation et du thermostat pour illustrer le comportement de l'estimateur et de la détection d'incohérences. Les simulations sur le comportement de l'estimation numérique nous ont permis :

- d'observer l'évolution de l'estimation numérique, qui suit l'état numérique de chaque système ;
- d'observer la **convergence de l'erreur quadratique** sur l'estimation numérique ;
- d'observer la diminution de cette erreur lorsque le nombre de particules augmente, mais que cette diminution était très faible pour un grand nombre de particules ; nous avons alors conclu sur le fait qu'**un millier de particules** était suffisant pour assurer une estimation correcte ;
- d'observer que l'efficacité de l'estimateur était **indépendante de l'état initial**.

La détection d'incohérences sur l'exemple du thermostat nous a permis de mettre en avant l'importance du modèle (le modèle du thermostat induit forcément des fausses détections), et nous avons également pu constater que certaines défaillances étaient détectées (Thermo24) alors que d'autres étaient absorbées par l'estimation (Thermo26).

Finalement, nous avons appliqué ce principe à la détection de conflits dans l'activité de pilotage. Les réseaux de Petri particuliers nous ont permis de modéliser le comportement de l'avion et les interactions entre équipage et automatismes. Nous avons pu, sur un scénario et avec trois pilotes, détecter des situations incohérentes, révélatrices de conflits réels dans l'activité de pilotage.

Perspectives

Propriétés des réseaux de Petri particuliers

Bien que notre contribution apporte un principe de détection d'incohérences et des résultats prometteurs, son application doit s'appuyer sur un "bon" modèle du système hybride.

Comme nous l'avons vu dans le cas du thermostat, un modèle mal "conçu" du système peut entraîner des fausses détections, ce que l'on veut éviter le plus possible.

Nous avons également vu que l'algorithme d'évolution isoparticule était délicat, puisque sa terminaison ne peut pas être prouvée dans un cadre général. Cela signifie qu'il peut y avoir des modèles, même bien conçus (non-Zénon par exemple) pour lesquels le processus d'estimation ne peut pas être utilisé.

Il nous semble donc indispensable de poursuivre des travaux sur la structure des réseaux de Petri particuliers, de façon à pouvoir :

- caractériser les réseaux de Petri particuliers **non-Zénon** (pour lesquels l'évolution isoparticule non bruitée termine),
- caractériser les réseaux de Petri particuliers pour lesquels l'algorithme d'évolution isoparticule bruitée **termine**,
- définir des propriétés garantissant une détection d'incohérences "correcte".

Ces travaux pourraient prendre les directions suivantes :

Conditions suffisantes pour les réseaux non-Zénon

Nous avons déjà vu qu'un graphe des marquages accessibles avec une structure d'arbre garantissait un comportement non-Zénon du réseau de Petri particulière marqué. D'autres conditions suffisantes pour montrer qu'un réseau est non-Zénon pourraient notamment provenir de l'analyse des conditions associées aux transitions.

Un cycle dans le graphe des marquages accessibles montre qu'une (ou plusieurs) transition peut être sensibilisée plusieurs fois. Pour espérer que ce comportement ne soit pas Zénon, on peut par exemple montrer que l'ensemble des conditions associées aux transi-

tions de ce cycle ne peuvent pas être satisfaites. Par exemple, dans le cas du problème de navigation (figure 3.2), $p_0 \longleftrightarrow p_1$ est un cycle, et représente donc une séquence de franchissement infinie dans le réseau sous-jacent. En revanche, cette séquence ne peut pas exister dans le réseau de Petri particulière puisque les conditions menant de p_0 à p_1 ($x > 1$) et de p_1 à p_0 ($x < 1$) ont une intersection vide : quelle que soit la valeur de la particule marquant p_0 , elle ne pourra pas franchir indéfiniment les transitions t_0 et t_2 .

Ainsi l'analyse de la satisfaction des conditions associées aux transitions peut permettre de définir des caractéristiques d'un réseau de Petri non-Zénon.

Diviser pour mieux régner

L'étude de la satisfaction des conditions d'un cycle semble intéressante pour prouver des propriétés des réseaux de Petri particulière. L'exemple bref de la navigation le montre, mais on peut imaginer que d'autres cycles, plus complexes, soient satisfiables, et donc potentiellement Zénon.

Une piste de recherche est de décomposer le réseau de Petri particulière en structures élémentaires (boucles, choix, séquences...) et d'étudier chacune de ces structures séparément.

Cet idée rejoint la notion de réduction d'un réseau de Petri (où on identifie des structures), et nous conduit donc à étudier la **réduction** d'un réseau de Petri particulière. Cette réduction, utilisant les règles classiques, et éventuellement des règles spécifiques prenant en compte la satisfaction de conditions (à définir), pourrait fournir des conditions suffisantes pour des propriétés de non-Zénon ou de terminaison de l'algorithme d'évolution isoparticule.

Benchmarks pour l'estimation des systèmes hybrides

Avant de se lancer dans la recherche de conditions suffisantes pour la terminaison de l'algorithme, observer le comportement de l'estimation sur différents exemples semble utile. Ces exemples pourront éventuellement permettre de déterminer des structures ou des conditions dans lesquelles l'algorithme termine ou ne termine pas.

Le problème de navigation nous a permis de facilement étudier le fonctionnement du principe d'estimation, notamment en fournissant de nombreuses instances de taille et de dynamique différentes.

Le problème du thermostat nous a lui permis de voir fonctionner la détection d'incohérences. Cependant il a une structure très réduite (3 places numériques, 2 places symboliques) et est très limité.

Il nous semble alors intéressant de chercher d'autres problèmes permettant d'évaluer le comportement de l'estimation et de la détection d'incohérences. Ces problèmes doivent être représentatifs des systèmes hybrides et permettre non seulement d'évaluer notre contribution, mais également de la comparer à d'autres principes de la littérature (filtrage particulière classique, réseaux bayésiens, automates hybrides...)

Chaque approche est testée sur un système hybride particulier avec ses propres caractéristiques, et il est assez difficile de pouvoir réellement conclure sur le fait que telle approche soit plus appropriée dans tel cas. Les instances sur lesquelles nous avons travaillé ont été proposées pour la vérification de propriétés dans les systèmes hybrides, et certaines (comme le problème de navigation) peuvent être utilisables pour l'estimation.

Il nous semble nécessaire de proposer des instances de problèmes spécifiques de l'estimation et de la détection d'anomalies dans les systèmes hybrides. Parmi ces **benchmarks**, le système BIO-Plex nous semble intéressant, puisqu'il est issu d'un problème réel, et qu'il couple une évolution continue des concentrations en oxygène à l'arrivée d'événements externes (déplacement des personnes dans les chambres).

Extensions des réseaux de Petri particuliers

La modélisation de systèmes complexes utilise de nombreuses extensions des réseaux de Petri. Ces extensions permettent de prendre en compte plusieurs systèmes, un système comportant plusieurs composants, des contraintes sur des ressources de ce système, des connaissances sur les durées de tâches...

Ces extensions peuvent s'appliquer aux réseaux de Petri particuliers pour permettre une modélisation plus précise et informée du système.

Réseaux de Petri temporels

Les réseaux de Petri temporels permettent de représenter des contraintes de temps dans le système. Ces contraintes peuvent être liées à des durées (minimales ou maximales) de tâches, associées aux places, ou à des fenêtres de franchissement des transitions.

Ces données temporelles peuvent facilement être ajoutées à un réseau de Petri particulier pour représenter des contraintes entre les changements de modes.

Cependant, la gestion du temps, globale pour le processus d'estimation, et locale pour les places et les transitions, fait apparaître quelques difficultés au niveau du processus d'estimation. Par exemple, une transition temporelle aura une fenêtre de tir différente pour chaque particule, ce qui peut complexifier les règles de franchissement.

Utiliser un **réseau de Petri particulier temporel** pour représenter le fonctionnement d'un système hybride nécessite donc l'analyse de l'influence des intervalles de temps

sur les règles d'évolution liées au jeu et à l'estimation du modèle.

Réseaux de Petri modulaires/colorés

La décomposition d'un système complexe en sous-systèmes est souvent modélisée, dans le formalisme des réseaux de Petri, par la décomposition d'un réseau en sous-réseaux, ou par l'ajout de couleurs dans le réseau.

La modularité consiste en la modélisation du système par plusieurs réseaux. Le fonctionnement simultané de ces réseaux est modélisé par des fusions de places ou de transitions : ces fusions représentent des éléments communs à plusieurs réseaux, que ce soit une ressource, une activité commune (places fusionnées), ou une synchronisation des sous-systèmes (transitions fusionnées).

La **modularité** est facilement utilisable dans le cadre des réseaux de Petri particuliers. Les fusions de places et de transitions permettent de créer des liens entre différents réseaux, mais les règles d'évolution restent identiques.

L'ajout de couleurs à un réseau de Petri permet de modéliser différents systèmes ayant une dynamique proche. La coloration est une extension intéressante dans le cadre des réseaux de Petri particuliers, car elle permettrait :

- au niveau des places symboliques, de représenter la redondance dans certaines configurations du système ; par exemple, on peut représenter de deux couleurs différentes les modes de fonctionnement de deux caméras d'observation, ou de deux interfaces du pilote et du copilote d'un avion ; ces deux jetons évoluent dans la partie symbolique du réseau, mais représentent des sous-systèmes différents ;
- au niveau des places numériques, de représenter l'interaction entre plusieurs systèmes ; dans le modèle du thermostat, on pourrait par exemple représenter par différentes couleurs les températures de différentes pièces (benchmark de partage de thermostats), ou dans le domaine aéronautique, de suivre le comportement de plusieurs avions, de façon à détecter des conflits entre ces avions.

La **coloration** de jetons apporte de nouvelles perspectives pour l'utilisation des réseaux de Petri particuliers. Cependant, elle nécessite une redéfinition des règles de franchissement (bruitées et non bruitées), en prenant en compte les couleurs dans les conditions associées aux transitions. Elle nécessite également de revoir le principe de détection d'incohérences, puisque les incohérences ne proviennent plus nécessairement de couples numérique/symbolique, mais peuvent provenir des différentes combinaisons liées aux systèmes en interaction.

Réseaux de Petri stochastiques/possibilistes

Nous avons vu au début de ce manuscrit que l'incertitude était souvent représentée par des modèles stochastiques ou possibilistes. Ces représentations limitent le champ

d'application des modèles, en particulier lorsque aucune information n'est disponible sur le système et qu'il est impossible de fournir des valeurs (probabilités, possibilités) réalistes. Cependant, dans de nombreux systèmes, on peut disposer d'une telle information, au moins en partie (sur quelques événements).

Il nous semble ainsi intéressant de pouvoir modéliser, dans un réseau de Petri particulière, des connaissances diverses sur l'arrivée d'événements, modélisée par une **probabilité**, une **possibilité** ou simplement une **éventualité** comme c'est le cas dans le formalisme que nous avons proposé.

Utiliser ces différentes notions peut ainsi améliorer l'efficacité de l'estimation et de la détection d'incohérences. Mener des recherches dans cette voie nous semble novateur, même si l'utilisation conjointe de plusieurs formalismes risque de complexifier les règles d'évolution des modèles.

Détection et prédiction d'incohérences

Nous avons vu au chapitre 8 un exemple de détection d'incohérences sur le problème du thermostat. Les simulations nous ont permis de mettre en avant la détection de comportements défaillants.

Lors de ces détections nous avons pu remarquer que les deux types d'incohérences, à partir des graphes de recalage par places et de recalage par poids, coïncidaient la plupart du temps (détection d'incohérences "rouges"), mais qu'il arrivait qu'une seule méthode détecte une incohérence ("orange").

Nous avons également constaté, sur les exemples du thermostat, que les détections "oranges" précédaient les détections "rouges".

Cette constatation nous incite à réaliser d'autres simulations, pour observer si ce phénomène se généralise et pouvoir, le cas échéant, instaurer deux niveaux d'incohérences, représentant l'**anticipation** (orange) et la **détection** (rouge) d'incohérences.

Dans les expérimentations sur le scénario de mise en palier, les incohérences détectées étaient uniquement rouges. La définition de deux niveaux d'incohérences n'apporterait donc aucune information pour l'anticipation. En revanche, la construction du graphe de recalage par poids était motivée par l'information que nous espérions tirer de sa structure : ce graphe contient un grand nombre de nœuds (une vingtaine en moyenne sur la mise en palier). Ainsi, il serait intéressant de pouvoir étudier, au sein du graphe de recalage par poids, les nœuds incohérents. Ces nœuds, lorsqu'ils ne sont pas dans le noyau du graphe (auquel cas il y a détection d'incohérence), représentent des situations incohérentes peu vraisemblables, mais possibles.

Dans l'expérimentation avec le pilote 3, le graphe de recalage par poids à l'instant 35

(juste avant la détection d’une incohérence) contient 187 nœuds, et le deuxième nœud (par ordre de préférence) est incohérent.

Nous proposons donc de poursuivre les travaux sur la détection d’incohérences par l’analyse de la **position des nœuds incohérents** dans le graphe de recalage par poids, et de l’évolution de cette position au cours du temps. De plus, la notion de “distance” entre les nœuds du graphe (définie à partir des poids des classes par exemple) peut aider à affiner l’anticipation et la détection d’incohérences.

Conflits et contre-mesures dans l’activité de pilotage

Dans le cadre de l’application de l’estimation et de la détection d’incohérences au suivi de l’activité de pilotage, nous avons vu des expérimentations mettant en avant la notion de conflits entre équipage et automatisme, conflits détectés sous forme d’incohérences.

À partir de cette détection, il s’agit de proposer à l’équipage une information, sous forme de message visuel, lui permettant de réagir à cette incohérence de façon à rétablir la cohérence de la situation.

Une situation conflictuelle mène souvent à un processus de persévération chez l’équipage (Dehais *et al.*, 2003) qui rend les alarmes classiques peu efficaces. L’information est donc envoyée sous forme de **contre-mesure** : une information présente dans le cockpit et intervenant dans le conflit est remplacée par un message adapté.

Pour mettre en place ce type de message, il faut donc :

- identifier les instruments de vol sur lesquels l’équipage est focalisé,
- envoyer un message adapté à la situation détectée sur ces instruments.

Les recherches et les expérimentations futures sont prévues pour permettre l’**estimation des points de focalisation de l’équipage**, à partir de la phase de vol en cours, c’est-à-dire de l’état estimé du système avion-équipage.

De plus, étudier l’**historique des marquages estimés** du réseau de Petri particulière peut aider à la compréhension de la cause de l’incohérence détectée, et servir à la **génération de messages** clairs, adaptés à la situation, et aidant l’équipage à résoudre le conflit.

Bibliographie

- C. ALCHOURRÒN, P. GARDENFORS et D. MAKINSON : On the logic of theory change : partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50, 1985.
- A. AMES, A. ABATE et S. SASTRY : Sufficient conditions for the existence of Zeno behavior. *In IEEE Conference on Decision and Control*, Séville, Espagne, 2005.
- E. ANDRÉ, G. HERZOG et T. RIST : On the simultaneous interpretation of real world image sequences and their natural language description : the system SOCCER. *In European Conference on Artificial Intelligence (ECAI)*, Munich, Allemagne, 1988.
- 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*, 50(2), 2002.
- F. BALDUZZI, A. GIUA et C. SEATZU : Modelling manufacturing systems with First-Order Hybrid Petri Nets. *International Journal of Production Research*, 39(2), 2001.
- S. BENFERHAT, D. DUBOIS et H. PRADE : Kalman-like filtering and updating in a possibilistic setting. *In European Conference on Artificial Intelligence (ECAI)*, Berlin, Allemagne, 2000.
- S. BENFERHAT, S. LAGRUE et O. PAPINI : Belief change and pre-orders : a brief overview. *International Journal of Geographical Information Science*, 18(4), 2004.
- S. BENFERHAT, S. LAGRUE et O. PAPINI : Revision of partially ordered information : axiomatization, semantics and iteration. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Edimbourg, Ecosse, Royaume-Uni, 2005.
- H. BLOM et E. BLOEM : Particle filtering for stochastic hybrid systems. *In IEEE Conference on Decision and Control*, Atlantis, Les Bahamas, 2004.
- H. BLOM et J. LYGEROS : HYBRIDGE final report. Rap. tech., HYBRIDGE Project, 2005.

- J. BLOOM, C. CLARK, C. CLIFFORD, A. DUNCAN, H. KHAN et M. PAPANTONIOU : Platform Independent Petri net Editor – final report. Rap. tech., Imperial College, Londres, Royaume-Uni, 2003.
- O. BONNET-TORRÈS, P. DOMENECH, C. LESIRE et C. TESSIER : Exhost-PIPE : PIPE extended for two classes of monitoring Petri nets. *In International Conference on Application and Theory of Petri Nets (ATPN)*, Turku, Finlande, 2006.
- O. BONNET-TORRÈS et C. TESSIER : From team plan to individual plans : a Petri net-based approach. *In International Joint Conference on Autonomous Agent and Multiagent Systems (AAMAS)*, Utrecht, Pays-Bas, 2005.
- A. BOURJIJ et D. KOENIG : An original Petri net state estimation by a reduced Luenberger observer. *In American Control Conference*, San Diego, Californie, USA, 1999.
- T. BROIDA, S. CHANDRASHEKHAR et R. CHALLAPPA : Recursive 3D motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4), 1990.
- T. CALLANTINE : Activity tracking for pilot error detection from flight data. *In European Conference on Human Decision Making and Control*, Glasgow, Ecosse, Royaume-Uni, 2002.
- J. CARDOSO, R. VALETTE et D. DUBOIS : Fuzzy Petri nets : an overview. *In IFAC World Congress*, San Francisco, Californie, USA, 1996.
- J. CARDOSO, R. VALETTE et D. DUBOIS : Possibilistic Petri nets. *IEEE Transactions on Systems, Man and Cybernetics (SMC)*, 29(5), 1999.
- G. CASELLA et C. ROBERT : Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1), 1996.
- M. CHACHOUA et D. PACHOLCZYK : A symbolic approach to uncertainty management. *Applied Intelligence*, 13(3), 2000.
- R. CHAMPAGNAT : *Supervision des systèmes discontinus : définition d'un modèle hybride et pilotage en temps-réel*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, 1998.
- D. CORONA, A. GIUA et C. SEATZU : Marking estimation of Petri nets with silent transitions. *In IEEE Conference on Decision and Control*, Atlantis, Les Bahamas, 2004.
- C. COSSART : *Un estimateur symbolique pour le suivi de situation*. Thèse de doctorat, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (ENSAE / Supaero), Toulouse, France, 1999.

- C. COSSART et C. TESSIER : Filtering vs revision and update : let us debate! *In European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, Londres, Royaume-Uni, 1999.
- D. CRISAN et A. DOUCET : A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3), 2002.
- J. CROW, D. JAVAUX et J. RUSHBY : Models and mechanized methods that integrate human factors into automation design. *In International Conference on Human Computer Interaction – applied to aeronautics (HCI-AERO)*, 2000.
- R. DAVID et H. ALLA : *Discrete, continuous and hybrid Petri nets*. Springer, 2005.
- R. DEARDEN, T. WILLEKE, F. HUTTER, R. SIMMONS, V. VERMA et S. THRUN : Real-time fault detection and situational awareness for rovers : report on the Mars technology program task. *In IEEE Aerospace Conference*, Big Sky, Montana, USA, 2004.
- F. DEHAIS, P. FABIANI, C. LESIRE et C. TESSIER : Situation monitoring and conflict detection in human-centric autonomous systems. *In Moving Autonomy Forward Conference*, Grantham, Royaume-Uni, 2006.
- F. DEHAIS, C. LESIRE, C. TESSIER et L. CHAUDRON : Conflits et contre-mesures dans l'activité de pilotage. *In Congrès Francophone de Reconnaissance des Formes et d'Intelligence Artificielle (RFIA)*, Toulouse, France, 2004.
- F. DEHAIS, A. GOUDOU, C. LESIRE et C. TESSIER : Towards an anticipatory agent to help pilots. *In AAAI Fall Symposium on "From reactive to anticipatory cognitive embodied systems"*, Arlington, Virginie, USA, 2005.
- F. DEHAIS, C. TESSIER et L. CHAUDRON : GHOST : experimenting conflicts countermeasures in the pilot's activity. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexique, 2003.
- R. DEMOLOMBE et E. HAMON : What does it mean that an agent is performing a typical procedure? A formal definition in the Situation Calculus. *In International Joint Conference on Autonomous Agent and Multiagent Systems (AAMAS)*, Bologne, Italie, 2002.
- P. DOMENECH : Outil pour la simulation de réseaux de petri. Rap. tech., École Nationale Supérieure d'Électrotechnique, d'Électronique, d'Informatique, d'Hydraulique et des Télécommunications (ENSEEIH), option Génie Électrique Automatique, Toulouse, 2005.
- A. DOUCET, N. de FREITAS, K. MURPHY et S. RUSSELL : Rao-Blackwellised particle filtering for dynamic Bayesian networks. *In International Conference on Uncertainty in Artificial Intelligence (UAI)*, Stanford, Californie, USA, 2000.

- C. DOUSSON : *Suivi d'évolutions et reconnaissance de chroniques*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, 1994.
- C. DOUSSON et T. V. DUONG : Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Suède, 1999.
- C. DOUSSON, P. GABORIT et M. GHALLAB : Situation recognition : representation and algorithms. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Chambéry, France, 1993.
- D. DUBOIS, H. FARGIER et P. PERNY : On the limitations of ordinal approaches to decision-making. *In International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Toulouse, France, 2002.
- A. EL BOUZIDI : Amélioration de la qualité du code d'un logiciel de réseaux de Petri. Rap. tech., Licence professionnelle SIL, option Qualité du logiciel, Université Paul Sabatier, Toulouse, 2006.
- A. FANNI, A. GIUA et N. SANNA : Control and error recovery of Petri net models with event observers. *In International Workshop on Manufacturing and Petri Nets*, Toulouse, France, 1997.
- A. FEHNER et F. IVANCIC : Benchmarks for hybrid systems verification. *In International Workshop on Hybrid Systems : Computation and Control (HSCC)*, Philadelphie, Pennsylvanie, USA, 2004.
- P. FRANK : Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy. *Automatica*, 26(3), 1990.
- A. GELB, éd. *Applied Optimal Estimation*. The M.I.T. Press, 2001.
- W. GLOVER et J. LYGEROS : A stochastic hybrid model for air traffic control simulation. *In International Workshop on Hybrid Systems : Computation and Control (HSCC)*, Philadelphie, Pennsylvanie, USA, 2004.
- S. GONG et H. BUXTON : On the visual expectations of moving objects. *In European Conference on Artificial Intelligence (ECAI)*, Vienne, Autriche, 1992.
- N. GORDON, D. SALMOND et A. SMITH : Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings*, 140(2), 1993.
- A. GRASTIEN, M.-O. CORDIER et C. LARGOUËT : Extending decentralized discrete-event modelling to diagnose reconfigurable systems. *In International Workshop on Principles of Diagnosis (DX)*, Carcassonne, France, 2004.

- M. HOFBAUR : Hybrid estimation and its role in process automation. Habilitation thesis, Graz University of Technology, 2004.
- M. HOFBAUR et B. WILLIAMS : Mode estimation of probabilistic hybrid systems. *In International Workshop on Hybrid Systems : Computation and Control (HSCC)*, Stanford, Californie, USA, 2002.
- G. HORTON, V. KULKARNI, D. NICOL et K. TRIVEDI : Fluid stochastic Petri nets : theory, applications, and solution. *European Journal of Operations Research*, 105(1), 1998.
- A. HUNTER et W. LIU : Measuring the quality of uncertain information using possibilistic logic. *In European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, Barcelone, Espagne, 2005.
- S. INTILLE : *Visual recognition of multi-agent action*. Thèse de doctorat, Massachusetts Institute of Technology, 1999.
- R. KALMAN : A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82, 1960.
- R. KARLSSON, T. SCHÖN et F. GUSTAFSSON : Complexity analysis of the marginalized particle filter. *IEEE Transactions on Signal Processing*, 53, 2005.
- H. KATSUNO et A. MANDELZON : On the difference between updating a knowledge base and revising it. *In International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Cambridge, Massachusetts, USA, 1991.
- X. KOUTSOUKOS, J. KURIEN et F. ZHAO : Estimation of distributed hybrid systems using particle filtering methods. *In International Workshop on Hybrid Systems : Computation and Control (HSCC)*, Prague, République Tchèque, 2003.
- S. KURKOVSKY et R. LOGANANTHARAJ : Extension of Petri nets for representing and reasoning with tasks with imprecise durations. *Applied Intelligence*, 23(2), 2005.
- C. LARGOUËT et M.-O. CORDIER : Timed automata model to improve the classification of a sequence of images. *In European Conference on Artificial Intelligence (ECAI)*, Berlin, Allemagne, 2000.
- B. J. LEE, Y. H. JOO et J. B. PARK : Maneuvering target tracking using intelligent control techniques. *In International Conference on Control, Automation, and Systems (ICCAS)*, Muju, Corée, 2002.
- U. LERNER : *Hybrid Bayesian networks for reasoning about complex systems*. Thèse de doctorat, Stanford University, Stanford, Californie, USA, 2002.

- C. LESIRE : Suivi de l'activité de pilotage par prédiction et recalage. *In Manifestation des Jeunes Chercheurs en Sciences et Techniques de l'Information et de la Communication (MaJeCSTIC)*, Marseille, France, 2003.
- C. LESIRE : A numerical/symbolic estimator for activity tracking. A preliminary report. *In International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Whistler, Canada, 2004a. Doctoral Consortium.
- C. LESIRE : Suivi de l'activité de pilotage par prédiction et recalage. *Informations, Savoirs, Décisions, Médiations (ISDM)*, (13), 2004b.
- C. LESIRE et C. TESSIER : Particle Petri nets for aircraft procedure monitoring under uncertainty. *In International Conference on Application and Theory of Petri Nets (ATPN)*, Miami, Floride, USA, 2005a.
- C. LESIRE et C. TESSIER : Réseaux de petri particuliers pour l'estimation symbolico-numérique. *In Journées Formalisation des Activités Concurrentes (FAC)*, Toulouse, France, 2005b.
- C. LESIRE et C. TESSIER : Estimation and conflict detection in human controlled systems. *In International Workshop on Hybrid Systems : Computation and Control (HSCC)*, Santa Barbara, Californie, USA, 2006a.
- C. LESIRE et C. TESSIER : Réseaux de Petri particuliers : estimation et détection d'incohérences dans les systèmes hybrides. *Journal Européen des Systèmes Automatisés (JESA)*, (spécial Systèmes Dynamiques Hybrides), soumis.
- C. LESIRE et C. TESSIER : A hybrid model for situation monitoring and conflict prediction in human supervised autonomous systems. *In AAAI Spring Symposium on "To boldly go where no human-robot team has gone before"*, Stanford, Californie, USA, 2006b.
- N. MOËNNE-LOCCOZ, F. BRÉMOND et M. THONNAT : Recurrent Bayesian network for the recognition of human behaviors from video. *In International Conference on Computer Vision Systems (ICVS)*, Graz, Autriche, 2003.
- P. D. MORAL et L. MICLO : Branching and interacting particle systems approximations of feynman-kac formulae with applications to nonlinear filtering. *In Séminaire de Probabilités*, vol. 1729 de *Lecture Notes in Mathematics*, p. 1–145. Springer, 2000.
- K. MURPHY : *Dynamic Bayesian networks : representation, inference and learning*. Thèse de doctorat, University of California, Berkeley, Californie, USA, 2002.
- B. NG, A. PFEFFER et R. DEARDEN : Continuous time particle filtering. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Edimbourg, Ecosse, Royaume-Uni, 2005.

- T. NIELSEN et F. JENSEN : Alert systems for production plants : a methodology based on conflict analysis. *In European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, Barcelone, Espagne, 2005.
- M. OISHI, I. MITCHELL, A. BAYEN, C. TOMLIN et A. DEGANI : Hybrid verification of an interface for an automatic landing. *In IEEE Conference on Decision and Control*, Las Vegas, Nevada, USA, 2002.
- B. PANNETIER, K. BENAMEUR, V. NIMIER et M. ROMBAUT : VS-IMM using road map information for a ground target tracking. *In International Conference on Information Fusion*, Philadelphie, Pennsylvanie, USA, 2005.
- C. PETRI : *Communication with automata*. Thèse de doctorat, Université de Bonn, Allemagne, 1962.
- M. ROMBAUT, I. JARKASS et T. DENŒUX : State recognition in discrete dynamical systems using Petri nets and evidence theory. *In European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, Londres, Royaume-Uni, 1999.
- N. ROTA : *Contribution à la reconnaissance de comportements humains à partir de séquences vidéos*. Thèse de doctorat, Université de Sophia-Antipolis, Nice, France, 2001.
- N. ROTA et M. THONNAT : Activity recognition from video sequences using declarative models. *In European Conference on Artificial Intelligence (ECAI)*, Berlin, Allemagne, 2000.
- S. RUSSELL et P. NORVIG : *Artificial Intelligence : a modern approach*, chap. Probabilistic reasoning over time. Prentice Hall, 2002.
- M. SHANAHAN : A logical account of perception incorporating feedback and expectation. *In International Conference on Principles of Knowledge Representation and Reasoning (KR)*, Toulouse, France, 2002.
- P. SMETS et R. KENNES : *Uncertainty management in information systems*, chap. The transferable belief model for belief representation. Kluwer Academic, 1996.
- A. SOTO : Self adaptative particle filter. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Edimbourg, Ecosse, Royaume-Uni, 2005.
- C. TESSIER : Towards a commonsense estimator for activity tracking. *In AAAI Spring Symposium on "Logical Formalization of Commonsense Reasoning"*, Stanford, Californie, USA, 2003.

- H. VEERARAGHAVAN et N. PAPANIKOLOPOULOS : Combining multiple tracking modalities for vehicle tracking in traffic intersections. *In IEEE International Conference on Robotics and Automation (ICRA)*, La Nouvelle-Orléans, Louisiane, USA, 2004.
- E. VILLANI, J. PASCAL, P. MIYAGI et R. VALETTE : Differential predicate transition Petri nets and objects, an aid for proving properties in hybrid systems. *In IFAC Conference on Analysis and Design of Hybrid Systems (ADHS)*, Saint-Malo, France, 2003.
- V.-T. VU, F. BRÉMOND et M. THONNAT : Automatic video interpretation : a novel algorithm for temporal scenario recognition. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexique, 2003.
- G. WELCH et G. BISHOP : An introduction to the Kalman filter. Rap. tech., Department of Computer Science, University of North Carolina, 2001.
- B. WILLIAMS, M. INGHAM, S. CHUNG et P. ELLIOTT : Model-based programming of intelligent embedded systems and robotic space explorers. *In IEEE*, vol. 91, 2003.
- L. ZADEH : Fuzzy sets. *Information and Control*, 8(3), 1965.
- F. ZHAO, X. KOUTSOUKOS, H. HAUSSECKER, J. REICH et P. CHEUNG : Distributed monitoring of hybrid systems : a model-directed approach. *In International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, Washington, USA, 2001.

Annexe A

Le filtre de Kalman

Le filtre de Kalman est une technique d'estimation numérique fréquemment utilisée en pistage et en commande. Elle consiste à estimer de manière optimale l'état $\mathbf{x}(t)$ d'un système dynamique (avion, navire, robot, réseau...) soumis à des entrées déterministes et aléatoires à partir de mesures bruitées $\mathbf{z}(t)$, d'un modèle d'évolution de l'état et d'un modèle d'observation de l'état. Le modèle discret du filtre de Kalman est le plus couramment utilisé.

1 Modèle discret du système

1.1 Equation d'évolution

Le filtre de Kalman discret repose sur l'hypothèse d'un système dynamique linéaire discret donc l'évolution entre les instants k et $k + 1$ est décrite par une équation de la forme

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{u}_k + b_k \quad (\text{A.1})$$

où

- \mathbf{x}_k est le vecteur d'état à l'instant k ,
- F_k une matrice déterministe,
- \mathbf{u}_k un signal déterministe connu (appelé *commande*),
- b_k un bruit blanc gaussien centré ($\mathbf{E}[b_k] = 0$) de covariance Q_k , appelé *bruit d'état* (ou de modèle).

L'état initial \mathbf{x}_0 est une variable gaussienne de moyenne $\hat{\mathbf{x}}_0$ et de covariance \mathbf{C}_0 .

1.2 Equation d'observation

L'état de ce système est observé par des mesures \mathbf{z}_k liées à l'état \mathbf{x}_k par une équation de la forme

$$\mathbf{z}_k = H_k \mathbf{x}_k + v_k \quad (\text{A.2})$$

où

- \mathbf{z}_k est le vecteur d'observation,
- H_k une matrice déterministe,
- v_k un bruit blanc gaussien centré de covariance R_k , appelé *bruit de mesure* (ou d'observation).

2 Estimation optimale de l'état

La méthode d'estimation de l'état exposée ci-dessous est une des démonstrations les plus générales des équations du filtre de Kalman discret.

2.1 Estimation par récurrence

Par hypothèse, l'état initial \mathbf{x}_0 est une variable gaussienne, de moyenne $\hat{\mathbf{x}}_0$ et de covariance \mathbf{C}_0 . La prise en compte d'une première observation \mathbf{z}_0 se traduit par une modification des connaissances relatives à \mathbf{x}_0 : $\mathbf{p}(\mathbf{x}_0|\mathbf{z}_0)$ est calculée à partir de la loi de Bayes :

$$\mathbf{p}(\mathbf{x}_0|\mathbf{z}_0) = \frac{\mathbf{p}(\mathbf{x}_0)\mathbf{p}(\mathbf{z}_0|\mathbf{x}_0)}{\int \mathbf{p}(\mathbf{z}_0|\mathbf{x}_0)\mathbf{p}(\mathbf{x}_0) d\mathbf{x}_0} \quad (\text{A.3})$$

$\mathbf{p}(\mathbf{x}_0)$ est connue. Dans la mesure où \mathbf{z}_0 se déduit linéairement de v_0 (qui est une variable gaussienne) et de \mathbf{x}_0 , $\mathbf{p}(\mathbf{z}_0|\mathbf{x}_0)$ est une variable gaussienne de moyenne $H_0\mathbf{x}_0$ et de covariance R_0 .

La poursuite des calculs montre alors que $\mathbf{p}(\mathbf{x}_0|\mathbf{z}_0)$ est la densité de probabilité d'une gaussienne.

Raisonnons maintenant par récurrence et supposons que $\mathbf{p}(\mathbf{x}_k|\mathbf{z}_{1:k})$ soit une variable gaussienne, d'espérance $\hat{\mathbf{x}}_{k|k}$ et de covariance $\mathbf{C}_{k|k}$. Avant de disposer de l'observation \mathbf{z}_{k+1} , il est possible, à partir du modèle du système, de prédire la loi de \mathbf{x}_{k+1} connaissant $\mathbf{z}_1 \dots \mathbf{z}_k$ et \mathbf{x}_k . On peut montrer alors que $\mathbf{p}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ est également une variable gaussienne, de caractéristiques

$$\hat{\mathbf{x}}_{k+1|k} = F_k \hat{\mathbf{x}}_{k|k} + \mathbf{u}_k \quad (\text{A.4})$$

$$\mathbf{C}_{k+1|k} = F_k \mathbf{C}_{k|k} {}^t F_k + Q_k \quad (\text{A.5})$$

La prise en compte de l'observation \mathbf{z}_{k+1} permet ensuite d'affiner la prédiction réalisée en calculant $\mathbf{p}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1})$ grâce à la loi de Bayes :

$$\mathbf{p}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) = \frac{\mathbf{p}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})\mathbf{p}(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})}{\mathbf{p}(\mathbf{z}_{k+1}|\mathbf{z}_k)} \quad (\text{A.6})$$

Toutes les densités de la formule précédente sont calculables, et on montre alors que $\mathbf{p}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1})$ est une variable gaussienne, de caractéristiques

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}(\mathbf{z}_{k+1} - H_k \hat{\mathbf{x}}_{k+1|k}) \quad (\text{A.7})$$

$$\mathbf{C}_{k+1|k+1} = \mathbf{C}_{k+1|k} - K_{k+1} H_{k+1} \mathbf{C}_{k+1|k} \quad (\text{A.8})$$

$$K_{k+1} = \mathbf{C}_{k+1|k} {}^t H_{k+1} (H_{k+1} \mathbf{C}_{k+1|k} {}^t H_{k+1} + R_{k+1})^{-1} \quad (\text{A.9})$$

2.2 Equations du filtre de Kalman

Le raisonnement précédent est basé sur le calcul des densités de probabilité conditionnelle. Or l'objectif du filtre est l'estimation optimale de l'état \mathbf{x}_k . Si l'estimé de l'état est défini comme **l'estimé qui maximise la densité de probabilité de \mathbf{x}_k étant donné les observations $\mathbf{z}_1 \dots \mathbf{z}_k$** , alors les équations (A.3) à (A.9) correspondent aux équations du filtre de Kalman. En effet, l'estimé de l'état \mathbf{x}_k va correspondre à son espérance $\hat{\mathbf{x}}_k$.

Le filtre de Kalman discret est alors décrit par les équations :

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k} = F_k \hat{\mathbf{x}}_{k|k} + \mathbf{u}_k \\ \mathbf{C}_{k+1|k} = F_k \mathbf{C}_{k|k} {}^t F_k + Q_k \\ \hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}(\mathbf{z}_{k+1} - H_k \hat{\mathbf{x}}_{k+1|k}) \\ \mathbf{C}_{k+1|k+1} = \mathbf{C}_{k+1|k} - K_{k+1} H_{k+1} \mathbf{C}_{k+1|k} \\ K_{k+1} = \mathbf{C}_{k+1|k} {}^t H_{k+1} (H_{k+1} \mathbf{C}_{k+1|k} {}^t H_{k+1} + R_{k+1})^{-1} \end{cases} \quad (\text{A.10})$$

2.3 Propriétés

Comme $\mathbf{E}[\hat{\mathbf{x}}_k|k] = \mathbf{E}[\mathbf{x}_k]$, l'estimateur est **non biaisé**.

Les variables étant gaussiennes, l'estimé ne correspond pas seulement à l'estimé qui maximise la densité de probabilité conditionnelle, mais aussi à **l'estimé à variance minimale** et à **l'estimé optimal linéaire**.

Quelques descriptions et preuves des équations précédentes, ainsi que des variantes du filtre de Kalman adaptées aux systèmes non-linéaires ou en temps continu peuvent être trouvées dans (Welch et Bishop, 2001; Gelb, 2001)

Annexe B

Le filtrage particulaire

Le filtrage particulaire connaît un fort développement dans de nombreux domaines relevant des sciences et techniques de l'information et de la communication (localisation, navigation, poursuite, vision, robotique...) Il s'agit d'une méthode de simulation séquentielle, de type Monte-Carlo, c'est-à-dire d'une méthode dans laquelle des particules explorent l'espace d'état, en évoluant de manière indépendante comme le processus sous-jacent, et qui interagissent sous l'effet d'un mécanisme de sélection, qui concentre automatiquement les particules dans les régions d'intérêt de l'espace d'état.

1 Introduction

En toute généralité, le processus sous-jacent est une chaîne de Markov, ou un processus de Markov à temps continu, dont l'espace d'état peut être fini, continu, hybride, variable dans le temps... La seule condition étant qu'il soit possible de le **simuler** facilement.

Dans le cas du filtrage particulaire, le mécanisme de sélection est assuré par la fonction de vraisemblance : à chaque pas de temps, une particule a d'autant plus de chance de se reproduire à la génération suivante qu'elle est cohérente avec l'observation courante. Ces méthodes sont très faciles à mettre en œuvre, puisqu'il suffit de savoir simuler de manière indépendante des trajectoires du processus sous-jacent ; elles se situent à l'interface des algorithmes génétiques et des systèmes de particules en interaction.

Une description du filtrage particulaire est donnée dans (Arulampalam *et al.*, 2002). Les aspects mathématiques sont détaillés dans (Moral et Miclo, 2000; Crisan et Doucet, 2002).

2 Algorithme de base

De nombreux algorithmes de filtrage particulaire existent dans la littérature. Nous commencerons par décrire l'algorithme de base, le premier historiquement à proposer une étape de redistribution des particules.

2.1 Modèle d'état non-linéaire non-gaussien

On considère le problème simple suivant, où il s'agit d'estimer l'état \mathbf{x}_k d'un système au vu d'observations $\mathbf{z}_1 \dots \mathbf{z}_k$ recueillies dans un bruit blanc additif, pas nécessairement gaussien. En général, les observations seules ne permettent pas d'estimer l'état caché, et il est nécessaire de disposer d'un modèle *a priori* pour décrire l'évolution de l'état entre deux instants d'observation.

$$\mathbf{x}_k = F_k(\mathbf{x}_{k-1}, b_k) \quad (\text{B.1})$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + v_k \quad (\text{B.2})$$

où l'incertitude sur l'état initial est décrite par une distribution de probabilité \mathbf{p}_0 , et b_k et v_k sont des bruits blancs indépendants entre eux et indépendants de la condition initiale \mathbf{x}_0 .

Sous ces hypothèses, les états \mathbf{x}_k suivent une dynamique markovienne et conditionnellement aux états $\mathbf{x}_0 \dots \mathbf{x}_k$, les observations sont indépendantes et la distribution de l'observation \mathbf{z}_k ne dépend que de l'état \mathbf{x}_k au même instant. On note

$$g_k(x) = q_k(\mathbf{z}_k - h_k(x)) \quad (\text{B.3})$$

la fonction de vraisemblance associée, qui quantifie l'adéquation de chaque état possible x par rapport à l'observation courante \mathbf{z}_k . Cette fonction dépend de l'écart entre l'observation à laquelle on s'attend si le système occupait l'état x et l'observation réelle.

2.2 Algorithme

L'idée est de représenter la distribution de probabilité conditionnelle \mathbf{p}_k de l'état \mathbf{x}_k sachant les observations $\mathbf{z}_1 \dots \mathbf{z}_k$ à l'aide d'un système de particules caractérisé par les positions $\pi_k^1 \dots \pi_k^N$ et les poids $\omega_k^1 \dots \omega_k^N$ soumis au mécanisme d'évolution suivant :

pondération (ou mutation) entre deux observations : les particules explorent l'espace d'état de façon indépendante, en imitant le comportement suivi par l'état, c'est-à-dire en suivant le modèle *a priori* ;

pondération : lorsqu'une nouvelle observation est disponible, l'adéquation de chaque particule avec cette observation est évaluée grâce à la fonction de vraisemblance : chaque particule est alors affectée d'un poids proportionnel à la valeur calculée ;

sélection : les particules sont éliminées ou multipliées en fonction de leurs poids, c'est-à-dire que les particules auront d'autant plus de descendants à la génération suivante que leur vraisemblance est grande.

Pour pouvoir mettre en œuvre l'algorithme B.1, il suffit de savoir :

- simuler une variable aléatoire selon la distribution initiale,
- simuler une variable aléatoire selon la dynamique du modèle,
- calculer, à une constante multiplicative près, la fonction de vraisemblance : il faut pour cela que la probabilité d'émission de $\mathbf{z}_k | \mathbf{x}_k = x$ ait une densité par rapport à une mesure ne dépendant pas de x et qu'on dispose d'une expression explicite pour cette densité.

Algorithme B.1 Algorithme de filtrage particulaire

```

1: à l'instant initial  $k = 0$  :
2: pour  $i = 1, \dots, N$  faire
3:    $\pi_0^i \sim \mathbf{p}_0$ 
4: fin pour
5: à tout instant  $k \geq 1$  :
6: pour  $i = 1, \dots, N$  faire
7:    $b_k^i \sim \mathbf{p}(b_k)$ 
8:    $\zeta_k^i = F_k(\pi_{k-1}^i, b_k^i)$ 
9: fin pour
10: pour  $i = 1, \dots, N$  faire
11:    $\omega_k^i = g_k(\zeta_k^i)$ 
12: fin pour
13: pour  $i = 1, \dots, N$  faire
14:    $\pi_k^i \sim \sum_{j=1}^N \omega_k^j \delta_{\zeta_k^j}^i$ 
15: fin pour

```

L'étape de sélection, au cours de laquelle les particules sont éliminées ou multipliées, est cruciale pour l'efficacité de l'algorithme. Si cette étape est omise, tout se passe comme si N trajectoires indépendantes étaient simulées une fois pour toutes, en aveugle, avant même que les observations soient disponibles. Il n'y a aucune raison pour que, parmi des N trajectoires, il s'en trouve une seule qui soit proche de la vraie trajectoire. En pratique, on constate d'ailleurs que très rapidement, une trajectoire concentre tout le poids, les autres trajectoires ayant un poids quasi-nul et ne contribuant pas à l'estimation de l'état. La solution à ce problème de dégénérescence des poids consiste précisément à redistribuer les particules, de façon à mieux utiliser les informations disponibles.

(Ng *et al.*, 2005; Soto, 2005) fournissent des variantes du filtrage particulaire adaptée au temps continu où à l'adaptabilité du nombre de particules. La Rao-Blackwellisation (Casella et Robert, 1996) est également une variante permettant d'améliorer le fonctionnement du filtre, en particulier pour les systèmes hybrides.

Annexe C

Exhost-PIPE

EXHOST-PIPE (EXtended High-level Orienter and STandard Platform Independent Petri net Editor) est une extension de PIPE pour l'édition, l'analyse et le jeu de réseaux de Petri ordinaires, modulaires, colorés, temporels. Nous avons intégré dans Exhost-PIPE la possibilité de travailler sur les réseaux de Petri particuliers. Dans ce chapitre, nous présentons l'outil, sa conception et ses fonctionnalités.

1 De PIPE à Exhost-PIPE

Au cours de nos travaux est apparu le besoin de disposer d'un outil permettant de mettre en œuvre nos propositions. Cet outil devait permettre de modéliser des systèmes hybrides par réseaux de Petri particuliers, de jouer ces réseaux, et d'implanter facilement un processus d'estimation.

Les travaux présentés dans ce chapitre se sont déroulés en plusieurs étapes :

1. la recherche d'un outil performant et modifiable,
2. l'implantation du modèle des réseaux de Petri particuliers et du principe d'estimation,
3. une amélioration de la qualité du code de l'outil, pour le rendre disponible et facilement utilisable par d'autres personnes.

1.1 PIPE

Au vu du nombre conséquent d'outils disponibles sur l'internet pour la modélisation, l'analyse et le jeu des réseaux de Petri, nous avons décidé de rechercher un outil existant répondant à nos critères, et que nous pourrions ensuite modifier à loisirs.

Ce travail a fait l'objet d'un stage de fin d'études (Domenech, 2005).

PIPE (Platform Independent Petri net Editor) a été développé par le département des sciences de l'Imperial College de Londres (Bloom *et al.*, 2003). Il s'agit d'un logiciel Open Source dont le code est disponible sur l'internet¹. Il est codé en Java et, comme son nom l'indique, présente l'avantage d'être multi plate-forme.

PIPE nous a semblé intéressant pour son interface intuitive, ses modules de simulation et d'analyse, une documentation UML et la possibilité de le modifier facilement.

PIPE permet de gérer des réseaux de Petri ordinaires et stochastiques. Il utilise une architecture MCV² qui sépare les aspects modélisation (structure des réseaux de Petri), contrôle (gestion des événements liés à l'utilisateur) et affichage.

De plus, il est possible d'implanter des modules assez facilement. Les modules existants dans PIPE concernent l'analyse de propriétés des réseaux de Petri (bornitude, vivacité...)

1.2 Conception d'Exhost-PIPE

L'étape suivante a été de modifier PIPE pour apporter les spécificités liées aux réseaux de Petri modulaires, dans le cadre des travaux de Bonnet-Torrès et Tessier (2005) et des réseaux de Petri particuliers (chapitre 3).

Le développement d'Exhost-PIPE a grandement modifié la structure et le fonctionnement de PIPE. Exhost-PIPE compte 21 paquetages et 133 classes, pour 15 paquetages et 110 classes dans la version originale de PIPE. Nous avons donc augmenté de 21% la "taille" du logiciel.

1.3 Amélioration de la qualité

L'amélioration de la qualité du code d'Exhost-PIPE avait plusieurs objectifs :

- agrémenter la documentation,
- simplifier les méthodes complexes,
- distribuer une version "propre" d'Exhost-PIPE en tant que logiciel Open Source, pour faciliter sa prise en main et sa modification par de futurs utilisateurs.

Ce travail a fait l'objet d'un stage de licence (El Bouzidi, 2006).

¹<http://pipe2.sourceforge.net>

²Model-Controller-View

2 Exhost-PIPE

L'interface d'Exhost-PIPE (figure C.1) est décomposée en plusieurs parties.

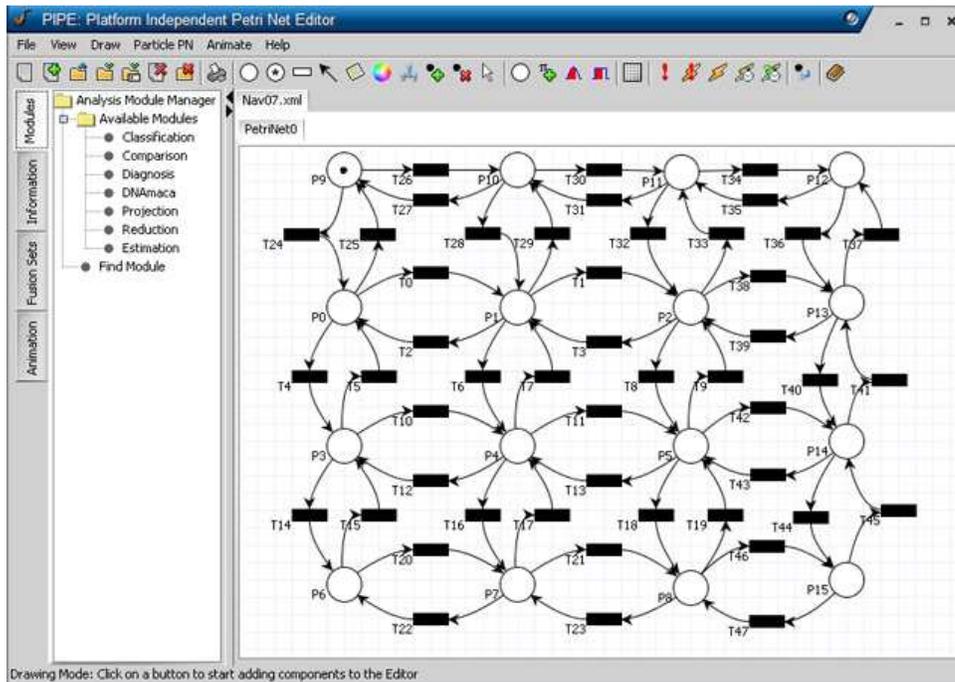


FIG. C.1: Interface d'Exhost-PIPE présentant le réseau de Petri particulière de l'instance Nav07.

Le haut de l'interface présente une barre de menu ainsi qu'une barre de boutons qui donnent accès aux fonctionnalités de création de réseaux de Petri, d'animation, et de gestion des fichiers (sauvegarde/chargement).

La partie centrale affiche le réseau courant. Les autres réseaux (ou sous-réseaux pour un réseau modulaire) sont accessibles via un ensemble d'onglets.

La partie gauche présente une série de panneaux d'affichage, disponibles par des onglets, et qui permettent d'accéder :

- à l'ensemble des modules disponibles (panneau *Modules*) ;
- aux informations liées au réseau (panneau *Information*, figure C.2), à savoir :
 - le marquage du réseau courant,
 - les propriétés d'une transition (nom, condition, taux de franchissement, couleur, ensemble de fusion),

- les propriétés d’une place (nom, couleur, marquage, équation d’évolution, ensemble de fusion),
 - les propriétés d’un arc (extrémités, couleur, poids);
- ces différentes informations sont disponibles en sélectionnant l’élément correspondant dans le panneau d’affichage;
- aux ensembles de fusion, de places ou de transitions, du réseau de Petri modulaire (panneau *Fusion Sets*);
 - à la gestion du jeu du réseau (panneau *Animation*), qui permet de connaître les transitions franchissables et, en fonction des modes, de pouvoir sélectionner la transition à tirer.

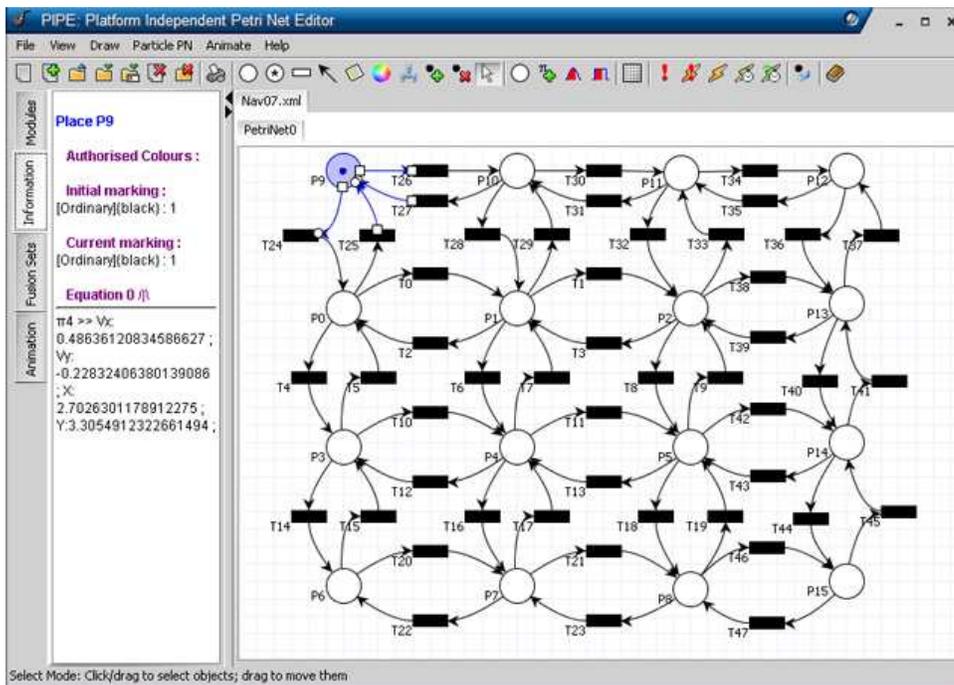


FIG. C.2: Interface d’Exhost-PIPE présentant les informations de la place p_9 .

2.1 Réseaux de Petri particulières

Exhost-PIPE permet de modéliser un réseau de Petri particulière. On peut :

- définir les paramètres du vecteur d’état numérique du système (figure C.3),
- créer des places numériques et définir leurs équations d’évolution, sous forme d’équation matricielles ($\mathbf{x}_{k+1} = A \cdot \mathbf{x}_k + B$),
- définir les configurations des places symboliques,

- définir des conditions pour les transitions portant sur les paramètres numériques du système,
- définir des événements attendus par certaines transitions.

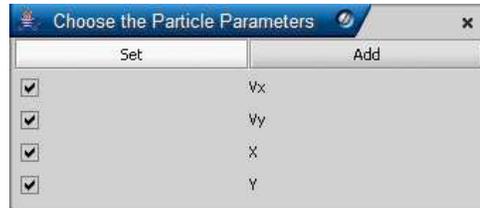


FIG. C.3: Panneau de définition des paramètres numériques du réseau de Petri particulaire.

2.2 Estimation

Avant de lancer l'estimation du système, nous pouvons créer le marquage initial de plusieurs façons :

- en créant les particules une à une dans le réseau,
- en générant, selon une distribution normale, N particules dans une place numérique,
- en générant, selon une distribution uniforme, N particules dans le réseau (figure C.4).

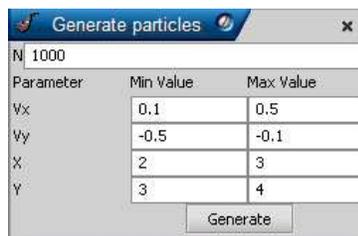


FIG. C.4: Panneau de génération de particules selon une loi uniforme.

Les algorithmes du principe d'estimation ont été implantés sous forme de modules. Lorsque le module *Estimation* est lancé, le panneau de contrôle s'affiche (figure C.5).

Sur ce panneau, il est possible de définir :

- le nombre de particules N ,
- les bruits de prédiction et d'observation, par leurs matrices de covariance,
- la source des observations, qui peut être un fichier, ou une connexion par socket,
- un fichier de log où sont enregistrées certaines informations, que l'on peut choisir parmi :
 - les temps de calcul,

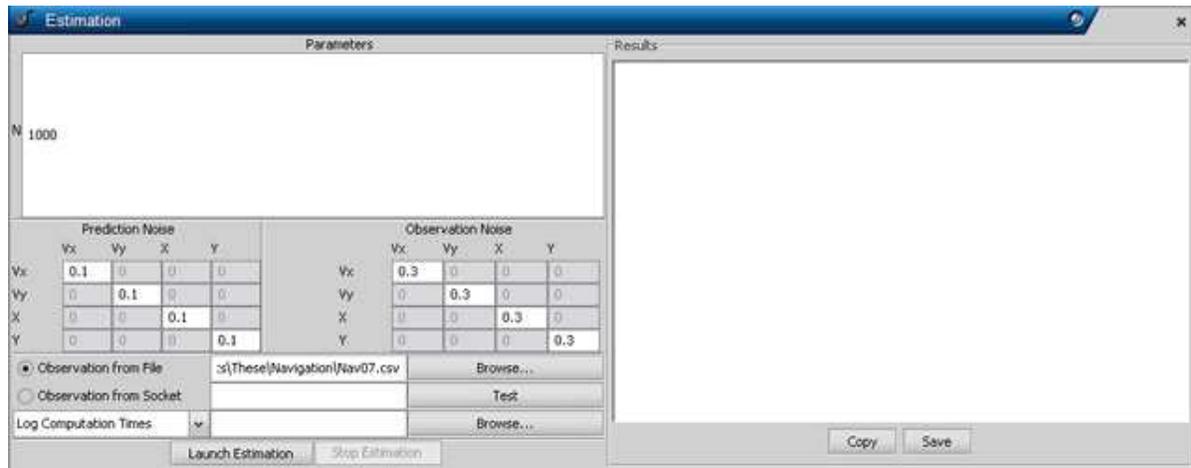


FIG. C.5: Panneau de contrôle du module d'estimation.

- l'estimation numérique moyenne, le mode numérique le plus probable et le mode symbolique préféré,
- l'ensemble des modes numériques.

Puis, lorsque l'estimation est lancée, ce panneau permet de contrôler le fonctionnement de l'estimateur (figure C.6), en fournissant des informations sur :

- le temps courant,
- l'observation reçue,
- les modes numériques estimés,
- le classement des configurations,
- les configurations supprimées,
- les classes d'équivalence des particules.

2.3 Résultats

Exhost-PIPE permet, dans le cadre de nos travaux, de modéliser des systèmes par réseaux de Petri particuliers, de jouer ces réseaux, et d'estimer l'état d'un système hybride à partir des algorithmes que nous avons définis.

Cette estimation peut se faire à partir d'un fichier (cas des simulations sur le problème de navigation et le thermostat) ou en recevant des paramètres par socket (connexion au simulateur de vol).

Les fichiers de sauvegarde de l'estimation ont permis d'obtenir l'ensemble des courbes de résultats présentées dans ce manuscrit.

Exhost-PIPE (Bonnet-Torrès *et al.*, 2006) a fait l'objet d'une démonstration au cours de la conférence ATPN'06, et est disponible en open source sur le site :

<http://www.onera.fr/dcsd-en/exhost-pipe/index.php>

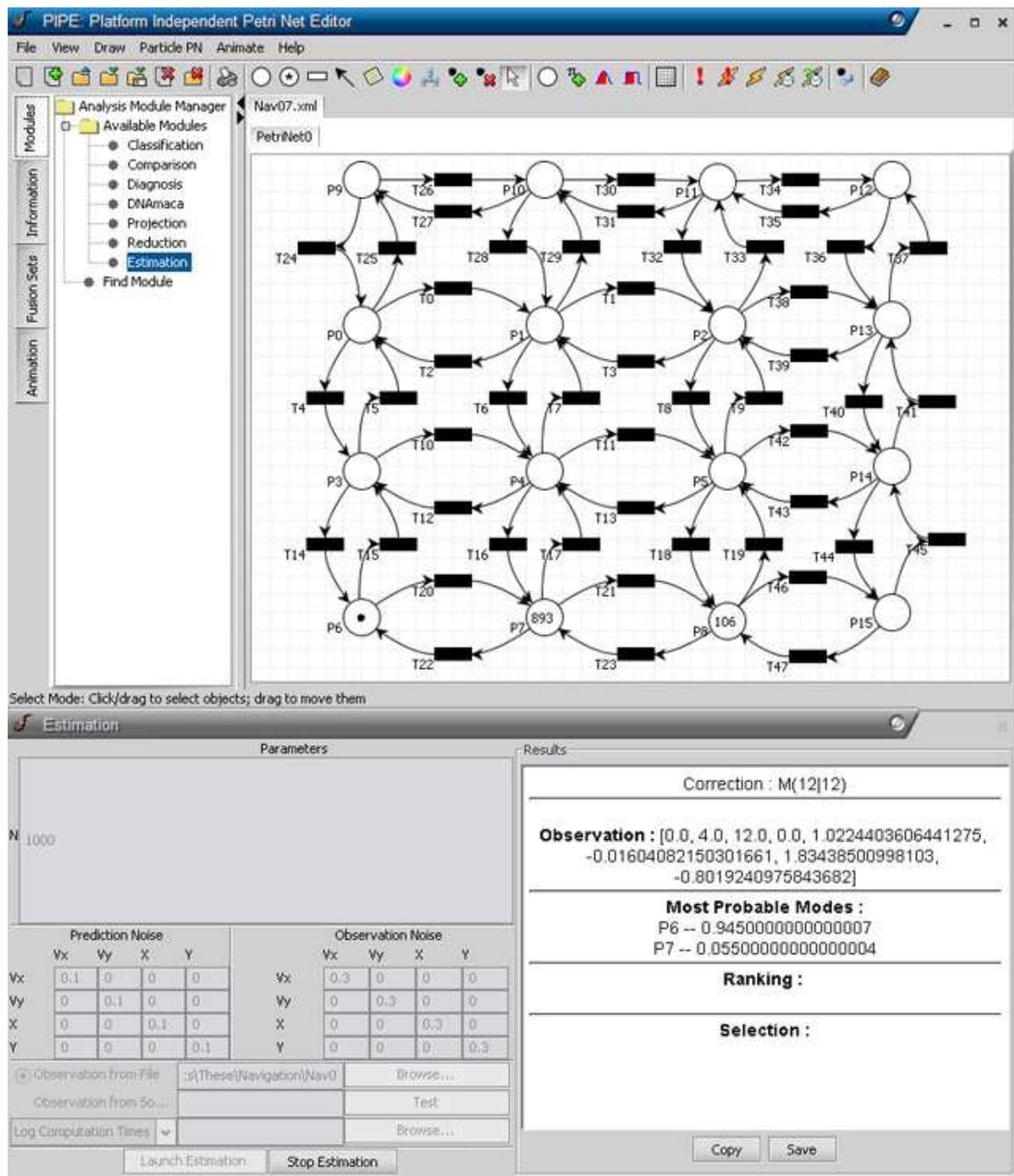


FIG. C.6: Estimation du système Nav07 : l'interface d'Exhost-PIPE fournit le marquage estimé courant, avec ses 1000 particules, dont les valeurs sont visibles dans le panneau d'information ; l'interface de l'estimation fournit les données liées au recalage (observation, estimation des modes numériques, classement et sélection des configurations, classes d'équivalence).

Annexe D

Problème de navigation

1 Description du problème

Le problème de navigation (Fehnker et Ivancic, 2004) consiste en l'étude d'un système en déplacement, dont le vecteur d'état contient sa position sur un plan et sa vitesse. Son mode de fonctionnement correspond à la zone de l'espace sur laquelle il se trouve.

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} \quad (\text{D.1})$$

La dynamique continue du système dépend de ce mode, et est définie par :

$$\dot{\mathbf{x}} = \begin{pmatrix} 0_2 & I_2 \\ 0_2 & \alpha \end{pmatrix} - \alpha.v_d \quad (\text{D.2})$$

où v_d est la vitesse désirée, dépendant du mode du système et α une constante de $\mathbb{R}^{2 \times 2}$.

La vitesse désirée est donnée par une direction désirée i . À partir de cette direction, la vitesse désirée est calculée selon l'équation :

$$v_d = \begin{pmatrix} \sin(i\frac{\pi}{4}) \\ \cos(i\frac{\pi}{4}) \end{pmatrix} \quad (\text{D.3})$$

2 Instances

Un certain nombre d'instances ont été définies par Fehnker et Ivancic (2004) et sont disponibles à l'adresse <http://www.cse.unsw.edu.au/~ansgar/benchmark/>. Les instances que nous utilisons dans ce manuscrit sont reprises ci-dessous.

2.1 Caractéristiques

Les instances sont définies par :

- un numéro, qui permet d'identifier l'instance ;
- la carte, qui donne la taille de l'instance et l'ensemble des directions désirées, sous forme de matrice M ;
- la matrice α ;
- la connaissance sur l'état initial du système, sous forme d'intervalles.

Nous ajoutons à cette description le nombre de places et de transitions correspondant à la modélisation de ces instances.

2.2 Description des instances

Nav01

$$M = \begin{pmatrix} B & 2 & 4 \\ 4 & 3 & 4 \\ 2 & 2 & A \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \quad \mathbf{x}_0 \in [2, 3] \times [1, 2] \times [-0.3, 0.3] \times [-0.3, 0]$$

Nav02

$$M = \begin{pmatrix} B & 2 & 4 \\ 4 & 3 & 4 \\ 2 & 2 & A \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \quad \mathbf{x}_0 \in [2, 3] \times [1, 2] \times [-0.3, 0.3] \times [-0.3, 0.3]$$

Nav03

$$M = \begin{pmatrix} B & 2 & 4 \\ 4 & 3 & 4 \\ 2 & 2 & A \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \quad \mathbf{x}_0 \in [2, 3] \times [1, 2] \times [-0.4, 0.4] \times [-0.4, 0.4]$$

Nav04

$$M = \begin{pmatrix} B & 2 & 4 \\ 2 & 2 & 4 \\ 1 & 1 & A \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \quad \mathbf{x}_0 \in [0, 1] \times [0, 1] \times [0.1, 0.5] \times [0.05, 0.25]$$

Nav05

$$M = \begin{pmatrix} B & 2 & 4 \\ 1 & 2 & 4 \\ 1 & 1 & A \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \quad \mathbf{x}_0 \in [0, 1] \times [0, 1] \times [0.3, 0.5] \times [0.05, 0.25]$$

Nav06

$$M = \begin{pmatrix} B & 2 & 4 \\ 3 & 0 & 4 \\ 2 & 1 & A \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \quad \mathbf{x}_0 \in [0, 1] \times [0, 1] \times [0.3, 0.5] \times [0.05, 0.25]$$

Nav07

$$M = \begin{pmatrix} 0 & 2 & 3 & 4 \\ B & 2 & 3 & 4 \\ 4 & 3 & 4 & 4 \\ 2 & A & 6 & 6 \end{pmatrix} \quad \alpha = \begin{pmatrix} -0.8 & -0.2 \\ 0.2 & -0.8 \end{pmatrix} \\ \mathbf{x}_0 \in [2, 3] \times [3, 4] \times [0.1, 0.5] \times [-0.5, -0.1]$$

Nav08

$$M = \begin{pmatrix} 4 & 6 & 3 & 4 \\ B & 6 & 3 & 4 \\ 0 & 7 & 4 & 4 \\ 2 & A & 6 & 6 \end{pmatrix} \quad \alpha = \begin{pmatrix} -0.8 & -0.2 \\ 0.2 & -0.8 \end{pmatrix} \\ \mathbf{x}_0 \in [2, 3] \times [3, 4] \times [0.1, 0.5] \times [-0.5, -0.1]$$

Nav09

$$M = \begin{pmatrix} 4 & 6 & 4 & 4 \\ B & 6 & 4 & 4 \\ 0 & 7 & 5 & 5 \\ 2 & A & 6 & 6 \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.8 & -0.2 \\ -0.2 & -1.8 \end{pmatrix} \\ \mathbf{x}_0 \in [2.5, 3.5] \times [3, 4] \times [-0.1, 0.1] \times [-0.5, -0.1]$$

Nav10

$$M = \begin{pmatrix} 2 & 4 & 6 & 6 & 6 \\ 2 & 4 & 7 & 7 & 4 \\ 2 & 4 & B & 3 & 4 \\ 2 & 4 & 6 & 6 & 6 \\ 2 & A & 0 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -0.8 & -0.1 \\ -0.2 & -0.8 \end{pmatrix} \\ \mathbf{x}_0 \in [3.5, 4] \times [3.5, 4] \times [-0.5, 0.5] \times [-0.5, 0]$$

Nav11

$$M = \begin{pmatrix} 2 & 4 & 6 & 6 & 6 \\ 2 & 4 & 7 & 7 & 4 \\ 2 & 4 & B & 3 & 4 \\ 2 & 4 & 6 & 6 & 6 \\ 2 & A & 0 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -0.8 & -0.1 \\ -0.2 & -0.8 \end{pmatrix} \\ \mathbf{x}_0 \in [3.5, 4] \times [3.5, 4] \times [-1, 0.5] \times [-1, 0.5]$$

Nav12

$$M = \begin{pmatrix} 2 & 4 & 6 & 6 & 6 \\ 2 & 4 & 7 & 7 & 4 \\ 2 & 4 & B & 3 & 4 \\ 2 & 4 & 6 & 6 & 6 \\ 2 & A & 0 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -0.8 & -0.1 \\ -0.2 & -0.8 \end{pmatrix}$$

$$\mathbf{x}_0 \in [3, 3.5] \times [3, 3.5] \times [0.5, 0.5] \times [-0.5, 0.5]$$

Nav13

$$M = \begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 4 & 4 & 6 & 6 & 6 & 6 \\ 2 & 4 & A & 6 & 6 & 6 & 6 \\ 2 & 4 & B & 1 & 7 & 7 & 0 \\ 2 & 4 & 2 & 1 & 7 & 7 & 0 \\ 2 & 2 & 2 & 1 & 1 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.2 \\ 0.1 & -1.2 \end{pmatrix}$$

$$\mathbf{x}_0 \in [4.5, 5.5] \times [1.5, 2.5] \times [0.5, 0.5] \times [-0.5, 0.5]$$

Nav14

$$M = \begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 4 & 4 & 6 & 6 & 6 & 6 \\ 2 & 4 & A & 7 & 7 & 7 & 6 \\ 2 & 4 & B & 1 & 7 & 7 & 0 \\ 2 & 4 & 2 & 1 & 7 & 7 & 0 \\ 2 & 2 & 2 & 1 & 1 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.2 \\ 0.1 & -1.2 \end{pmatrix}$$

$$\mathbf{x}_0 \in [4.5, 5.5] \times [1.5, 2.5] \times [0.5, 0.5] \times [-0.5, 0.5]$$

Nav15

$$M = \begin{pmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 4 & 5 & 6 & 6 & 6 & 6 \\ 2 & 4 & A & 7 & 7 & 7 & 6 \\ 2 & 4 & B & 1 & 7 & 7 & 0 \\ 2 & 4 & 2 & 1 & 7 & 7 & 0 \\ 2 & 2 & 2 & 1 & 1 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -1.2 & 0.2 \\ 0.1 & -1.2 \end{pmatrix}$$

$$\mathbf{x}_0 \in [4.5, 5.5] \times [1.5, 2.5] \times [0.5, 0.5] \times [-0.5, 0.5]$$

Nav16

$$M = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 4 & 4 & 4 & 4 \\ 1 & 1 & 3 & 2 & 3 & 4 & 4 & B & 4 \\ 1 & 1 & 3 & 2 & 3 & 2 & 3 & 4 & 5 \\ 1 & 1 & 3 & 2 & 3 & 2 & 3 & 4 & 5 \\ 1 & 1 & 3 & 2 & 3 & 2 & 3 & 4 & 5 \\ 1 & 1 & 3 & 2 & 3 & 2 & 3 & 4 & 6 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & A & 6 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \quad \alpha = \begin{pmatrix} -0.2 & -0.2 \\ 0 & -0.7 \end{pmatrix}$$

$$\mathbf{x}_0 \in [0.5, 1.5] \times [0.5, 1.5] \times [0, 0.5] \times [0, 0.5]$$

Nav17

$$M = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & B & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 3 & 4 & 5 \\ 0 & 0 & 3 & 3 & 3 & 2 & 3 & 4 & 6 \\ 0 & 0 & 3 & 3 & 3 & 3 & 3 & A & 6 \\ 1 & 1 & 2 & 2 & 3 & 3 & 3 & 0 & 0 \\ 1 & 1 & 2 & 2 & 2 & 2 & 2 & 0 & 0 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} -1.5 & -0.2 \\ -0.5 & -1.2 \end{pmatrix}$$

$$\mathbf{x}_0 \in [0.5, 1.5] \times [0.5, 1.5] \times [0, 0.5] \times [0, 0.5]$$

Nav18

$$M = \begin{pmatrix} 2 & 2 & 2 & 2 & 3 & 4 & 4 & 1 & 1 \\ 0 & 0 & 1 & 2 & 3 & 4 & 4 & B & 1 \\ 0 & 0 & 1 & 2 & 3 & 2 & 4 & 4 & 4 \\ 0 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 6 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & A & 6 \\ 1 & 1 & 2 & 2 & 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 2 & 2 & 2 & 2 & 2 & 0 & 0 \end{pmatrix}$$

$$\alpha = \begin{pmatrix} -1.2 & -0.2 \\ 0.5 & -0.7 \end{pmatrix}$$

$$\mathbf{x}_0 \in [0.5, 1.5] \times [0.5, 1.5] \times [0, 0.5] \times [0, 0.5]$$

2.3 Taille des instances

Instance	Grille	$ P $	$ T $	$ \mathcal{R} = P + T $
Nav01 Nav02 Nav03 Nav04 Nav05 Nav06	3×3	9	24	33
Nav07 Nav08 Nav09	4×4	16	48	64
Nav10 Nav11 Nav12	5×5	25	80	105
Nav13 Nav14 Nav15	6×7	42	142	184
Nav16 Nav17 Nav18	9×9	81	288	369

TAB. D.1: Taille des instances du problème de navigation.

Annexe E

Problème du thermostat

1 Description du problème

L'état du système que l'on veut estimer est composé d'un vecteur numérique contenant la température de la pièce contrôlée par le thermostat et la température extérieure :

$$\mathbf{x} = \begin{pmatrix} \theta \\ \theta_{ext} \end{pmatrix} \quad (\text{E.1})$$

Le mode de fonctionnement du système est $\boldsymbol{\mu} \in \{on, idle, off\}$.

L'évolution de la température de la pièce est régie par l'équation :

$$\dot{\mathbf{x}} = \alpha + \beta \cdot \mathbf{x} \quad (\text{E.2})$$

2 Instances

Les instances que nous proposons ici sont décrites par les constantes α et β et par les conditions initiales.

Thermo01

$$\alpha = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad \beta = \begin{pmatrix} -0.1 & 0.1 \\ 0 & 0 \end{pmatrix} \quad \mathbf{x}_0 \in [19, 20] \times [9, 10]$$

Thermo02

$$\alpha = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \quad \beta = \begin{pmatrix} -0.2 & 0.2 \\ 0 & 0 \end{pmatrix} \quad \mathbf{x}_0 \in [19, 20] \times [10, 12]$$

Thermo03

$$\alpha = \begin{pmatrix} 6 \\ 0 \end{pmatrix} \quad \beta = \begin{pmatrix} -0.3 & 0.3 \\ 0 & 0 \end{pmatrix} \quad \mathbf{x}_0 \in [19, 20] \times [6, 7]$$

Résumé Nous proposons un modèle pour la supervision des systèmes hybrides et la détection d'incohérences dans leur comportement : les réseaux de Petri particuliers. Ce modèle permet une représentation de la dynamique discrète du système par la structure du réseau de Petri et une représentation de la dynamique continue par des équations d'évolution associées aux places du réseau. À partir de ce modèle, nous définissons un principe d'estimation récursive, qui prédit le comportement du système à partir du jeu du réseau, et met à jour cette prédiction à partir d'une observation du système. Les résultats et propriétés sont donnés sur deux exemples classiques : le problème de navigation et le problème du thermostat. L'estimation du comportement du système nous permet ensuite de comparer les jetons numériques recalés et les jetons symboliques recalés. Cette comparaison permet de mettre en évidence les états qui sont incohérents, c'est-à-dire inatteignables étant donné l'état initial et le modèle du système. La présence d'incohérences permet de suspecter une situation anormale. Nous avons finalement appliqué ce principe d'estimation et de détection d'incohérences au suivi de l'activité de pilotage d'un avion de ligne.

Mots-clés Systèmes hybrides, Estimation, Détection d'incohérences, Réseaux de Petri, Filtrage particulière, Activité de pilotage.

Numerical/symbolic estimation for hybrid activity tracking

Abstract A model for hybrid system supervision and the detection of inconsistencies in their behaviours is proposed : particle Petri nets. This model allows to represent the discrete dynamics of the system through the Petri net structure and the continuous dynamics by evolution equations associated to places of the net. From this model, a recursive estimation process has been defined to predict the system behaviour using the Petri net firing rules and update the prediction from an observation of the system. Some results and properties are given on two examples : a navigation problem and a thermostat problem. The estimation of the system behaviour allows corrected numerical tokens and corrected symbolic tokens to be compared. The comparison highlights inconsistent states, i.e. the states that are unreachable from the initial state and the system model. Inconsistencies may reveal an abnormal situation. We have finally applied the estimation principle and the inconsistency detection to the tracking of an airliner pilot's activity.

Keywords Hybrid systems, Estimation, Inconsistency detection, Petri nets, Particle filtering, Pilot's activity.

