



HAL
open science

Diagnostic des systèmes hybrides: développement d'une méthode associant la détection par classification et la simulation dynamique

Aimed Mokhtari

► To cite this version:

Aimed Mokhtari. Diagnostic des systèmes hybrides: développement d'une méthode associant la détection par classification et la simulation dynamique. Automatique / Robotique. INSA de Toulouse, 2007. Français. NNT: . tel-00200034

HAL Id: tel-00200034

<https://theses.hal.science/tel-00200034>

Submitted on 20 Dec 2007

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

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

THÈSE

Préparée au

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

En vue de l'obtention du titre de *Docteur de l'Université de Toulouse*

Délivré par l'Institut National des Sciences Appliquées de Toulouse

Spécialité

Systèmes industriels

Par

Aimed MOKHTARI

Diagnostic des systèmes hybrides : développement d'une méthode associant la détection par classification et la simulation dynamique

Soutenue le 23 octobre 2007 devant le jury :

Président : Jean-Marc Le Lann

Rapporteurs : Claudia Frydman
Hervé Guéguen

Directeur de thèse : Marie-Véronique Le Lann

Co- encadrant Gilles Hétreux

Examineur : Naly Rakoto

Avant-propos

Je voudrais utiliser quelques lignes de ce document pour exprimer ma gratitude aux personnes qui m'ont aidé à accomplir l'un des plus importants projets professionnels de ma vie.

Les travaux présentés dans ce mémoire ont été réalisés au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS-CNRS) au sein du groupe Diagnostic, Supervision et Conduite qualitatifs (DISCO). Je tiens tout d'abord à remercier les directeurs successifs du LAAS, Messieurs Malik Ghallab et Raja Chatilla pour leur accueil.

Je tiens à remercier Monsieur Joseph Aguilar-Martin et Madame Louise Travé-Massuyès, successivement responsables du groupe DISCO, de m'y avoir accueilli pendant ces années de thèse.

J'exprime toute ma gratitude à mon Directeur de Recherche, Madame Marie-Véronique Le Lann, qui sur le plan personnel m'a permis d'établir une relation de confiance, partage et amitié et qui, du côté professionnel, m'a encadré avec patience, rigueur et compétence. Je la remercie pour ses inépuisables disponibilités, critique, soutien et encouragement.

Je voudrais étendre cette gratitude à Monsieur Gilles Hétreux à qui je dois également la réalisation et la finalisation de mes travaux.

Je suis très reconnaissant envers Madame Claudia Frydman, Professeur à l'Université d'Aix Marseille 3, d'avoir accepté avec Monsieur Hervé Gueguen, Professeur à Supelec de Rennes, d'étudier mes travaux et d'en être les rapporteurs ainsi que pour l'intérêt et l'attention qu'ils ont accordés à cette étude.

Je remercie énormément les membres du jury pour les conseils et remarques qui m'ont beaucoup aidé à la finalisation de ce document :

Jean-Marc Le Lann, président du jury, Professeur à l'INP de Toulouse.

Naly Rakoto, Maître-Assistant à l'école des mines Nantes.

Je voudrais aussi inscrire dans ces lignes le plaisir que j'ai eu à travailler parmi vous, au LAAS. Avoir un quotidien énormément convivial avec M. Sahraoui que je remercie pour ces conseils et orientations, je remercie aussi Hamid, Nabil, Mourad, Rabah, Lotfi, Adel et Hcen pour leurs soutiens et leurs disponibilités. J'y ai passé d'inoubliables petits moments de discussion sur les problèmes du monde et de réflexion sur la vie avec Toufik, Hervé, Carine, Mahdi, les doctorants du groupe DISCO, les gens du foot, et d'autres personnes

involontairement oubliées. J'ai rencontré des personnes très sympathiques dans les couloirs, été accueilli par un personnel administratif et technique sympathique et patient.

Enfin, il y a des choses que ne peuvent pas être remerciées par écrit, néanmoins je voudrais remercier ma famille des deux rives, ma grand-mère, Ghenima, Gérard, Malika, Rabah, Dada et Khalti Drifa, Toufik pour leur accueil et leur soutien tout le long de ma thèse.

A mes chers parents, mes sœurs Nassima et Sonia, mon frère Mehena et sa femme Samia, pour leur soutien et leur amour : là où je serai je sais que ma famille sera toujours près de moi.

A ma chère femme Djaouida merci d'avoir été toujours à mes côtés -malgré la distance- dans cette aventure, de croire en moi plus que personne d'autre, de ne pas m'avoir laissé baisser les bras dans les moments les plus difficiles et bien sûr d'avoir partagé aussi les moments les plus heureux.

Résumé

Ce travail s'inscrit dans le domaine du diagnostic des systèmes hybrides et est basé sur l'utilisation d'un modèle. Il a pour objectif de diagnostiquer les fautes à partir de la connaissance structurelle, comportementale ou fonctionnelle du système en représentant ces connaissances (modèle du système) séparément de la connaissance sur la tâche de diagnostic.

Les systèmes hybrides incluent à la fois des variables continues et discrètes. La dynamique continue est généralement fournie par des équations différentielles et algébriques alors que la partie discrète est modélisée par des automates ou des systèmes à transition. Le formalisme adopté dans ce travail pour modéliser ces systèmes s'appuie sur le modèle « Réseau de Petri Différentiel à Objet » (RdPDO) qui est intégré dans la plate forme de simulation PrODHyS (*Process Object Dynamic Hybrid Simulator*). Il possède l'avantage de prendre en compte le comportement hybride d'une part, en associant les variables continues aux jetons et d'autre part, en associant un système algébro-différentiel aux places permettant de faire évoluer les variables d'état continues.

La méthodologie de diagnostic proposée s'effectue en deux étapes. La première étape consiste à détecter à l'aide d'une classification floue, des fautes qui présentent les mêmes symptômes - à partir d'une connaissance préalable des états de défaillance obtenue par apprentissage - afin de réduire les chemins ou les scénarios à explorer lors de la seconde phase. Cette dernière sert à diagnostiquer la faute parmi celles détectées à l'étape précédente en levant l'ambiguïté. Pour ce faire, deux raisonnements ont été suivis. Le premier, qualifié de raisonnement avant, consiste à former pour chaque faute incriminée, un critère d'écarts entre les mesures effectuées sur le système et celles émanant du modèle avec la faute simulée, sur une fenêtre temporelle et d'isoler ainsi la faute aboutissant au critère le plus faible. Le second raisonnement qualifié de raisonnement arrière, effectue des calculs similaires mais sur l'évolution temporelle passée du système par une simulation arrière effectuée avec PrODHyS, offrant la possibilité supplémentaire par rapport au premier raisonnement de remonter à l'instant de la défaillance. La méthodologie développée est illustrée sur un système hydraulique souvent utilisé comme « benchmark ». Comme nous ne disposons pas d'un système réel, celui-ci est simulé à l'aide d'un modèle de simulation de type RdPDO qui cette fois-ci contient les états de défaillances et des différences (bruits, erreurs de modélisation) par rapport au modèle utilisé pour le diagnostic.

Mots clés : Simulation dynamique, systèmes hybrides, diagnostic, réseaux de Petri, classification floue

Table des matière

Résumé	2
Table des matière	3
Table des figures	7
Introduction générale.....	10
Chapitre 1 Diagnostic : méthodes et modèles	13
1 Les approches relationnelles	14
2 Les méthodes de traitement de données.....	16
2.1 Le diagnostic par Analyse Qualitative de Tendence(AQT).....	16
2.2 Analyse en Composantes Principales (ACP)	17
2.3 Le diagnostic par reconnaissance de formes	18
2.3.1 Le principe de base.....	18
2.4 Conclusions sur les méthodes de traitement de données.....	19
3 Les méthodes à base de modèles.....	20
3.1 Approches basées sur la cohérence	20
3.1.1 Principe de base.....	20
3.1.2 Utilisation d'un modèle de dysfonctionnement	21
3.2 Les approches à base de modèles quantitatifs (FDI).....	22
3.2.1 Les méthodes basées sur l'approche mono-signal	22
3.2.2 L'approche multi-signaux	23
3.3 Les méthodes qualitatives basées sur un raisonnement causal	25
4 Conclusion.....	27
Chapitre2 : Caractérisation et simulation des.....	28
systèmes dynamiques hybrides	28
1 Introduction	28
2 Systèmes dynamiques hybrides.....	29
2.1 Extension des modèles à dynamique continue.....	30
2.2 Extension des modèles à événements discrets	31

2.2.1	Les automates temporisés.....	31
2.2.2	Les réseaux de Petri temporisés, temporels et stochastiques	32
2.2.3	Modèles mixtes	37
2.2.4	Conclusion.....	39
3	Simulation des systèmes dynamiques hybrides	40
3.1	Introduction	40
3.2	Pourquoi simuler?	40
3.3	Validation ou vérification fonctionnelle	41
3.4	Architecture générale d'un simulateur dynamique hybride	42
3.4.1	Simulation de la partie discrète	43
3.4.2	Simulation de la partie continue.....	43
3.4.3	Mise en place du modèle continu global.....	43
3.5	Classification des outils de simulation.....	43
4	CONCLUSION	45
Chapitre 3 : La plate forme de simulation PrODHyS		46
1	Introduction	46
2	Processus de développement logiciel.....	46
3	Architecture globale de la plate-forme <i>PrODHyS</i>	48
4	La couche modélisation.....	50
4.1	Modélisation de la matière	50
4.2	Modélisation des appareils	51
4.2.1	Ports définis dans <i>PrODHyS</i>	51
4.2.2	Les appareils élémentaires	51
4.2.3	Les appareils composés	52
4.2.4	Les appareils spécifique	52
5	La couche simulation	53
5.1	Le noyau numérique.....	53
5.1.1	Structure d'un modèle en dynamique	53
5.1.2	Résolution.....	54
5.2	Le module hybride	55
5.2.1	Sémantique et règles d'évolution du modèle RdPDO.....	55
5.3	LE GESTIONNAIRE DE SIMULATION	60
6	Conclusion.....	61

Chapitre 4 Méthodologie pour le diagnostic	62
Introduction	62
1 Schéma et méthodologie du diagnostic	63
1.1 Méthode de diagnostic par raisonnement avant	63
1.1.1 Modélisation de la recette du système en mode dégradé	64
1.1.2 Notion de scénario redouté.....	67
1.1.3 Travaux au LAAS sur les scénarii redoutés	69
1.1.4 Diagnostic par reconnaissance de formes	70
1.1.5 La méthode de classification <i>LAMDA</i>	73
1.2 Méthode de diagnostic par raisonnement arrière	81
1.2.1 Réseau de Petri inversé	82
1.2.2 Accessibilité entre deux marquages : deux approches duales pour les systèmes à événements discrets	82
1.2.3 Extention par PrOHdYS du raisonnement arrière aux systèmes hybrides.....	84
2 Conclusion.....	87
Chapitre 5 Application à un système hydraulique	89
1 Description du procédé considéré	89
1.1 Schémas d'instrumentation	90
1.2 Conditions initiales.....	91
1.3 Les défaillances considérées	92
1.4 Les équations dynamiques	92
2 Processus de modélisation du procédé	94
2.1 Modélisation du procédé	95
2.2 Modélisation de la recette	95
2.2.1 Génération des alimentations et soutirages	97
2.2.2 Sécurité contre le débordement de la cuve 1	97
2.2.3 Loi de commande $h_{2min} < h_2 < h_{2max}$	98
2.2.4 Éviter l'assèchement dans la cuve C2	99
2.3 Simulation	99
2.3.1 Simulation de fautes	100
3 La classification.....	103
3.1 L'outil SALSA	103
3.2 Apprentissage hors ligne	104

3.3	Diagnostic en ligne.....	106
3.3.1	Diagnostic des différentes fautes constituant la classe 2.....	108
3.3.2	Diagnostic des différentes fautes constituant la classe 3.....	117
4	Conclusion.....	120
	Conclusion générale	121
	Bibliographie.....	124

Table des figures

Figure 1.1	Une classification générale des approches de diagnostic.....	14
Figure 1.2	L'approche d'AQT pour le diagnostic de défaut	17
Figure 1.3	Diagnostic d'un système avec l'approche par reconnaissance de formes.....	19
Figure 1.4	Diagnostic à base de cohérence.....	21
Figure 1.5	Architecture générale de la détection de défauts à base de modèles.....	24
Figure 2.1	Systèmes hybrides : interaction du continu et du discret	29
Figure 2.2	Exemple illustratif : transfert et mélange des produits A et B	31
Figure 2.3	Automate temporisé simplifié	32
Figure 2.4	Réseau de Petri t-temporisé.....	32
Figure 2.5	Système de 2 Réservoirs	34
Figure 2.6	Réseau de Petri de l'exemple des 2 réservoirs	36
Figure 2.7	Architecture générale d'un simulateur dynamique hybride	42
Figure 2.8	Outils de simulation	44
Figure 3.1	Conception itérative	47
Figure 3.2	Structure logicielle de la plate-forme PrODHyS.....	48
Figure 3.3	Création d'un univers thermodynamique	50
Figure 3.4	Représentation des ports définis dans PrODHyS.....	51
Figure 3.5	Ports d'un appareil composé	52
Figure 3.6	Sémantique du modèle	56
Figure 3.7	Sensibilisation des arcs « classiques » et des arcs inhibiteurs.....	57
Figure 3.8	Représentation des places discrètes et différentielles	58
Figure 3.9	Modélisation des places temporisées selon le formalisme RDPDO	59
Figure 3.10	Noyau de simulation.....	60
Figure 4.1	Schéma général de la méthode de diagnostic par raisonnement Avant	64
Figure 4.2	Système de régulation de deux réservoirs	65

Figure 4.3	RdP du fonctionnement normal du système.....	66
Figure 4.4	RdP du modèle avec dysfonctionnement	67
Figure 4.5	Modèle de réseau de Petri de la constellation	68
Figure 4.6	Arbre de défaillance de la constellation	68
Figure 4.7	La forme observée X est ici associée à la classe C_5	70
Figure 4.8	Structure d'un système de diagnostic par reconnaissance des formes	71
Figure 4.9	Structure générale du classificateur.....	72
Figure 4.10	Schéma général du calcul de l'adéquation d'un objet à une classe.....	76
Figure 4.11	Algorithme général de LAMDA	79
Figure 4.12	Méthode de diagnostic par raisonnement arrière	81
Figure 4.13	Exemple de réseau de Petri marqué	83
Figure 4.14	Réseau de Petri inverse	85
Figure 4.15	Résultats de la simulation arrière	85
Figure 4.16	Présentation de l'exemple de la cuve	86
Figure 5.1	Le système hydraulique à deux réservoirs	90
Figure 5.2	Les quatre cas des conditions initiales	91
Figure 5.3	Débit traversant les deux réservoirs	92
Figure 5.4	Débit dans un seul réservoir	93
Figure 5.5	Exemple de configuration	93
Figure 5.6	Modèle associé à l'exemple d'application	95
Figure 5.7	Recette associée au système	96
Figure 5.8	Génération des alimentations et soutirage.....	97
Figure 5.9	RdP de sécurité contre le débordement de la cuve 1	98
Figure 5.10	Loi de commande $h_{2min} < h_2 < h_{2max}$	98
Figure 5.11	Rdp pour éviter l'assèchement dans la cuve 2	99
Figure 5.12	Niveaux du liquide dans le procédé en fonctionnement normal	99
Figure 5.13	Simulation de la pompe P1 bloquée en ouverture.....	100
Figure 5.14	Simulation de la pompe P2 bloquée en ouverture.....	101
Figure 5.15	Simulation de la pompe P2 bloquée fermée.....	102
Figure 5.16	Simulation de la vanne V2 bloquée ouverte.....	102
Figure 5.17	Simulation de la vanne V2 bloqué fermé	103
Figure 5.18	Structure de SALSA hors ligne.....	104
Figure 5.19	Structure de SALSA en ligne	104

Figure 5.20	Classification des fautes	105
Figure 5.21	Nombre de classes selon les fonctions d'appartenance.....	106
Figure 5.22	Schéma du diagnostic en ligne du procédé	106
Figure 5.23	Méthodologie de Diagnostic en ligne.....	107
Figure 5.24	Simulation de la faute P1BO avec erreur sur le modèle	108
Figure 5.25	Reconnaissance de classe pour la faute P1BO avec erreur sur le modèle du procédé	109
Figure 5.26	Calcul de l'écart avec 5% d'erreur sur le modèle du procédé.....	110
Figure 5.27	Simulation de la faute P2BF avec une erreur sur le modèle du procédé.....	110
Figure 5.28	Reconnaissance de classe pour la faute P2BF.....	111
Figure 5.29	Calcul de l'écart avec 5% d'erreur sur le modèle du procédé.....	111
Figure 5.30	Simulation de la faute V2BO avec une erreur de 5% sur le modèle du procédé	112
Figure 5.31	Reconnaissance de classe dans le cas de la faute V2B	112
Figure 5.32	Calcul de l'écart avec 5% d'erreur sur le modèle du procédé.....	113
Figure 5.33	Simulation de la faute P1BO avec 5% d'erreur de mesure	113
Figure 5.34	Simulation de la faute P2BF avec 5% d'erreur de mesure.....	114
Figure 5.35	Simulation de la faute V2BO avec 5% d'erreur de mesure	114
Figure 5.36	Reconnaissance de la faute P1BO avec 5% d'erreur de mesure	115
Figure 5.37	Reconnaissance de la faute P2BF avec 5% d'erreur de mesure.....	115
Figure 5.38	Reconnaissance de la faute V2BO avec 5% d'erreur de mesure	116
Figure 5.39	Les critères d'écart avec 5% d'erreur sur les mesures	116
Figure 5.40	Faute P2BO	117
Figure 5.41	Faute V2BF Phase de reconnaissance.....	117
Figure 5.42	Reconnaissance de la faute P2BO	118
Figure 5.43	Reconnaissance de la faute V2BF	118
Figure 5.44	Critère d'écart avec 5% erreur de mesure	119
Figure 5.45	Critère d'écart avec 5% d'erreur sur le procédé.....	119

Introduction générale

En raison d'une modernisation incessante des outils de production, les systèmes industriels deviennent de plus en plus complexes et sophistiqués. En parallèle, la fiabilité, disponibilité, sûreté de fonctionnement sans oublier la protection de l'environnement sont devenues de véritables enjeux pour les entreprises actuelles. Le diagnostic des systèmes est apparu dans le but d'améliorer les points précédents. Discipline de l'automatique à part entière, ce module de surveillance qu'est le diagnostic fait l'objet d'un engouement prononcé depuis le début des années 70. En effet, la recherche dans ce domaine n'a fait que prendre de l'importance dans le monde entier, aussi bien de manière théorique que pratique.

Dans ce contexte, de nombreuses approches sont développées, en vue de la détection de défaillances et du diagnostic, par les différentes communautés de recherche en automatique, productive et intelligence artificielle. Les méthodes se différencient par rapport au type de connaissance a priori sur le processus qu'elles nécessitent. Ainsi, elles peuvent être classées, de façon générale, comme des méthodes à base de modèles, à base de connaissances et des méthodes à base de données historiques. Les méthodes à base de modèles considèrent un modèle structurel du comportement du processus basé sur des principes physiques fondamentaux. Ces modèles peuvent être de type quantitatif, exprimés sous forme d'équations mathématiques ou bien de type qualitatif, exprimés par exemple sous forme de relations logiques. Les méthodes à base de connaissance exploitent les compétences, le raisonnement et les connaissances des experts sur le processus pour les transformer en règles, de manière à résoudre des problèmes spécifiques. Enfin, les méthodes à base de données cherchent à découvrir des informations, sous forme d'exemples type ou tendances, au sein des mesures venant des capteurs et des actionneurs, pouvant identifier le comportement du procédé. Ces méthodes comprennent, parmi d'autres, les méthodes d'apprentissage et de classification (ou reconnaissance de formes).

Sachant que nous ne disposons pas souvent d'un modèle de comportement réel, un travail de simulation s'impose. Au cours de ces vingt dernières années, les outils informatiques pour la modélisation et la simulation des procédés se sont développés conjointement avec les outils et techniques informatiques. La technologie des ordinateurs a considérablement évolué et les langages ont progressé, passant d'une approche procédurale à une approche orientée objet. Dans les années 90, les simulateurs dynamiques se sont améliorés en termes de structure et de fonctionnalité ; ils ont profité, notamment, d'avancées importantes dans la résolution de grands systèmes d'équations. Ainsi, la contribution de l'informatique dans le domaine du génie des procédés par exemple s'est largement accrue et a donné naissance à

l'ingénierie des procédés assistés par ordinateurs' (IPAO), facilitant ainsi considérablement la tâche des ingénieurs de procédé.

Avec l'objectif de suivre au mieux cette tendance, une plate-forme de *simulation dynamique hybride*, nommée *PrODHyS*, a été développée au sein du *Laboratoire de Génie Chimique*. Elle est issue de l'unification des travaux entrepris depuis plus de dix ans (Jourda, 1996; Moyse, 2000; Sargousse, 1999). Elle a aussi profité de la longue expérience acquise au sein de l'équipe *AFP (Analyse Fonctionnelle des Procédés)* dans le domaine de la modélisation et de la simulation des procédés et plus spécifiquement dans le traitement algébro-différentiel (travaux de P. Cuillé (1986), J Albet (1992), A. Sargousse (1999) et HDR de JM. Le Lann (1999)) et dans le domaine de la gestion et la supervision des processus discontinus (travaux de B. Daubas (1994), B. Palomino (2000) et HDR de H. Pingaud (1998)) ainsi que des travaux menés au *LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes)* sur la commande et la supervision des systèmes (travaux de D. Andreu (1996) et HDR de Je. Pascal(1999)).

Par ailleurs, Les travaux développés par le groupe DISCO (DIagnostic, Supervision et COnduite) au sein du LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) se situent dans ce cadre de la maîtrise des systèmes complexes. Les thèmes de recherche abordés couvrent les aspects : commande (régulation, poursuite) supervision, surveillance, diagnostic, aide à la conduite et interaction homme-système. Les travaux développés s'appuient sur trois types d'approche que sont les approches à base de modèles, à base de connaissances et par apprentissage et reconnaissance de formes.

Notre travail s'inscrit dans le domaine du diagnostic à base de modèle en associant la simulation dynamique à la classification. La présentation du manuscrit s'articule autour de cinq chapitres.

Dans le premier chapitre, nous analysons les principales approches de diagnostic liées à notre problématique de diagnostic et de modélisation que nous avons organisées selon la classification suivante : les méthodes relationnelles, les méthodes de traitement de données et les méthodes à base de modèles.

Dans le second chapitre, nous dressons un état des lieux des modèles et outils existants dans le domaine des systèmes dynamiques hybrides, ainsi que les notions de simulation de ces derniers.

Le troisième chapitre est consacré à la présentation de la plate-forme de simulation *PrODHyS* dans son architecture globale en spécifiant ses deux couches de modélisation et simulation et en détaillant le modèle réseau de Petri différentiel à objet (RDPDO) utilisé pour la modélisation.

La méthodologie de diagnostic que nous avons développée s'articule autour de la mise de collaboration de deux approches : la classification et la simulation dynamique ; elle est présentée dans le chapitre quatre. Il explique les deux possibilités de raisonnement le raisonnement avant et le raisonnement arrière et revient plus en détail sur les méthodes de classification et de traitement de données utilisées pour le diagnostic.

Un exemple d'application permet d'illustrer la méthodologie proposée est détaillé dans le chapitre 5. Après la présentation du système hydraulique considéré, nous développerons le réseau de Petri associé au modèle pour ensuite détailler la phase de diagnostic pour différents

cas de fautes en nous attachant tout particulièrement à des fautes pouvant conduire à des symptômes identiques afin de montrer l'apport de cette méthodologie. Nous terminerons bien sur par une conclusion générale et des perspectives à ce travail.

Chapitre 1 Diagnostic : méthodes et modèles

Ce chapitre est consacré à la présentation des concepts fondamentaux des approches de diagnostic issues des communautés FDI, SED, SDH et IA. Elle s'appuie sur la classification des approches selon le type de méthode ou de modèle utilisés. Dans la plupart des cas, les méthodes de diagnostic sont liées à la connaissance disponible sur le procédé et à sa représentation et sont classées de différentes façons par de nombreux auteurs [Frank & Köppen-Seliger, 97], [Isermann, 97], [Travé-Massuyès et al., 97], [Chantler et al., 98], [Gentil, 03], [Venkatasubramanian et al., 03a]. La terminologie et la classification ne sont pas toujours homogènes, influencées par les contextes et les terminologies particulières à chaque communauté et domaine d'application. De plus, très souvent les méthodes et les modèles sont intimement liés.

A partir de ces considérations, nous proposons une classification non exhaustive des méthodes de diagnostic selon trois axes : les approches relationnelles, les méthodes de traitement de données (méthodes qualitatives et méthodes quantitatives) et les approches à base de modèles. Pour ce dernier, nous avons fait apparaître deux branches concernant les méthodes quantitatives (communauté FDI) et les méthodes qualitatives (communauté IA) liées au domaine du continu et une branche spécifique concernant les méthodes discrètes. Cette organisation est présentée sur la figure 1.1 où apparaissent également les modèles liés aux méthodes.

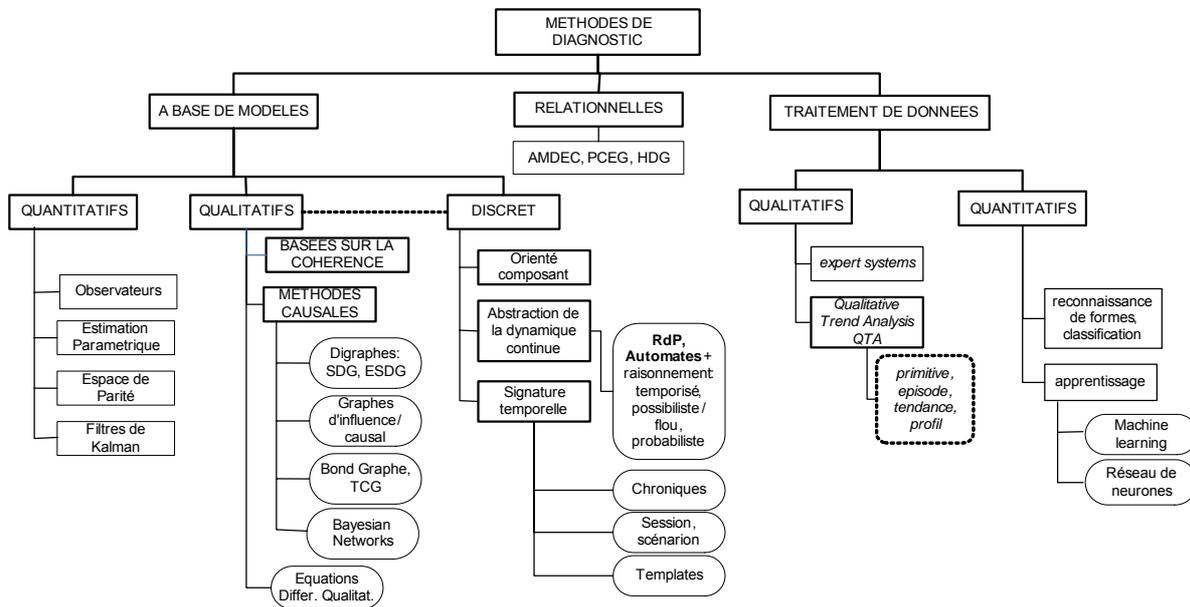


Figure 1.1 Une classification générale des approches de diagnostic

1 Les approches relationnelles

Les *approches relationnelles* sont des méthodes issues du contexte de la sûreté de fonctionnement qui associe des causes à des symptômes. Elles sont basées, en général, sur des systèmes à base de règles, de dictionnaires de défauts ou de graphes avec un raisonnement de parcours ou adductif, sur la méthode AMDEC (*Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités*), sur les graphes PCEG (*Possible Cause and effect graph*), HDG (*Hazop-digraph*) ou sur les réseaux Bayesiens. Ce sont des approches de diagnostic fondées sur une connaissance associative dépendante du système et sur une connaissance a priori des défauts et de leurs effets. Elles manquent d'un pouvoir de réutilisabilité et nécessitent une énumération de tous les défauts possibles. Dans le cas des procédés complexes qui nous préoccupent, le nombre élevé de variables, de composants et de modes opératoires, rend leur utilisation peu adaptée.

Nous présentons ces méthodes, issues des études de risques et de sûreté de fonctionnement, qui peuvent être utilisées pour identifier les causes des défaillances de processus industriels. Il s'agit, en fait, de l'analyse fonctionnelle et matérielle du système.

*AMDE(Analyse des Modes de Défaillance et de leurs Effets). La méthode de l'Analyse des Modes de Défaillance et de leurs Effets fût employée pour la première fois, à partir des années 1960, dans le domaine de l'aéronautique pour l'analyse de la sécurité des avions [Rechet, 66]. Cette méthode permet une analyse systématique et très complète,

composant par composant, de tous les modes de défaillance possibles et précise leurs effets sur le système global [Villemeur, 88].

La démarche consiste d'abord à définir le système, ses fonctions et ses composants. Ensuite, l'ensemble des modes de défaillances des composants doit être établi. Pour chaque mode de défaillance, sont recherchées ensuite les causes possibles de son apparition. Finalement, une étude des effets sur le système et sur l'opérateur est faite pour chaque combinaison (cause, mode de défaillance). Les résultats sont présentés sous forme de tableau. L'AMDE est parfois complétée par une analyse de criticité ; elle devient AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité). La criticité permet d'extraire les modes de défaillance les plus critiques. L'analyse de criticité est évaluée à partir du couple probabilité-gravité.

L'utilisation des tableaux d'AMDE(C) à des fins de diagnostic industriel conduit à utiliser une procédure déductive, c'est-à-dire à utiliser ces tableaux comme un outil d'identification des causes de défaillances à partir des effets observés [Zwingelstein, 95]. La modélisation des relations de cause à effet, réalisée par l'AMDEC rend cette démarche très puissante pour la résolution des problèmes de diagnostic de défaillances de procédés industriels.

L'AMDE est donc très répandue dans des nombreux domaines industriels, l'aéronautique en particulier, cependant elle est lourde et insuffisante. En effet, il est nécessaire d'identifier a priori les défauts et/ou les dysfonctionnements pouvant apparaître, à un certain moment, dans le système supervisé ; le recensement préalable des défaillances et leurs relations éventuelles ne peut jamais être exhaustif et requiert en général une longue expérience. En plus, toute modification ou évolution du système nécessite une réécriture du tableau. Enfin, cette méthode ne peut traiter les cas de défaillances multiples et intégrer l'aspect fonctionnel.

**L'Analyse de fonctionnement (HAZOP, Hazard and operability study)* est une autre extension de l'analyse des modes de défaillance (AMDE). La méthode HAZOP explore systématiquement l'aspect fonctionnel d'un système en identifiant, à la fois, les dangers et les dysfonctionnements d'une installation. Elle est mise en œuvre à la fin de la phase de conception puisqu'elle s'appuie sur les schémas détaillés du système.

**Arbres de Défaillances (ou Arbres de Causes)* L'arbre de défaillance est l'un des outils majeurs d'analyse du risque technologique, développé au début des années 1960 aux Etats-Unis [Villemeur, 88]. C'est la méthode la plus couramment utilisée dans les analyses de fiabilité, de disponibilité ou de sécurité des systèmes. Il s'agit d'une méthode optimisée qui permet la détermination des chemins critiques dans un système. Cette méthode a pour objet de déterminer les diverses combinaisons possibles d'événements qui entraînent la réalisation d'un événement indésirable unique. Elle permet donc d'identifier les points faibles de la conception. C'est une méthode déductive dont la représentation graphique des combinaisons est réalisée par une structure arborescente (arbre), permettant un traitement à la fois qualitatif et quantitatif. Cet arbre est établi sous forme d'un diagramme logique et comporte au sommet l'événement indésirable. Les causes immédiates qui produisent cet événement sont ensuite hiérarchisées à l'aide de symboles logiques "ET" et "OU". De cette façon, l'arbre est créé pas à pas pour atteindre, à la base, un ensemble d'événements considérés comme élémentaires. Une extension des arbres de défaillances est la méthode du diagramme de causes-

conséquences [Villemeur, 88]. Cette méthode combine les principes utilisés par l'analyse déductive de l'arbre des défaillances et l'analyse inductive de l'arbre des conséquences. Le principal inconvénient des arbres de défaillances est que le développement est sensible aux erreurs commises aux différentes étapes. En effet, l'arbre construit est seulement aussi bon que le modèle mental de son créateur. Pour exécuter un diagnostic correct à partir des arbres de défaillances, ceux-ci doivent largement représenter toutes les relations causales du processus, c'est-à-dire qu'ils doivent être capables d'expliquer tous les scénarios de défauts possibles. De plus, l'emploi de cette méthode se révèle difficile pour les systèmes fortement dépendants du temps. Enfin, il n'y a pas de méthode formelle pour vérifier l'exactitude de l'arbre développé.

2 Les méthodes de traitement de données

Les méthodes de traitement de données consistent dans l'exploitation des observations (le plus souvent quantitatives mais aussi parfois qualitatives). La seule information disponible, sur le procédé est sous la forme de données historiques ou de résultats de traitement en ligne des signaux issus de capteurs.

2.1 Le diagnostic par Analyse Qualitative de Tendances(AQT)

La représentation et l'analyse qualitative (symbolique) de signaux à travers l'information des dérivés de premier et deuxième ordre représentent l'une de principales techniques pour le diagnostic et la supervision basées sur des données et sur l'abstraction de la dynamique continue. L'*analyse qualitative de tendance* des signaux est basée, en général, sur la segmentation temporelle des signaux en unités, appelées épisodes, en fonction de leur pente et de leur convexité et d'une classification de ces épisodes. [Meléndez & Colomer, 01] énumèrent les besoins des systèmes de supervision qui motivent l'utilisation des méthodes d'analyse qualitative de tendances :

- L'incertitude, l'incomplétude et l'hétérogénéité (différentes échelles de temps par exemple) des données demandent un raisonnement qualitatif.
- Un raisonnement temporel est nécessaire non seulement sur des informations instantanées mais aussi sur le comportement historique.
- Face au volume de données présent dans les procédés complexes, il est souhaitable de posséder une représentation symbolique et adaptée au système de surveillance.
- La représentation de la connaissance doit être adaptée à l'interprétation de l'opérateur humain.

Relativement simple conceptuellement, l'AQT se trouve alourdie par certaines contraintes. L'observation des dérivées des signaux s'avère délicate. L'AQT implique la mise en œuvre d'un étage de filtrage de bruits et de traitement des discontinuités ou même la suppression de l'information d'ordre supérieur (dérivées). Pour le suivi et le diagnostic, le contexte temporel des tendances est fondamental. Le synchronisme ou l'alignement temporel entre le dictionnaire (signature) de tendances et les observations n'est pas facile à obtenir et des écarts faibles peuvent induire des erreurs d'interprétation et d'inférence au niveau du diagnostic.

D'autres travaux fondés sur l'analyse qualitative de tendances (AQT) sont proposés pour le suivi, le diagnostic et la supervision des procédés complexes. La classification de tendances pour le diagnostic de procédés chimiques à travers une inférence floue est proposée par [Dash et al., 03]. Une matrice de similarité des primitives quantifie la mesure de distance (indices de similarité) (SI_j) entre l'observation et la signature à travers une inférence floue qui permet une robustesse de l'identification de tendance. SI_j considère non seulement la séquence des primitives mais aussi l'intervalle de temps ($\Delta t_i = t_{ui+1} - t_{ui}$). Un module de connaissance à base de règles s'appuie sur les indices des variables pour raisonner sur les hypothèses des défauts et les classer en fonction de l'indice de confiance globale (CI_i). La figure 1.2 montre le schéma de l'approche.

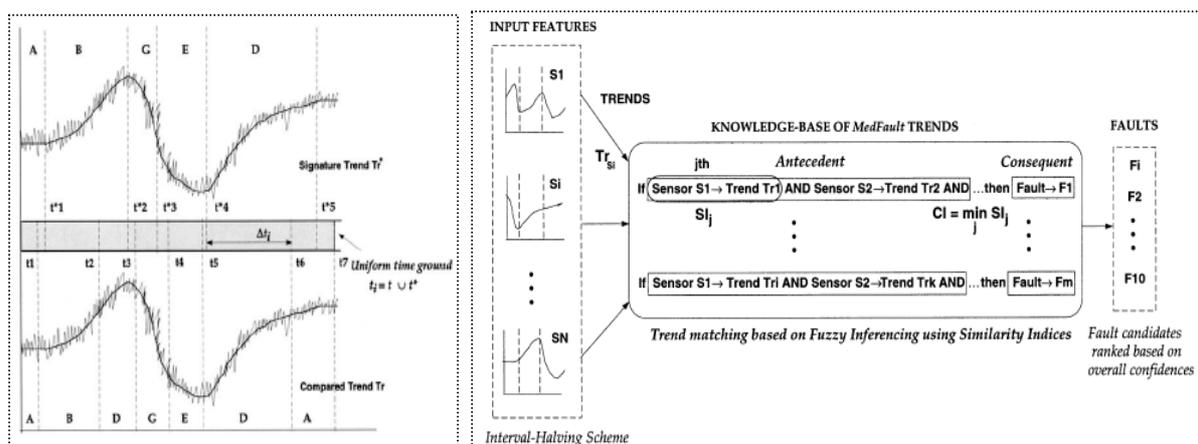


Figure 1.2 L'approche d'AQT pour le diagnostic de défaut

D'une manière générale, les approches d'AQT présentent une contrainte importante pour le diagnostic. L'information individuelle de chaque variable, sans aucune vérification des influences entre elles, exige une description exhaustive des modes du procédé, et des comportements de défauts $f_i \in F$. La discriminabilité de défauts dépend alors d'un grand volume de données. L'information sur la dérivée de premier ordre de la trajectoire de comportement normal ne subissant pas un changement abrupt, l'approche s'avère inappropriée dans le cas de dérives lentes. Le mécanisme d'inférence floue de [Dash et al., 03] ne distingue pas les écarts marginaux observés entre le dictionnaire et les observations. Il nous semble cependant que la démarche pourrait être étendue vers une solution permettant la discrimination de la nature des écarts.

2.2 Analyse en Composantes Principales (ACP)

L'analyse en composantes principales (ACP) est une technique statistique multivariable. Les techniques statistiques multi-variables sont des outils puissants, capables de compresser des données et de réduire leur dimensionnalité de sorte que l'information essentielle soit maintenue et plus facile à analyser que dans l'ensemble original des données. Ces techniques peuvent également manipuler le bruit et la corrélation pour extraire efficacement l'information. La fonction principale de ce type de techniques est, par le biais

d'une procédure mathématique, de transformer un certain nombre de variables corrélées en un ensemble plus petit de variables non corrélées.

L'ACP est essentiellement basée sur une décomposition orthogonale de la matrice de covariance des variables du processus le long des directions qui expliquent la variation maximale des données, c'est-à-dire que cette méthode recherche une projection des observations sur des axes orthogonaux. De ce fait, le premier axe contient la plus grande variation. Le deuxième axe contiendra la seconde plus grande variation orthogonale au premier.

Le but principal de l'ACP est donc de trouver un ensemble de facteurs (composantes) qui ait une dimension inférieure à celle de l'ensemble original de données et qui puisse décrire correctement les tendances principales. L'ACP est une procédure qui ne prend en compte que les variables du processus. Parfois, un ensemble additionnel de données est disponible, e.g. variables de qualité du produit. Il est souhaitable d'inclure toutes les données disponibles pour la surveillance du procédé et d'utiliser de cette façon les variables du processus pour prédire et détecter des changements dans les variables de qualité du produit. Pour cela, la méthode des *Moindres Carrés Partiels (MCP)* peut être utilisée [Venkatasubramanian et al., 03b]. Cette méthode modélise le rapport entre deux blocs de données tout en les comprimant simultanément. Elle est employée pour extraire les variables latentes qui expliquent la variation des données du processus.

Divers travaux menés par MacGregor et al [MacGregor, 95] [Nomikos, 94] ont montré l'application des méthodes statistiques multi-variables telles que les ACP et MCP, dans l'analyse et la commande de processus, la détection et le diagnostic de défaillances, aussi bien dans le cas de procédés continus que dans les cas des procédés "*batch*" (continus par lots). Une limitation importante de la surveillance basée sur l'ACP est que la représentation obtenue est invariante dans le temps, tandis que la plupart des processus réels évoluent au cours du temps. Par conséquent, la représentation issue de l'ACP nécessite également d'être mise à jour périodiquement. Un autre inconvénient est qu'elle ne possède pas des propriétés de *signature* pour le diagnostic, ce qui rend l'isolation des défaillances difficile.

2.3 Le diagnostic par reconnaissance de formes

2.3.1 Le principe de base

Un problème de diagnostic peut se définir comme un problème de reconnaissance de formes. Il a pour but la reconnaissance d'une forme parmi différentes possibilités à partir d'observations bruitées. Il existe plusieurs approches : la reconnaissance de forme *structurelle ou syntaxique* qui exploite les relations entre les composants de la forme et la reconnaissance de formes de *type numérique* (statistique, floue, etc) qui exploite des modélisations de formes probabilistes ou floues [Denoeux, 01].

Une *forme* est représentée par un ensemble de paramètres ou *attributs* caractéristiques qui peuvent être numériques ou symboliques. Un *prototype* est défini par des valeurs précises de l'ensemble des paramètres caractéristiques d'une situation particulière (défaillance dans notre contexte). Une classe est définie par un ensemble possible de valeurs des attributs et est représentée par son prototype. La classification d'un objet (la reconnaissance d'une forme) parmi M classes est la décision d'affecter l'objet à une classe particulière. Les classes peuvent

être définies de façon *probabiliste* ou encore comme des *classes floues*. Dans le cadre du diagnostic, le prototype 1 peut être lié au fonctionnement normal du procédé alors que les prototypes 2 et 3 peuvent représenter deux modes de défauts distincts. Le diagnostic consiste à décider de quel prototype cette forme est la plus proche en fonction d'un critère d'évaluation comme les distances euclidiennes (classes circulaires) ou les distances Mahalanobis (classes elliptiques) par exemple. Dans [Dubuisson, 01], les approches probabilistes et non probabilistes (*flou* et *crédibiliste*) sont développées en détail. La figure 1.3 nous résume le principe du diagnostic par reconnaissance de formes.

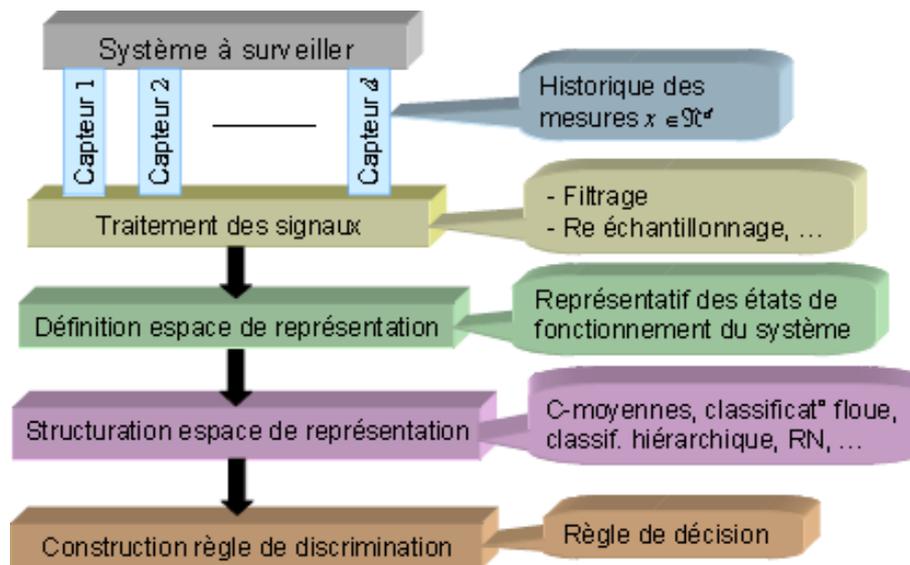


Figure 1.3 Diagnostic d'un système avec l'approche par reconnaissance de formes

2.4 Conclusions sur les méthodes de traitement de données

Les méthodes de traitement de données exploitent des observations quantitatives et/ou qualitatives disponibles sous la forme de données historiques ou de résultats de traitement en ligne de signaux issus des capteurs. Ce sont des approches envisageables quand l'obtention d'un modèle analytique du procédé s'avère difficile, et lorsqu'un raisonnement sur les comportements dynamiques (variables et relations) du procédé n'est pas utile. Le comportement de référence d'un signal, en général *statistique*, peut être représentatif de l'état normal de l'installation ou d'un défaut particulier. Les approches de classification de données (*reconnaissance de formes*) sont fondées sur l'analyse des données issues des signaux corrélés entre eux pour la discrimination des différents modes de fonctionnement et certains modes de défaut. Toutes ces approches sont fortement dépendantes d'un grand volume des données, ce qui les limite en général à la détection. Le diagnostic dépend donc de la représentation et de la discrimination de tous les modes de défaut. Dans le cadre du suivi de régions transitoires, les méthodes d'AQT sont les plus utilisées. Le diagnostic cependant est fondé sur un mécanisme d'inférence qui dépend d'une connaissance assez large des modes de défaut et de la prise en compte des techniques d'alignement temporel.

3 Les méthodes à base de modèles

Les *approches à base de modèles* s'appuient sur des modèles comportementaux explicites du système soumis au diagnostic. Un grand avantage de ces approches par rapport aux approches relationnelles et de traitement de données, réside sur le fait que seule l'information du comportement normal du procédé est prise en compte par l'intermédiaire d'un modèle de référence. La précision du modèle, liée aux besoins de la surveillance et aux critères de performance du diagnostic, définit le choix de l'utilisation de modèles quantitatifs, qualitatifs ou semi-qualitatifs. Selon [Travé-Massuyès et al., 97], les méthodes de diagnostic à base de modèles présentent les avantages suivants :

- La connaissance sur le système est découplée de la connaissance de diagnostic
- Il s'agit de connaissance de conception plutôt que d'exploitation
- Les fautes et les symptômes ne doivent pas être anticipés
- Le coût de développement et de maintenance est moindre
- Les modèles fournissent un support adéquat pour l'explication (structure du système explicitement représentée).

Nous trouvons souvent dans la littérature, une classification des approches de diagnostic à base de modèles selon trois branches principales:

- Les approches issues de la *communauté IA* utilisent une connaissance profonde orientée composant (équipement) basée sur une théorie logique de raisonnement [Reiter, 87]. L'analyse fondamentale envisage l'obtention de cohérence entre les observations et le modèle à travers la suppression des hypothèses sur le comportement de quelques composants [Dague, 01]. La *modélisation et le raisonnement qualitatif* et la *modélisation causale* sont les deux tendances principales proposées par cette communauté.
- Les approches issues de la *communauté FDI* sont fondées sur la modélisation et le contrôle des systèmes industriels, avec, contrairement aux approches IA, *un modèle dynamique quantitatif* généralement représenté par des équations différentielles ou à différences avec une précision définie par l'objectif du diagnostic.
- Les approches issues de la *communauté SED* sont basées sur une représentation purement discrète du procédé [Lunze, 00] due à la nature discrète des variables de ses composants (p.ex. vanne tout ou rien). Elles peuvent être également basées sur l'abstraction des dynamiques continues par des états finis qualitatifs (discrets) pour la description qualitative de l'espace d'état continu global du sous-système ou pour son association (états qualitatifs) à des états fonctionnels d'un composant (orientation composant) [Lunze, 98], [Blanke et al., 03], [Fanni & Giua, 98], [Hélias et al., 04].. Le diagnostic est fondé sur la vérification des critères comme l'accessibilité d'état et la cohérence événementielle entre l'observation et le modèle. La reconnaissance de séquences d'événements pour l'identification de scénarios de défaut constitue également une autre branche.

3.1 Approches basées sur la cohérence

3.1.1 Principe de base

Reiter (1987) a proposé une théorie logique de diagnostic habituellement intitulée diagnostic issu des *principes premiers (diagnosis from the first principles)*. Le problème de

diagnostic est la détermination des composants du système dont le fonctionnement anormal peut expliquer l'incohérence entre les comportements prédit et observé (figure 1.4). Cette approche a été étendue et formalisée dans [Kleer et al., 92]. Le cadre théorique qui formalise plus en détail les concepts est développé dans un certain nombre de travaux [Travé-Massuyès et al., 97], [Cordier et al., 00], [Dague, 01], [Kleer, 03].

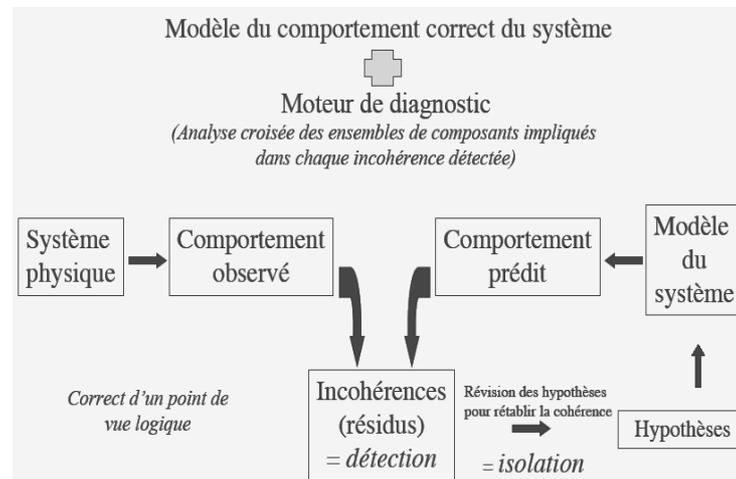


Figure 1.4 Diagnostic à base de cohérence

Le diagnostic est basé sur la notion de *conflit*. La détection de *conflits* consiste à vérifier les contradictions entre les observations et les prédictions effectuées par le modèle de référence, en ne se contentant pas d'indiquer la présence de défauts (*détection*) mais également en renseignant sur la *localisation* de ces défauts. Il s'agit, pour cela, d'utiliser les prédictions antérieurement enregistrées qui conduisent aux contradictions en question. Dans une approche basée sur des composants, si la prédiction, issue d'un ensemble de modèles de bon comportement des composants $COMPS = \{c_1, c_2, \dots, c_n\}$, entre en contradiction avec les observations, l'un des composants est nécessairement défaillant et les composants forment un *conflit*.

L'étape suivante est la *génération des hypothèses* qui rendent compte de tous les conflits, c'est-à-dire de toutes les contradictions détectées. Un ensemble de composants qui, cessant d'être supposés corrects, rétablit la cohérence avec les observations est précisément appelé un *diagnostic* (ou *diagnostic à base de cohérence*¹). Afin de faire disparaître chaque conflit, il faut nécessairement changer l'hypothèse de fonctionnement d'au moins un de ses éléments (de correct à incorrect).

3.1.2 Utilisation d'un modèle de dysfonctionnement

L'idée première du diagnostic à base de modèles est de se passer des connaissances sur les défauts ou les dysfonctionnements. Mais certaines connaissances de ce type, si elles sont disponibles, peuvent aider à la localisation des défauts et il serait dommage de s'en passer. De plus, il est généralement indispensable d'avoir de telles connaissances si l'on souhaite identifier les défauts après les avoir localisés. C'est pourquoi des extensions du

¹ de l'anglais : *consistency-based diagnosis*

formalisme ont été proposées, qui permettent d'exprimer des modèles de dysfonctionnement [Kleer et al., 92].

Il est important de remarquer que, contrairement aux approches traditionnelles, où les relations entre défauts et symptômes sont empiriques, spécifiques au système à diagnostiquer et ne peuvent en aucun cas garantir la validité logique du diagnostic établi, cette validité demeure garantie dans les extensions en question. Remarquons que l'on peut toujours introduire en plus, si besoin est, une telle connaissance empirique, mais uniquement cette fois au titre d'heuristiques aidant à parcourir éventuellement plus vite l'espace de recherche conduisant aux solutions mais sans influence sur celles-ci, qui reposent ici sur des principes logiques rigoureux. Au lieu de n'avoir comme précédemment que deux modes de comportement par composant, correct et incorrect, dont seul le premier est modélisé, s'ajouteront cette fois aux modes corrects plusieurs modes de dysfonctionnements (en général deux à deux exclusifs) modélisés. Mais pour tenir compte de l'impossibilité d'une énumération exhaustive de tous les défauts possibles, il sera toujours ajouté un mode inconnu dépourvu de tout modèle qui est censé regrouper tous les comportements défectueux non répertoriés.

3.2 Les approches à base de modèles quantitatifs (FDI)

3.2.1 Les méthodes basées sur l'approche mono-signal

3.2.1.1 Redondance matérielle

C'est une méthode employée dans des installations critiques (l'aérospatial, le nucléaire). L'utilisation de plusieurs capteurs en vue d'obtenir la même information sur une variable permet de détecter les déviations par rapport à un état normal et de localiser un défaut de capteur. Les inconvénients de cette méthode sont l'accroissement du coût de l'installation et l'augmentation de la probabilité de pannes de capteurs – donc d'un besoin de maintenance supplémentaire.

3.2.1.2 Analyse spectrale

Les signaux sont analysés en état normal de fonctionnement ; les hautes fréquences sont reliées au bruit et les basses fréquences aux évolutions propres de l'état du procédé. Ensuite, toute déviation des caractéristiques fréquentielles d'un signal est reliée à une situation de défaillance. Cette approche se révèle très utile pour analyser des signaux qui montrent des oscillations avec des périodes longues (les courants électriques, les débits, les pressions...). L'inconvénient est la sensibilité aux bruits de mesure quand ceux-ci coïncident avec la zone fréquentielle d'intérêt et la nécessité d'un échantillonnage fréquent pour permettre de reconstituer le signal de départ tout en minimisant la perte de fréquence. Les méthodes d'auto-corrélation, la densité spectrale des signaux, la transformée de Fourier, les ondelettes sont bien appropriées dans le cas où les fréquences représentatives de défauts sont connues. Dans le cas contraire, il est cependant préférable d'utiliser des modèles paramétriques des signaux qui permettent d'estimer en ligne les fréquences et les valeurs moyennes des paramètres.

3.2.1.3 Approches statistiques :

Ces approches se basent sur l'hypothèse de changements rapides (et non sur l'amplitude) des caractéristiques des signaux ou des paramètres des modèles par rapport à des dynamiques considérées comme étant lentes (procédés quasi stationnaires). Elles sont utilisées pour la détection de changements graduels avec des seuils de détection faibles. Les informations fournies par le nombre croissant de capteurs installés sur les procédés rendent très difficile l'analyse des résultats. L'analyse en composantes principales (*ACP*) et les moindres carrés partiels (*MCP*) permettent de réduire le nombre de variables à traiter par la détermination de relations linéaires entre elles (variables latentes), permettant ainsi d'expliquer la variance dans des séries de données.

3.2.2 L'approche multi-signaux

Ce sont de méthodes qui utilisent plus d'information que celles apportées par les seuls capteurs physiques. Ces informations peuvent provenir de la connaissance du comportement entrée/sortie d'un procédé ou des processus internes qui en gouvernent l'évolution.

Selon la méthode, différents types de modèle sont utilisés. Par exemple, pour les approches utilisant l'estimation d'état ou l'estimation paramétrique, on utilisera des modèles analytiques alors que pour l'approche systèmes experts, on recourra à des modèles de type base de connaissance. Le modèle servant directement de référence pour la détection de défauts, la qualité du résultat dépend directement de la qualité des modèles. La mise en œuvre de ces méthodes nécessite donc une modélisation précise.

La détection de défaut basée sur l'utilisation de modèles peut être divisée en deux étapes principales : la génération de résidus et la prise de décision. Lors de la première étape, les signaux d'entrée et de sortie du système sont utilisés pour générer un résidu – c'est-à-dire un signal mettant en évidence la présence d'un défaut. En général, en régime de fonctionnement normal, ce signal est statistiquement nul et s'écarte notablement de zéro en présence de défaut. La génération de résidus est propre à la méthode utilisée. Durant la seconde étape, les résidus sont analysés pour décider s'il y a ou non présence de défaut, sur quelle composante du système il est intervenu (localisation) et pour déterminer la nature du défaut et sa cause (identification). La décision peut s'effectuer à l'aide d'un simple test de dépassement de seuil sur les valeurs instantanées ou les moyennes de résidus, en utilisant des fonctions floues, en faisant appel également à la reconnaissance de formes ou en utilisant des seuils adaptatifs qui évoluent en fonction du point de fonctionnement du processus surveillé.

La figure 1.5 présente l'architecture générale de la détection de défauts basée sur l'utilisation de modèles [Isermann, 1984]. La génération de résidus s'effectue sur la base d'estimations (états ou paramètres) ou à l'aide d'un raisonnement heuristique. L'étape de prise de décision permet ensuite de détecter la présence de changements, d'identifier la nature, l'amplitude du changement voire localiser les défauts éventuels.

Parmi les différentes méthodes de détection et de diagnostic utilisant des modèles mathématiques, nous trouvons principalement l'espace de parité, les observateurs et l'estimation paramétrique.

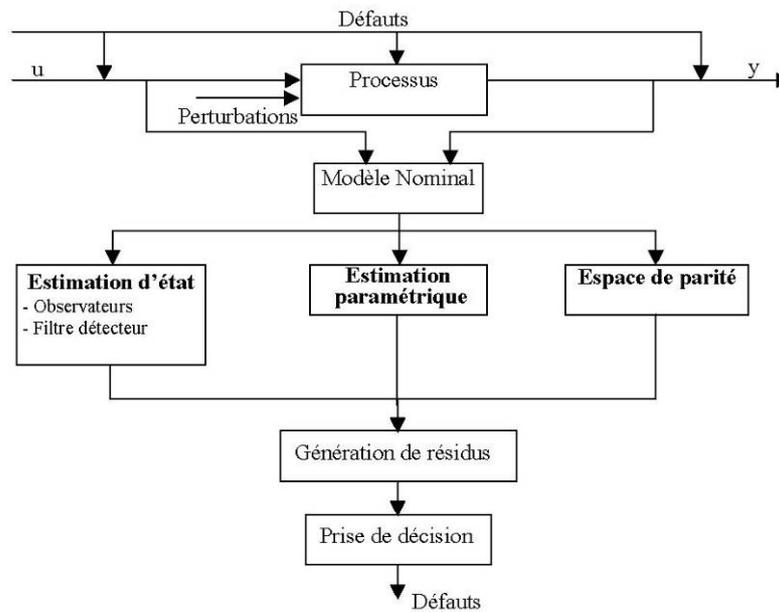


Figure 1.5 Architecture générale de la détection de défauts à base de modèles

3.2.2.1 Espace de parité

Une relation de redondance analytique est une équation dans laquelle toutes les variables sont connues. La génération de telles relations permet d'engendrer des résidus. Pour la détection de défauts basée sur l'utilisation de modèles, un résidu est un signal temporel, fonction des entrées et des sorties du processus, indépendant (le plus possible) du point de fonctionnement de celui-ci. En l'absence de défauts, ce résidu est statistiquement nul. Lors de l'apparition d'un défaut, son amplitude évolue de manière significative.

L'approche la plus classique est celle dite de l'espace de parité. Les relations de parité utilisent la redondance directe au moyen de relations algébriques statiques liant les différents signaux ou la redondance temporelle issue de l'utilisation de relations dynamiques. Le terme « parité » a été emprunté au vocabulaire employé pour les systèmes logiques où la génération de bits de parité permet la détection d'erreur.

3.2.2.2 Observateurs

La génération de résidus à l'aide d'une estimation d'état consiste à reconstruire l'état ou, plus généralement, la sortie du processus à l'aide d'observateurs et à utiliser l'erreur d'estimation comme résidu. Cette méthode s'est beaucoup développée car elle donne lieu à la conception de générateurs de résidus flexibles.

3.2.2.3 Estimation paramétrique

L'approche d'estimation paramétrique considère que l'influence de défauts se reflète sur les paramètres et non pas uniquement, comme c'est le cas des observateurs, sur les variables du système physique. Le principe de cette méthode consiste à estimer en continu des paramètres du procédé en utilisant les mesures d'entrée/sortie et en évaluant la distance qui les sépare des valeurs de référence de l'état normal du procédé. L'estimation paramétrique possède l'avantage d'apporter de l'information sur l'importance des déviations. Toutefois, un

des inconvénients majeurs de la méthode réside dans la nécessité d'avoir un système physique excité en permanence. Ceci pose donc des problèmes d'ordre pratique dans le cas de procédés dangereux, coûteux ou fonctionnant en mode stationnaire. De plus, les relations entre paramètres mathématiques et physiques ne sont pas toujours inversibles de façon unitaire, ce qui complique la tâche du diagnostic basé sur les résidus.

3.3 Les méthodes qualitatives basées sur un raisonnement causal

Les systèmes de surveillance et de diagnostic basés sur des modèles quantitatifs, et d'une certaine manière les approches de classification, ont une capacité explicative quasiment inexistante, ce qui ne répond pas à l'objectif d'assistance à l'opérateur en cas de fonctionnement anormal du procédé. Ils sont conçus surtout pour la phase de détection. La communauté de l'IA propose un raisonnement qualitatif basé sur une modélisation causale qui offre un support explicatif et adductif nécessaire au diagnostic et notamment à la tâche de localisation. En effet, le diagnostic est typiquement un processus causal puisqu'il consiste à établir des hypothèses sur les composants défectueux qui sont à l'origine du dysfonctionnement observé.

Le raisonnement qualitatif et les modèles causaux permettent de surmonter les difficultés liées aux procédés complexes évoqués précédemment. Ils s'appuient sur une décomposition du procédé en un nombre de sous-modèles élémentaires. De ce fait, il est possible de focaliser le raisonnement sur des relations spécifiques pour la localisation de défauts.

Une structure causale est une description qualitative de l'effet ou de l'influence qu'ont les entités du système (p.ex. variables, défauts) les unes sur les autres. Elle peut être représentée par un graphe direct (*digraphe*). Quand les nœuds du graphe représentent les variables du système et les arcs symbolisent les relations normales et déterministes entre elles, le graphe causal est souvent appelé *graphe d'influence* [Gentil et al., 04].

Le diagnostic basé sur un graphe d'influence consiste à rechercher la variable source dont la déviation est suffisante pour expliquer toutes les déviations détectées sur d'autres variables [Travé-Massuyès et al., 97]. L'algorithme de diagnostic est, en général, une procédure de recherche avant/arrière depuis la variable incohérente (détection). La recherche arrière limite l'espace de défaut en enlevant les observations normales en amont sur le chemin causal. Ainsi chaque déviation primaire génère une hypothèse qui conduit à une recherche avant utilisant les états des variables et les fonctions attachées aux arcs. La phase de localisation consiste à chercher quel composant du système ne fonctionne pas correctement en utilisant la connaissance sur la structure du système, ses défaillances potentielles et les observations disponibles. Le résultat du diagnostic peut être un arc pointant soit sur la variable source (panne de composant) soit sur une perturbation non mesurable qui affecte directement cette variable. Un grand avantage des approches causales est que, en général, la connaissance de comportements (modèles) de défaut n'est pas nécessaire pour la localisation.

Deux types principaux de structure causale sont proposés :

i) Le digraphe représente la calculabilité issue de la connaissance des relations

mathématiques (équations différentielles ou à différences). Il peut être obtenu à travers un mécanisme d'*ordonnement causal*² [Iwasaki & Simon, 86], [Pons & Través-Massuyès, 97] ou bien par la théorie de *graphe bipartite* [Blanke et al., 03][Gehin et al., 00].

- ii) Le digraphe représente une connaissance fonctionnelle du procédé; les nœuds sont liés aux variables considérées significatives par la surveillance et les arcs sont liées aux phénomènes physiques (p.ex. bilan de matière) [Kleer, 86].

Ainsi, le premier type lie la causalité aux équations décrivant le système (analyse globale) tandis que le deuxième axe lie la causalité à la structure du système (analyse locale). Issu d'une connaissance profonde des mécanismes physiques, nous avons également le *bond-graph*, modèle de base de nombreuses approches de diagnostic comme le graphe causal temporel de [Mostermann & Biswas, 99], [Mostermann, 01]. Le diagnostic (suivi et localisation) de défauts abrupts y est traité. Historiquement l'approche a été plus particulièrement envisagée pour la modélisation de la dynamique de systèmes mécaniques et électroniques.

La construction d'un graphe causal est un processus assez complexe qui demande une connaissance structurelle et fonctionnelle de l'installation. La connaissance experte est également prise en compte pour la définition des besoins de surveillance. [Heim et al., 03] énumère un certain nombre de consignes pour la construction d'un graphe causal dans un contexte de système complexe: i) identification du système physique, ii) division en sous-systèmes, iii) définition et affectation d'une configuration à chaque sous-système, iv) identification d'un ensemble de relations physiques, v) connexion des relations aux composants physiques, vi) détermination de causalité, vii) réduction (élimination de variables non mesurables), viii) approximation (élimination de relations négligeables) et ix) quantification (identification des paramètres de la fonction de transfert).

Le contexte de la représentation causale et les besoins du diagnostic conduisent à différents types de relations qui étiquettent les arcs et induisent la démarche de suivi et le mécanisme de propagation d'influence. Le type de graphe causal le plus simple est le graphe orienté signé (SDG³) qui décrit les variables dans un espace qualitatif restreint $\{-,0,+ \}$ liées par des arcs qui modélisent le signe des influences $\{-,+ \}$. La faiblesse de représentation qui induit des ambiguïtés est résolue à travers l'augmentation de l'espace qualitatif *crisp* ou flou. *Biotech* [Bousson et al., 98] est un système conçu pour la surveillance et l'aide à la conduite de procédés chimiques complexes (fermentation) qui s'appuie sur une connaissance experte des relations d'influence entre les variables définies dans un domaine symbolique élargi $\{pp, p, m, f, ff\}$, sur un graphe causal. Cependant, ces approches souffrent du manque d'information temporelle, ce qui empêche leur utilisation pour le diagnostic de comportements dynamiques. Toutefois, un diagnostic basé sur le SDG, proposé par [Scenna, 00] dans le cadre de procédés batch, surmonte cette limitation. Mais cela est fait au détriment d'une complexité combinatoire significative qui résulte de la proposition d'une nouvelle structure de SDG à chaque changement de tendance des variables concernées.

Des nombreux travaux sont basés sur des modèles causaux. Dans [Través-Massuyès et al., 97], les principales approches sont passées en revue et un historique de l'utilisation des

² De l'anglais : *causal ordering*.

³ De l'anglais : *signed directed graph*

représentations causales est présenté. Nous allons nous focaliser sur certaines approches liées à notre problématique de modélisation et de surveillance des systèmes dynamiques hybrides complexes.

4 Conclusion

Nous avons dans, ce chapitre réalisé une étude bibliographique générale mais non exhaustive sur les méthodes de diagnostic de type traitement données et les méthodes à base de modèles dans le but d'identifier et d'évaluer des solutions qui concernent notre problématique. Nous avons vu que le diagnostic dépend d'une représentation et d'une discrimination de tous les modes de défaut à travers des mécanismes d'inférences (à base de règles) ou d'un mécanisme de classification en ligne. La localisation de défaut dépend de ces mécanismes qui associent les modes de défaut répertoriés à des composants.

Chapitre2 : Caractérisation et simulation des systèmes dynamiques hybrides

1 Introduction

Les innovations technologiques ont eu un impact considérable sur l'apparition des processus dynamiques ayant une nature hétérogène mélangeant le continu et le discret. Ces progrès technologiques ont un impact similaire dans le domaine de la recherche scientifique où on constate un intérêt particulier pour l'étude de ces systèmes dits "hybrides". De tels systèmes sont caractérisés par l'interaction des parties continues régies par des équations différentielles et des parties discrètes, décrites par des machines à états finis ou des réseaux de Pétri.

En automatique, les systèmes physiques sont souvent représentés par un modèle dynamique continu ou par un modèle à événements discrets. La nature de chaque modèle est définie selon les variables utilisées pour décrire l'état du système et la variable caractérisant le temps. Il est important dans de nombreux cas d'utiliser l'une de ces deux catégories de modèles. Cependant la majorité des systèmes complexes réalistes mélangeant le continu et le discret ne peuvent pas être classés ni dans la catégorie "système continu" ni dans la catégorie "système discret". Il est nécessaire alors d'utiliser des modèles hybrides permettant la prise en compte à la fois des variables continues et des variables discrètes ainsi que l'interaction entre elles. Plusieurs modélisations hybrides ont été développées dans la littérature. Elles peuvent être classées en trois catégories principales [Zaytoon, 01] :

- Les approches basées sur une extension de techniques de modélisation de systèmes continus comme le formalisme de bond-graph mixte.
- Les approches basées sur l'extension de modèles de systèmes à événements discrets parmi lesquels nous pouvons citer les réseaux de Petri hybrides.
- Les approches mixtes, combinant à la fois le continu et le discret dans une structure unique.

L'approche mixte repose sur une collaboration de deux modèles. Chaque aspect est décrit sous une forme classique et l'aspect hybride est pris en compte dans l'interface entre les deux modèles. La figure 2.1 illustre la structure mixte d'un SDH (Système Dynamique

Hybride) où la partie événementielle est décrite par un automate à états finis et la partie continue est donnée par une collection de modèles. Chaque partie du SDH dispose des entrées (discrète σ , continue u) et des sorties (discrète Ψ , continue y). L'état discret q détermine la dynamique continue spécifique de la partie continue du SDH par l'interface discret-continu (D/C). Par l'intermédiaire de l'interface continu-discret (C/D), les signaux provenant de la partie continue déterminent des événements permettant des transitions de l'état discret d'une situation vers une autre. Ainsi l'interface (D/C - C/D) permet l'interaction des variables discrètes et des variables continues.

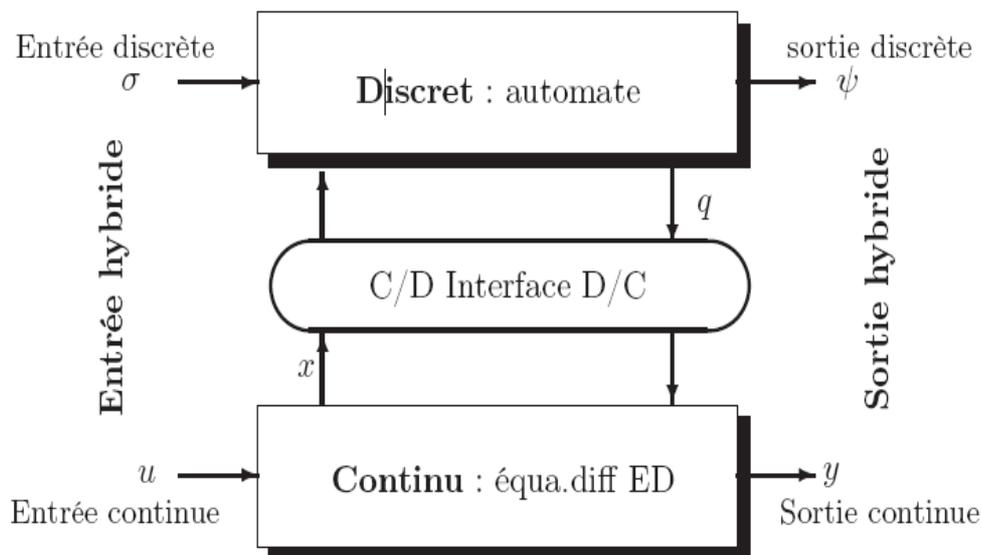


Figure 2.1 Systèmes hybrides : interaction du continu et du discret

2 Systèmes dynamiques hybrides

Les systèmes réels sont souvent des systèmes complexes dont la dynamique est modélisée, d'un point de vue macroscopique, par des phénomènes discrets et continus. Il apparaît alors évident que ces systèmes ne peuvent se contenter d'une représentation homogène à dynamique purement continue ou purement événementielle. Des études visant à considérer les interactions entre ces deux mondes s'imposent. Dans ce contexte, une attention particulière s'est portée sur les SDH ces dernières années [Alur et al., 95]; [Engell, 97]; [Branicky, 95]; [Guéguen et Lefebvre, 01] et de nombreuses approches de modélisation traitant à la fois les aspects continus et discrets ont été proposées. Les formalismes qui en découlent présentent des propriétés plus ou moins intéressantes en termes de modularité, réutilisabilité, structuration des données, etc. Dans ce contexte, la section suivante est consacrée aux différentes approches offertes pour la modélisation des SDH. La présentation qui en est faite ne se veut pas exhaustive et le lecteur pourra consulter les articles référencés dans les sections concernées pour plus d'informations. Signalons enfin que notre attention s'est particulièrement portée sur les modèles dont la partie événementielle s'appuie sur les réseaux de Petri, formalisme sur lequel est basé le modèle développé dans nos travaux.

La modélisation des systèmes dynamiques hybrides doit décrire deux comportements: d'une part, la dynamique continue généralement représentée par un système d'équations différentielles et algébriques et d'autre part, la dynamique discrète représentée par un ensemble d'états et de transitions. Les études menées pour concilier les composantes continue et discrète ont conduit à de nombreux formalismes cités dans notre introduction.

2.1 Extension des modèles à dynamique continue

Dans le cadre de l'extension des modèles continus, une première approche proposée repose sur l'utilisation de *variables mixtes*. Celle-ci consiste à introduire des variables booléennes ou entières au sein du modèle continu. Par exemple, l'état « ouvert » ou « fermé » d'une vanne tout ou rien peut être représenté par une variable booléenne égale respectivement à *un* ou à *zéro*. Cette approche peut également être illustrée au travers de l'exemple de l'évaporation de Rayleigh, qui consiste à chauffer un mélange liquide jusqu'à son point de bulle afin de le vaporiser.

Pour illustrer ceci, considérons une cuve, contenant initialement un binaire ($n_c = 2$) de composition x , de rétention liquide U_1 (la rétention vapeur est supposée négligeable devant la rétention liquide) et recevant une quantité de chaleur Q . Au point de bulle, une phase vapeur de composition y apparaît et s'échappe dans l'atmosphère avec un débit V . Le modèle continu global associé à ce système est le suivant:

$$\text{Bilan matière global : } \frac{dU_1}{d\theta} + V = 0 \quad 2.1$$

$$\text{Bilans matière partiels : } \frac{d(U_1 \cdot x_i)}{d\theta} + V \cdot y_i = 0 \quad (i = 1, n_c) \quad 2.2$$

$$\text{Bilan d'énergie : } \frac{d(U_1 \cdot h)}{d\theta} + V \cdot H - Q = 0 \quad 2.3$$

$$\text{Relation d'équilibre : } y_i - K_i x_i = 0 \quad (i = 1, n_c) \quad 2.4$$

$$\text{Somme des fractions molaires : } \sum_i x_i - \sum_i y_i - \tau = 0 \quad 2.5$$

$$\text{Equation de « switch » : } V \cdot \tau = 0$$

La gestion de la discontinuité liée au passage de l'état liquide à l'état liquide / vapeur est gérée grâce à l'introduction de la variable booléenne τ et de l'équation de « switch » associée. Initialement, le système est à l'état liquide et la variable τ est égale à 1. L'équation de switch garantit alors que le débit vapeur est égal à 0. Lorsque la température de bulle est atteinte, la variable τ change de valeur et devient égale à 0, entraînant alors la prise en compte de la phase vapeur dans le modèle.

2.2 Extension des modèles à événements discrets

2.2.1 Les automates temporisés

Les *automates temporisés* [Alur et Dill, 94] constituent une extension des *automates à états finis*. Ils sont pourvus, en plus, d'un ensemble d'*horloges* qui permettent de mesurer et de contrôler le temps écoulé entre les actions. Des contraintes sur les horloges peuvent être associées aux états et aux transitions du modèle.

A titre d'exemple, construisons l'automate temporisé associé au système de la figure 2.2

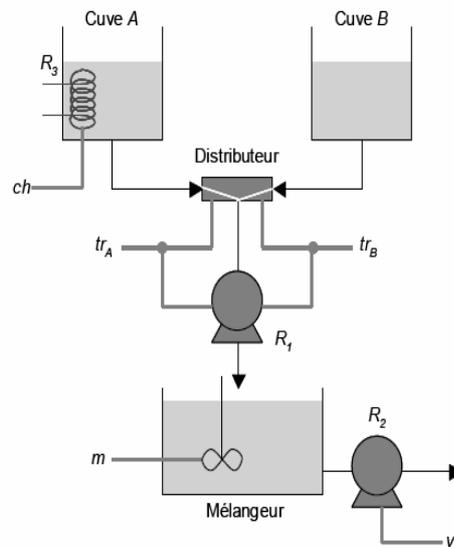


Figure 2.2 Exemple illustratif : transfert et mélange des produits A et B

Le système étudié est constitué de deux cuves d'alimentation (contenant respectivement les produits A et B) et d'un mélangeur. Il décrit le transfert des produits A et B via une pompe unique R1. Cette ressource est partagée et l'introduction des produits A et B dans le mélangeur ne peut se faire simultanément. Avant d'être introduit, le produit A est chauffé et seul son transfert peut stopper la chauffe ; la ressource correspondante R3 (l'échangeur de chaleur) est alors instantanément libérée. Les deux produits sont mélangés dans la cuve M pendant un certain temps puis vidangés vers une autre cuve grâce à la pompe R2. On suppose qu'aucun transfert de A ou de B n'est autorisé pendant une opération de mélange ou de vidange du mélangeur. La figure 2.3 met en évidence les différentes commandes appliquées au procédé: ch correspond au démarrage de la chauffe, tr_A et tr_B désignent respectivement le transfert de produit A et du produit B, m marque le début de l'opération de mélange et v indique le début de l'opération de vidange du mélangeur.

Le temps de chauffe du produit A est piloté et surveillé par l'horloge y. Cette dernière est remise à zéro dès le début d'une chauffe au niveau des transitions $T_B \rightarrow CT_B$ et $R \rightarrow C$. Le temps de mélange est également surveillé et les produits contenus dans le mélangeur ne peuvent être vidangés qu'après vingt minutes de mélange. Cette durée est contrôlée par l'horloge x qui est réinitialisée au début de chaque opération de mélange au niveau des transitions $T_A \rightarrow M$ et $T_B \rightarrow M$.

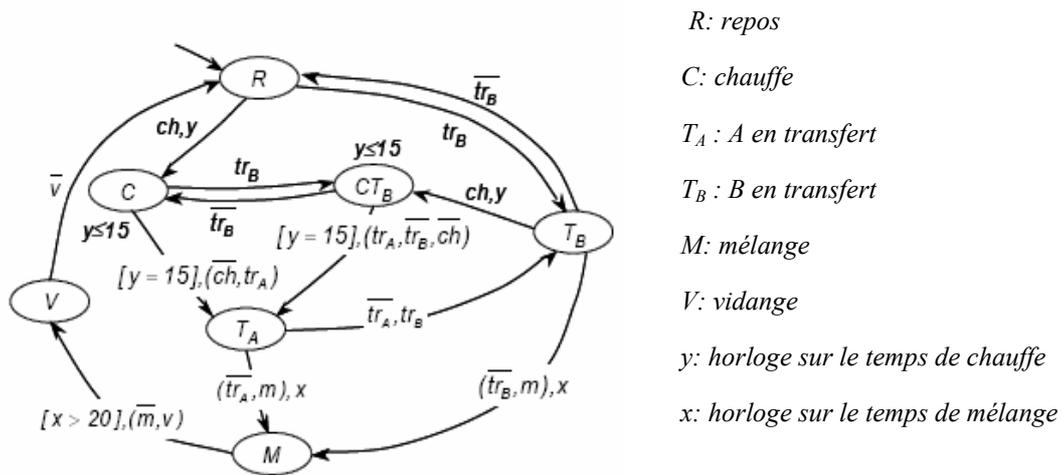


Figure 2.3 Automate temporisé simplifié

2.2.2 Les réseaux de Petri temporisés, temporels et stochastiques

La prise en compte des aspects quantitatifs du temps dans les réseaux de Petri a donné naissance aux réseaux de Petri temporisés, aux réseaux de Petri temporels et aux réseaux de Petri stochastiques.

2.2.2.1 Les réseaux de Petri temporisés

Dans les réseaux de Petri temporisés, une durée de franchissement est associée soit à la place (*Rdp p-temporisée*) [Sifakis, 77], soit à la transition (*RdP t-temporisée*) [Ramchandani, 74] et le jeton est indisponible pendant toute la durée de la temporisation; il est réservé. Afin d'illustrer le formalisme *RdP t-temporisée*, reprenons l'exemple précédent pour lequel, des contraintes temporelles ont été introduites, et construisons le réseau de Petri correspondant.

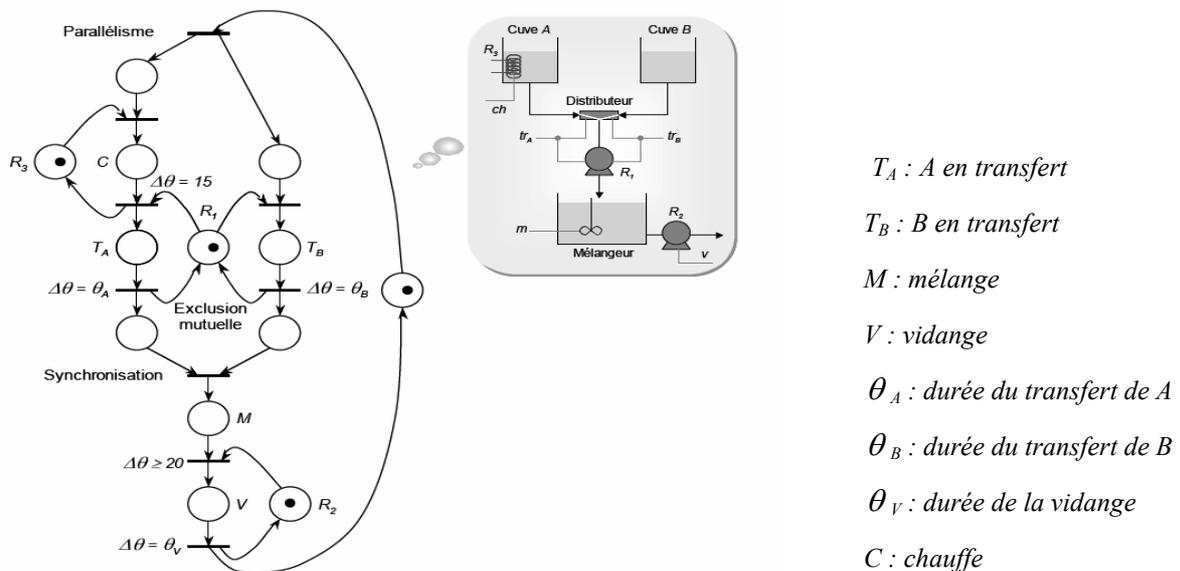


Figure 2.4 Réseau de Petri t-temporisé

En plus des contraintes temporelles introduites en début de section, des temps de transfert θ_A , θ_B et θ_V calculés au préalable en fonction du débit et de la quantité de produit à transférer, sont ajoutés aux transitions. Si ce modèle permet d'intégrer les contraintes temporelles fixées a priori, il ne peut, en revanche, tenir compte de l'évolution des variables d'état du système. En effet, supposons qu'au cours de la simulation, une perturbation provoque une diminution de la valeur de débit, l'objectif restant cependant identique à savoir transférer la même quantité de produit. En toute rigueur, le temps de transfert devrait en être augmenté. Or, ceci n'est pas observé car la perturbation n'est pas détectable pendant la simulation. Cet exemple souligne la nécessité de tenir compte de l'évolution continue non seulement de la variable temporelle, mais également des variables d'état du système.

2.2.2.2 Les réseaux de Petri temporels

Les *réseaux de Petri temporels* [Merlin et Segall,76];[Berthomieu et Menasche, 83] étendent les *réseaux de Petri ordinaires* en associant à la transition t , un intervalle fermé rationnel $[\theta_{\min}, \theta_s, \theta_{\max})$. La transition n'est alors franchissable que lorsque la durée de sensibilisation, θ_s , est comprise entre les bornes de l'intervalle θ_{\min} et θ_{\max} . Contrairement aux réseaux de Petri temporisés, les jetons marquant les places d'entrée de la transition sont ici *disponibles* et peuvent éventuellement être engagés dans une autre transition en conflit avec la transition t . Notons par ailleurs que les RdP temporels permettent de représenter aisément le mécanisme classique dit du «chien de garde» [Valette, 02].

2.2.2.3 Les réseaux de Petri stochastiques

Dans un RdP temporisé ou temporel, des durées sont associées aux places ou aux transitions du réseau. Comme indiqué dans la section 2.3.2.1 ces modèles sont bien adaptés pour étudier les systèmes pour lesquels les aspects quantitatifs sont bien définis. En revanche, certains phénomènes (tels que l'apparition de pannes dans un parc de machines ou la connaissance imprécise de certains paramètres tels qu'un débit) sont imprévisibles et ne peuvent alors plus être traités par ce type de modèle. Dans ce contexte, les *réseaux de Petri stochastiques* ont été introduits par Florin et Natkin [Florin et Natkin, 85]; [Florin *et al.*, 91]; [Marsan,89] afin de modéliser efficacement les phénomènes aléatoires. Ainsi, dans un RdP stochastique, les durées de sensibilisation associées aux transitions résultent d'un tirage aléatoire. L'hypothèse la plus couramment employée est que cette durée est distribuée selon des lois exponentielles. Une application utilisant ce formalisme pour modéliser et analyser les performances d'un atelier de maintenance est traitée dans [Abbou *et al.*, 03].

2.2.2.4 Les réseaux de Petri continus

Dans un réseau de Petri ordinaire, le marquage d'une place est discret et, pour un système complexe, l'analyse du réseau qui en résulte conduit souvent à l'énumération d'un très grand nombre d'états accessibles. Afin de rendre le modèle plus compact, une autre forme de réseau de Petri, le *réseau de Petri continu*, étend l'évolution du marquage dans l'espace des réels et considère le franchissement des transitions comme un processus continu [David et Alla, 87]. Le franchissement des transitions n'est désormais plus instantané puisque la

marque⁴ franchit la transition par quantité infinitésimale; dans ce contexte, des vitesses de franchissement sont associées aux transitions. Selon la nature de ces vitesses de franchissement, plusieurs modèles dérivent des RdP continus: les *RdP continus à vitesses constantes* [David et Alla,89], les *RdP continus à vitesses variables* [David et Alla,89], les *RdP continus asymptotiques* [Le Bail et al.,93]. Cette approche s'avère particulièrement bien adaptée pour représenter les systèmes à flots continus fonctionnant en régime permanent. Un exemple d'application concernant l'analyse des lignes de fabrication est traité dans [Zerhouni et Alla, 92].

2.2.2.5 Les réseaux de Petri hybrides

Il est apparu nécessaire, dans certains cas, de représenter sur un même modèle des phénomènes à la fois continus (comme le traitement d'un flux de matière continu) et discrets (comme l'occupation d'une ressource). Ni le formalisme *RdP discret*, ni le formalisme *RdP continu* ne permettait de représenter de tels systèmes. Cela a ainsi conduit à la définition d'une nouvelle extension : les *réseaux de Petri hybrides* [Le Bail et al.,91] dans lesquels coexistent des places et des transitions continues et discrètes. Plusieurs applications utilisant ce formalisme ont été réalisées. Citons par exemple la modélisation d'un système de production de la société Motorola [Alla et al., 92], la modélisation d'un réseau d'eau potable constitué de réservoirs interconnectés par des vannes deux états [Alla, 94] ou la modélisation d'un système de production d'énergie hydraulique [David, 91].

Afin d'illustrer le modèle basé sur les réseaux de Petri hybrides, considérons comme exemple le cas du système de deux réservoirs représenté sur la figure 2. 5.

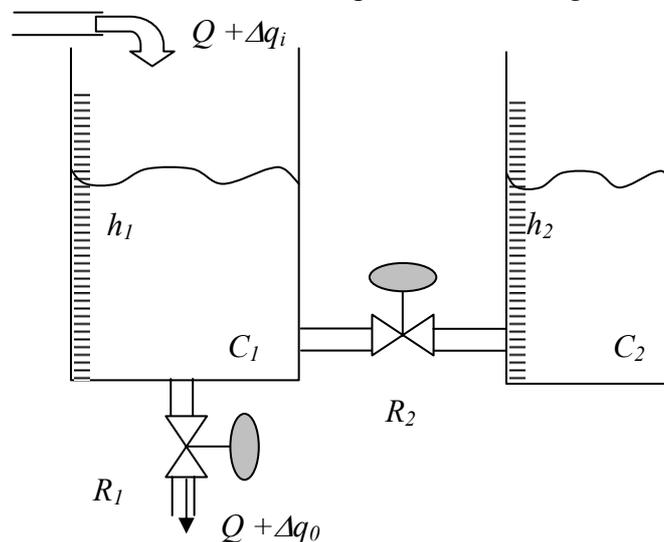


Figure 2.5 Système de 2 Réservoirs

⁴ Le terme « jeton » est réservé au marquage discret alors que le terme « marque » est réservé au marquage continu.

En utilisant l'hypothèse que l'écoulement dans les tuyaux est laminaire, le système peut être modélisé avec des équations linéaires. On peut définir la résistance d'écoulement comme la relation entre le changement de la différence de niveau et le changement du débit,

$$R = \frac{\Delta_h}{\Delta_q} \quad 2.6$$

et la capacité de réservoir comme la relation entre le changement du volume stocké et le changement du niveau,

$$c = \frac{\Delta_v}{\Delta_h} \quad 2.7$$

Nous considérons que le robinet R_1 est ouvert et le robinet R_2 peut être ouvert ou fermé. Quand le robinet R_2 est fermé (sans écoulement entre les réservoirs) le comportement du système peut être modélisé par les équations suivantes :

$$\frac{dh_1}{dt} = -\frac{1}{R_1 C_1} h_1 + \frac{1}{C_1} q_i \quad 2.8$$

$$\frac{dh_2}{dt} = 0 \quad 2.9$$

Quand le robinet R_2 est ouvert le comportement du système peut être décrit par les équations suivantes :

$$\frac{dh_1}{dt} = -\left(\frac{1}{R_1 C_1} + \frac{1}{R_2 C_1}\right) h_1 + \frac{1}{R_2 C_1} + \frac{1}{C_1} q_i \quad 2.10$$

$$\frac{dh_2}{dt} = \frac{1}{R_2 C_2} h_1 - \frac{1}{R_2 C_2} h_2 \quad 2.11$$

Le réseau de Pétri différentiel est défini par $B = [R, f, M_0, Time]$, où :

- R est un Réseau de Pétri défini comme $R = [P, T, Pre, Post]$ avec :
 - P : un ensemble fini de places avec $|P| = n$
 - T: un ensemble fini de transitions avec $|T| = m$
 - $P \cap T = \{\}$ et $P \cup T \neq \{\}$
 - $Pre(P_i, T_j)$ est une fonction qui associe des arcs depuis une place vers une transition.
 - $Post(P_i, T_j)$ est une fonction qui associe des arcs depuis une transition vers une place.
- $f : P \cup T \rightarrow \{D, DF\}$, appelée « fonction différentielle », s'associe à chaque nœud si c'est un nœud discret ou différentiel.

- M_0 est le marquage initial.
- $Time$ est une fonction qui associe un nombre rationnel à chaque transition qui peut évoluer dans le temps, et également un retard pour la transition différentielle (représentant le pas d'intégration).

Le réseau de Pétri Différentiel Synchronisé représentant l'exemple du système de 2 Réservoirs possède les éléments suivants :

- $R = [P, T, Pre, Post]$

- $P = \{P_1, P_2, P_3, P_4\}$

- $T = \{T_1, T_2, T_3, T_4, T_5, T_6\}$

- $Pre = \begin{bmatrix} 1/C_1 & -1/R_1C_1 & -(1/R_1C_1 + 1/R_2C_1) & 0 & 0 & 0 \\ 0 & 0 & -1/R_2C_2 & -1/R_2C_2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- $Post = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$

- $f = \{P_1=P_2=DF, P_3=P_4=D, T_1=T_2=T_3=T_4=DF, T_5=T_6=D\}$
- $M(0) = [h_{10} \ h_{20} \ 1 \ 0]^T$
- $Time = \{(T_1, (q_1, T_s)), (T_2, (m_1, T_s)), (T_3, (m_2, T_s)), (T_4, (m_2, T_s)), (T_5, T_s), (T_6, T_s)\}$, où T_s est un temps d'échantillonnage approprié et m_1, m_2 sont les marquages dans P_1 et P_2 respectivement.

Un diagramme de ce réseau de Petri est donné par la figure 2.6 :

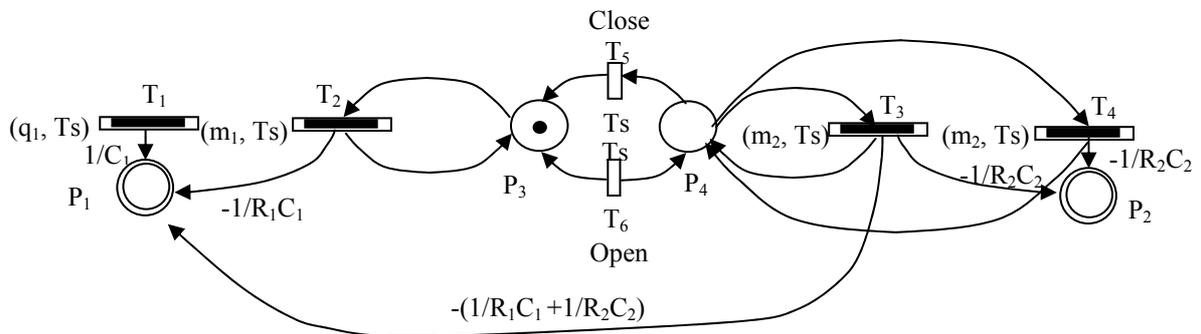


Figure 2.6 Réseau de Petri de l'exemple des 2 réservoirs

2.2.2.6 Les réseaux de Petri lots

Les *réseaux de Petri lots* (Demongodin *et al.*, 93, 98; Demongodin, 01; Demongodin et Rouibia, 03 ; Caradec et Prunet, 98) étendent les RdP hybrides pour modéliser les systèmes de production dans lesquels coexistent simultanément des phénomènes *d'accumulation* et de *retard de propagation*. Par exemple, la disponibilité d'un lot de produit qui entre sur un convoyeur à bande n'est pas instantanée car il doit être transporté de l'amont vers l'aval du convoyeur (*phénomène de retard de propagation*). Par ailleurs, une partie du lot peut être accumulée à la sortie si la totalité du flux sortant du convoyeur ne peut être absorbée par le convoyeur ou par la machine située en aval (*phénomène d'accumulation*). Ni les *réseaux de Petri temporisés*, ni les *réseaux de Petri hybrides* ne permettent de modéliser la simultanéité des deux phénomènes. En revanche, les *réseaux de Petri lots* lèvent cette difficulté et définissent ainsi un nouveau type de place (*place lot*) et de transition (*transition lot*). Un jeu de paramètre décrivant le lot (densité, longueur, etc.) et une loi continue d'évolution temporelle du système active pendant la durée du marquage sont attribués à la place. De plus, des vitesses de franchissement liées au contenu informationnel des places situées en amont et en aval des transitions sont associées aux transitions. Un exemple d'application utilisant les *RdP lots* pour l'optimisation du fonctionnement des lignes de production à haute cadence est traité dans [Audry *et al.*, 94].

2.2.3 Modèles mixtes

Les deux approches décrites précédemment reposent sur l'intégration d'aspects hybrides dans le modèle continu ou le modèle discret du système. Malheureusement, elles ne permettent pas de représenter efficacement à la fois la partie discrète et la partie continue mais se contentent de privilégier l'une des deux au détriment de l'autre. Une troisième approche de modélisation des systèmes réels, basée sur les *modèles mixtes*, permet d'apporter une solution à ces problèmes. Dans une *approche mixte*, chacune des deux composantes (discrète et continue) est représentée de façon rigoureuse et explicite selon un des formalismes décrits dans les sections précédentes et leur collaboration est prise en compte dans l'interface qui les relie. La résolution du modèle continu déclenche l'évolution des variables au cours du temps et valide certaines transitions. L'évolution du modèle discret engendre alors la mise en place d'un nouvel état discret qui se traduit par l'élaboration d'un nouveau système d'équations. Plusieurs modèles mixtes ont été définis. Cette section se propose d'en présenter quelques-uns.

2.2.3.1 Les automates hybrides

Les *automates hybrides* [Alur *et al.*, 95] sont une extension des *automates à états finis*. Ils représentent des systèmes qui intègrent deux composantes : celle ayant un comportement discret, modélisée naturellement par un *automate à états finis* et celle dont le comportement varie de manière continue dans le temps, modélisée par un *système algébro-différentiel*. Notons qu'un automate temporisé est un cas particulier d'un automate hybride pour lequel le temps constitue la seule variable continue. Les problèmes soulevés par cette approche résident toujours dans le manque de structuration des données et de lisibilité du modèle.

2.2.3.2 Les « statecharts » hybrides

Les « *statecharts* » hybrides [Kesten et Pnueli,92], qui constituent une extension des « *statecharts* », associent un ensemble d'équations différentielles aux états. Cette représentation renforce la puissance du modèle liée à la structuration des modèles : en effet, l'utilisation de la hiérarchie des états permet de structurer la partie événementielle et l'association d'équations aux états permet de structurer les relations entre les grandeurs continues.

2.2.3.3 Les réseaux de Petri mixtes

L'approche basée sur les *réseaux de Petri mixtes (RdPM)* [Valentin-Roubinet, 99] consiste à interpréter les RdP par des équations différentielles. Un RdP mixte est constitué [Zaytoon,01] :

- d'un ensemble de *RdP interprétés* (David et Alla, 94; Daubas *et al.*, 94) synchronisés par des événements internes. Dans cette représentation, la place peut être interprétée soit comme une activité (un système algébro-différentiel lui est alors associé et le nombre de jetons qu'elle peut accepter est limité à *un* afin de garantir la cohérence du modèle), soit comme une condition logique, une ressource disponible ou l'état d'un stock (aucune équation ne lui est attribuée et le nombre de jetons est illimité) ;
- d'une partie continue qui comprend l'ensemble des variables continues (globales et accessibles de partout dans le modèle et à tout instant) et l'ensemble des équations;
- d'une interface hybride qui assure la collaboration entre la dynamique continue et la dynamique événementielle. Elle est caractérisée par :
 - un ensemble d'équations actives pour chaque place marquée;
 - un ensemble d'équations résultant de la concaténation des équations actives de toutes les places marquées des réseaux définies pour chaque configuration de marquage;
 - une application qui associe à chaque transition, un vecteur dont chaque composante correspond à la fonction de saut sur chacune des variables ;
 - une application qui associe à chaque transition, une condition sur les variables continues devant être vérifiée pour la franchir.

Cette approche a notamment été utilisée dans le cadre de l'ordonnancement. Un cas industriel est traité dans [Pagès et Pingaud, 95] et présente un outil d'aide à l'ordonnancement d'ateliers « batchs » à production continue. De même, [C. Valentin-Roubinet 98] utilise les RdP mixtes pour modéliser une unité de stockage de gaz. Cependant, notons que l'inconvénient majeur des réseaux de Petri mixtes réside dans l'utilisation de variables globales, ce qui ne contribue aucunement à la structuration des modèles.

2.2.3.4 Les réseaux de Petri prédicats-transitions différentiels

Les *réseaux de Petri prédicats-transitions différentiels (RdPPTD)* s'appuient sur le formalisme *réseau de Petri mixte* présenté précédemment et sur les *réseaux de Petri prédicats-transitions*.

En effet, ils sont nés avec l'idée de tirer avantage des aspects « structuration des données » propres aux *RdPPT* d'une part et de la représentation rigoureuse de la partie continue propre aux *RdPM* d'autre part. Dans ce contexte, un *RdPPTD* représente un système dont le comportement discret est décrit par des *réseaux de Petri prédicats-transitions* et le comportement continu par des systèmes *d'équations différentielles algébriques* (Champagnat,98; Champagnat *et al.*,98a,99). Dans cette représentation, le comportement hybride est pris en compte d'une part, en associant les variables continues aux jetons et d'autre part, en associant un système algébro-différentiel aux places permettant de faire évoluer les variables d'état continues. Un jeton mis dans une place déclenche la résolution du système différentiel correspondant et provoque ainsi l'évolution continue au cours du temps des variables d'état. Ceci constitue une différence fondamentale avec le modèle *RdPPT* dans lequel les attributs des jetons ne sont modifiés que lors du franchissement des transitions.

Un exemple d'application concernant un processus continu de stockage de gaz est traité dans [Champagnat *et al.*, 1998b].

Le modèle *réseau de Petri différentiel à objets*, développé dans ce manuscrit, est fondé sur des principes analogues à ceux du modèle *RdPPTD*. Pour cette raison, une description plus détaillée des *RdPPTD* est réalisée dans le chapitre 3 traitant de la formalisation du modèle *RdPDO*.

2.2.4 Conclusion

Les formalismes proposés pour représenter les systèmes dynamiques hybrides sont nombreux. Ils se différencient par la technique de modélisation adoptée qui repose soit sur une approche visant à étendre un modèle discret ou un modèle continu, soit sur une approche visant à utiliser un modèle mixte. Afin de clarifier les idées, le tableau 2.1 propose une synthèse des formalismes utilisés pour la représentation des systèmes hybrides présentés dans cette section.

Extension des modèles continu	Extension des modèles discrets	Modèles mixtes
	Automates temporisés	
Utilisation de variables mixtes	RdP temporisés	Automates hybrides
Modèle de Tavernini	RdP temporels	« Statecharts » hybrides
Modèle unifié de Branicky	RdP stochastiques	RdP mixtes
Bond-graphs à commutation	RdP continus	RdPPTD
	RdP hybrides	RdPDO
	RdP lots	

Tableau 2.1 : synthèse des formalismes utilisés pour la représentation des systèmes hybrides

La technique de modélisation consistant à étendre un formalisme initialement prévu pour le domaine du discret ou du continu a montré ses limites. D'une part, elle privilégie l'une des deux dynamiques au dépend de l'autre ; or, les systèmes réels ont des comportements complexes qui nécessitent une modélisation précise tant au niveau de la partie continue que de

la partie discrète. D'autre part, elle apporte peu de clarté et de lisibilité dans la délimitation entre la vision discrète et la vision continue du système. Elle ne respecte pas, par ailleurs, certains principes d'évolution de marquage classiquement attribués aux réseaux de Petri : c'est le cas, par exemple, du réseau de Petri continu dans lequel deux transitions en conflit peuvent être franchies simultanément. Enfin, cette technique génère souvent des modèles à caractère spécifique et dont les applications sont réduites.

En revanche, la technique de modélisation reposant sur une approche mixte est plus générale et offre un potentiel de modélisation élevé dans chaque domaine (continu et discret). Par ailleurs, elle permet d'utiliser, pour la partie événementielle, des modèles de haut niveau (ex: *RdPPT*).

3 Simulation des systèmes dynamiques hybrides

3.1 Introduction

Les difficultés d'analyse et de synthèse des systèmes dynamiques hybrides font de leur simulation un outil d'étude privilégié. Les logiciels de simulation se sont développés depuis une trentaine d'années, initialement pour traiter des processus particuliers : physique nucléaire [Nakhle, 83], convertisseurs statiques [Foché,74, Iung, 76]. Ces logiciels sont en évolution constante, ils tendent à simuler des systèmes généraux [Branicky, 97] et sont passés aujourd'hui du domaine de la recherche aux applications industrielles. Les programmes de simulation sont très variés, ils dépendent souvent des modèles utilisés pour représenter le système dynamique hybride (Réseaux de Petri, automate hybride, etc.), de leur objectif (orienté vers une classe d'applications ou au contraire à vocation générale), de la manière dont ils ont été construits (à partir d'un logiciel orienté vers la simulation des systèmes discrets, ou d'un logiciel destiné à simuler des processus continus, ou encore directement pensés pour simuler un SDH). Ceci conduit à un foisonnement dans lequel il n'est pas toujours commode de se retrouver pour choisir un produit adéquat.

3.2 Pourquoi simuler?

A l'heure actuelle, les recherches concernant les méthodes et outils formels relatifs à l'analyse du comportement des systèmes dynamiques hybrides et à la synthèse de leurs lois de commande en sont encore à leurs débuts. La simulation reste donc un passage obligé lorsque l'on envisage:

- d'aider à la conception d'une installation et de valider cette conception,
- de valider le modèle élaboré (dans un but de prévision, par exemple) pour une installation existante,
- de valider la commande conçue pour une installation.

La simulation a ainsi pour objectif de détecter les erreurs dans les modèles élaborés avant d'implanter la commande correspondante. Il ne faut cependant pas perdre de vue qu'une simulation n'est qu'une représentation de la réalité et non la réalité elle-même... avec toutes les précautions que cela induit lors des premiers essais de fonctionnement !

Nous allons aborder les problèmes posés par la simulation numérique des systèmes dynamiques hybrides. Nous insisterons particulièrement sur les aspects qui ne dépendent pas

de l'outil de modélisation utilisé. Peu importe la modélisation choisie pour décrire la partie discrète, les tâches inhérentes à la simulation d'un SDH seront toujours les mêmes :

- simuler le fonctionnement de la partie discrète,
- en déduire le ou les modèles actifs pour la partie continue,
- résoudre les équations différentielles ou aux différences.

Rappelons ici une évidence parfois oubliée : si la qualité d'un modèle se mesure par sa capacité à prévoir le comportement du système lui-même, il faudra évaluer la qualité d'une simulation en comparant ses résultats avec ceux attendus du modèle. Lorsque l'on compare les résultats d'une simulation avec ceux du système physique on peut conclure, quand il y a coïncidence, que le modèle représente correctement le système et que la simulation donne une image fidèle du modèle. Si la simulation fournit des résultats discordants par rapport au système physique, il faudra s'interroger pour savoir qui, de la modélisation ou de la simulation, est à mettre en cause. Ce point est très délicat, car il est en général difficile d'obtenir des expressions analytiques pour représenter le comportement d'un modèle de système dynamique hybride, il est donc très difficile de comparer les résultats issus d'une simulation numérique avec ceux obtenus par calcul direct des trajectoires.

Nous allons présenter par la suite les principales précautions à prendre pour assurer la fidélité de la simulation par rapport à la modélisation.

3.3 Validation ou vérification fonctionnelle

Lorsque la simulation sert à valider un modèle en vue de l'aide à la conception d'un système dynamique hybride ou à prévoir son fonctionnement, la validation ne pose pas de problème particulier: la comparaison des résultats de simulation avec ce qu'attend le spécialiste ou avec un comportement connu de la partie opérative suffit.

Le problème devient plus délicat lorsqu'il s'agit de valider ou de vérifier la conception du dispositif de commande d'un SDH. Dans le but de discriminer l'origine des erreurs, et donc de faciliter leur localisation et leur correction, la validation de l'application consiste à vérifier le dispositif de commande conçu par rapport à la partie opérative concernée. L'absence d'outils formels de synthèse des commandes et de prévision des performances en boucle fermée pour les SDH rend nécessaire cette simulation de la partie opérative et de la commande qui a été élaborée.

En effet, la modélisation d'un SDH et la conception du dispositif de commande associé, aboutissent généralement à un ensemble de modules de régulation, continus ou échantillonnés, et de commandes séquentielles conçues séparément. Avant de passer à la réalisation de ce dispositif, il est impératif de s'assurer :

- de la validité de ces modules, ainsi que de leurs interactions,
- de la conformité de tous les comportements possibles de l'ensemble (partie opérative, commande) par rapport aux attentes (spécifiées ou non) du spécialiste du processus,
- du réglage des régulateurs: période d'échantillonnage, gains, paramètres de conception de haut niveau dans le cas où l'on utilise des techniques de commande avancée,
- de la validité des choix de certains paramètres,

-des performances que l'on peut attendre en fonction de ces paramètres (précision, dynamiques, robustesse, etc.).

Le deuxième point est particulièrement délicat. Il découle du fait que, même pour des installations de complexité relativement réduite, processus électro-mécaniques par exemple, il n'est pas rare que le spécialiste soit dans l'impossibilité d'envisager, et donc de spécifier, tous les comportements correspondant à tous les états possibles de la commande. Il se contente de définir un comportement global : le passage du mode de fonctionnement X au mode Y doit toujours se faire en respectant la condition $\alpha < \alpha_{\max}$ sur la grandeur continue α , par exemple. Si l'étude révèle que la passage $X \rightarrow Y$ peut se faire dans cinquante configurations différentes d'événements et/ou de conditions, il faudra que la simulation permette au spécialiste de vérifier que les algorithmes conçus donnent bien le comportement global attendu dans tous les cas. Ce type de vérification implique que l'on est capable de faire passer la simulation par tous les scénarios possibles de la commande, même si la probabilité de passage est très faible, ce qui exclut des essais statistiques de type Monte Carlo [Ibrahim ,93].

3.4 Architecture générale d'un simulateur dynamique hybride

L'architecture générale d'un simulateur dynamique hybride est décrite sur la figure 2.27

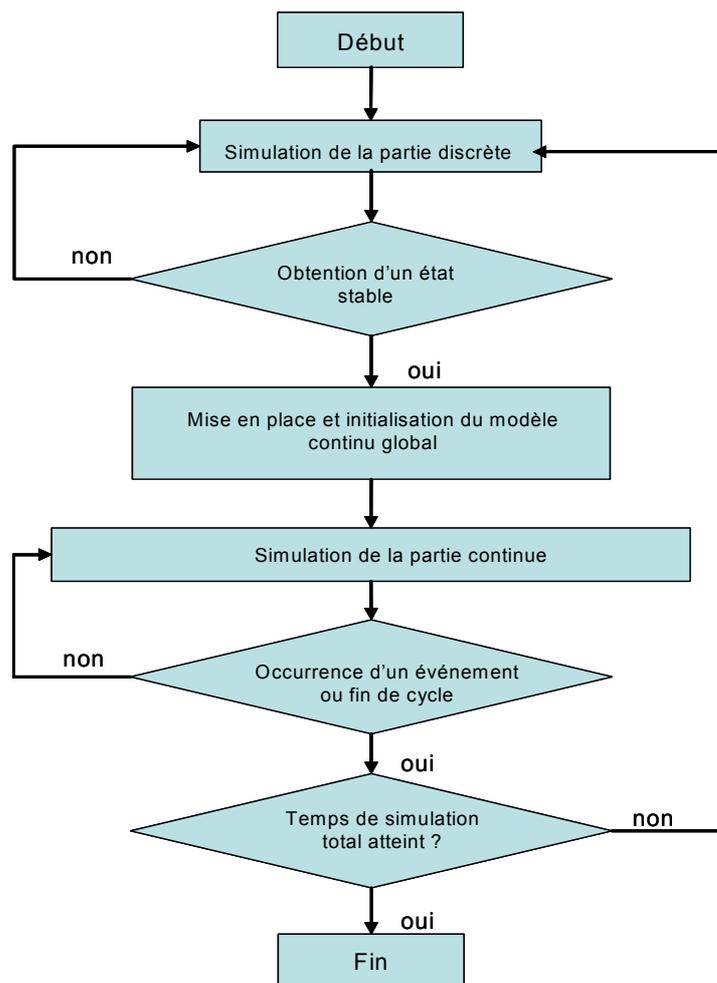


Figure 2.7 Architecture générale d'un simulateur dynamique hybride

La simulation dynamique hybride intègre plusieurs étapes : la simulation de la partie discrète, la simulation de la partie continue et la mise en place du modèle continu global.

3.4.1 Simulation de la partie discrète

La simulation de la partie discrète consiste à jouer le modèle discret : les marques se déplacent sur le graphe en franchissant les transitions jusqu'à l'obtention d'un état stable (c'est à dire jusqu'à ce que plus aucune transition ne puisse être franchie).

3.4.2 Simulation de la partie continue

La simulation de la partie continue consiste à résoudre un système d'équations composé traditionnellement d'équations différentielles ordinaires et partielles et d'équations algébriques.

Plusieurs méthodes de résolution existent : les méthodes multi-pas dont la méthode BDF (Backward Differentiation Formulas) et la méthode de Gear (1971) (méthode implicite), et les méthodes à un pas comme la méthode IRK (Runge Kutta Implicite) qui est particulièrement efficace dans le cas des systèmes caractérisés par de fréquentes discontinuités [Brenan et al.,89].

Notons par ailleurs que pour pouvoir utiliser ces méthodes en simulation continue, il doit exister un mécanisme permettant de stopper l'intégration lorsqu'un événement survient. Deux types d'événements existent : les événements temporels pour lesquels la date d'occurrence est prédéfinie et les événements d'états pour lesquels la date d'occurrence n'est pas connue a priori puisqu'ils se déclenchent lorsque la valeur d'une variable d'état franchit un seuil donné. Une fois l'événement détecté, il est alors nécessaire de le localiser temporellement de manière précise.

3.4.3 Mise en place du modèle continu global

Le modèle continu global est obtenu par concaténation des sous-modèles actifs du système. La structure qui le caractérise est par conséquent souvent variable, puisque le nombre d'équations et le nombre de variables sont amenés à changer au cours de la simulation. Les intégrateurs doivent alors être dotés d'algorithmes de vérification et d'analyse du système d'équations afin de réduire l'index du problème, vérifier la cohérence du modèle (nombre d'équations égal au nombre de variables), de permettre l'initialisation ou la réinitialisation du modèle, etc.

3.5 Classification des outils de simulation

Devant la complexité grandissante des procédés, l'IPAO (Ingénierie des Procédés Assistée par Ordinateurs) s'est fortement développée et a donné naissance à de nombreux progiciels (ou «CAPE Tools » en anglais), intégrant les fonctions de conception, d'exploitation et de conduite.

Ainsi, face à un problème de modélisation, de simulation ou d'optimisation et selon l'acteur concerné, plusieurs types d'outils sont actuellement proposés [Moyses, 00] (figure 2.8.)

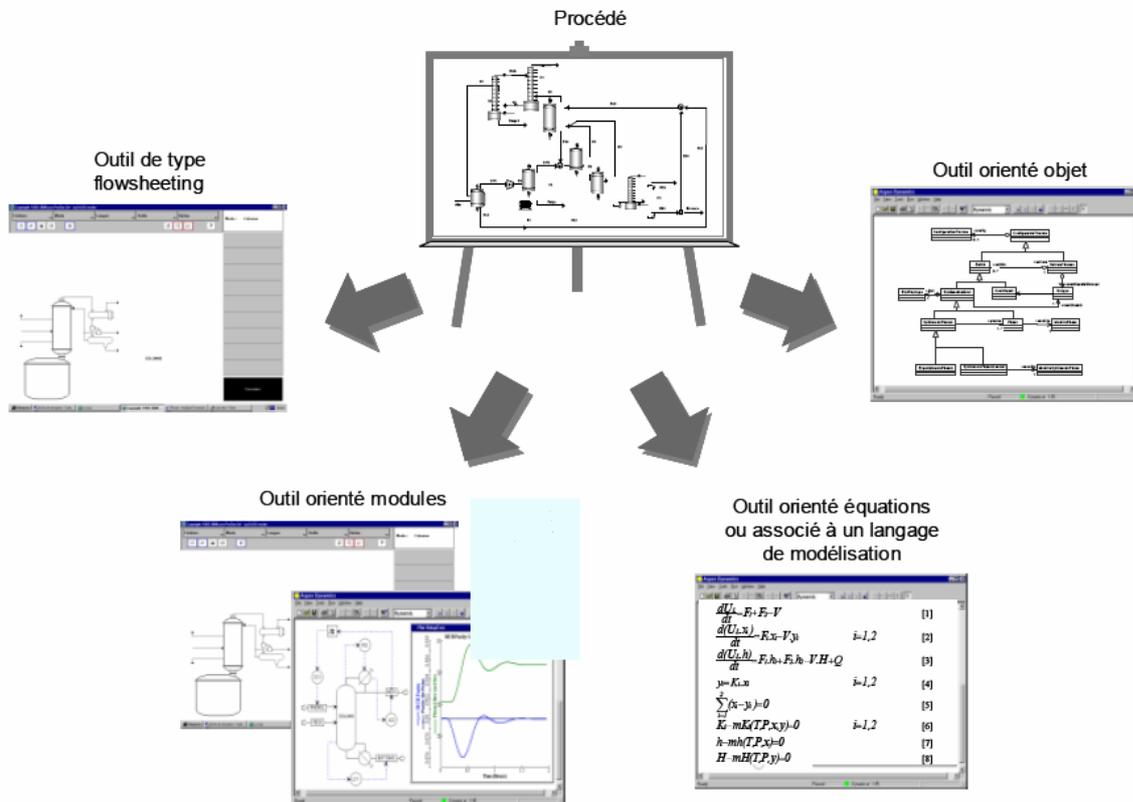


Figure 2.8 Outils de simulation

Un premier type d'outil rencontré est basé sur une *approche de type «flowsheeting»*. L'ingénieur de procédé construit sans programmer les équations décrivant son procédé en connectant des appareils prédéfinis regroupés dans des bibliothèques dédiées au génie des procédés. Ce type de logiciel est souvent convivial et facilement exploitable. Son utilisation se fait soit à travers une interface graphique de type «drag and drop» («glisser et poser»), soit par un langage dédié, non extensible. Chaque procédé est représenté par des blocs prédéfinis connectés entre eux pour des échanges d'énergie, de matière et d'information. Ces connexions se font par des interfaces et des courants standardisés. Le simulateur *Prosim Plus*, par exemple, adopte ce type d'approche.

Avec l'approche « flowsheeting », l'ingénieur dispose ainsi d'un environnement de simulation dont le but est d'induire un travail d'implémentation réduit grâce à l'intégration de l'ensemble des outils de modélisation, simulation et analyse au sein de la même structure. Cependant, un environnement intégré complet qui répond à tous les besoins n'existe pas, tant la complexité et l'éventail des systèmes et des savoir-faire requis est vaste. Par ailleurs, dans cette approche, l'acteur se heurte à des systèmes propriétaires souvent fermés et monolithiques, offrant peu de possibilités d'extension durant l'analyse de systèmes spécifiques. En effet, les modèles sont pré-codés et l'acteur ne peut utiliser que les appareils proposés dans la bibliothèque. Pour pallier cette limitation, un groupe d'industriels, de

fournisseurs de logiciels et d'universitaires se sont regroupés et ont travaillé conjointement pendant plusieurs années sur la définition d'un standard international CAPE-OPEN [Belaud et Pons, 02 ; Gani et Braunschweig, 02]. Le troisième type d'outil s'adresse plutôt au modélisateur de procédés qui préfère manipuler des *équations et des variables* ou un *langage de modélisation*. Les langages de modélisation permettent en effet à l'utilisateur de coder ses propres modèles à partir de quelques concepts de base (modèles, équations, variables, etc.). Ils sont extrêmement flexibles et plus généraux car non dédiés à un domaine particulier. Le langage est extensible, c'est-à-dire que l'utilisateur peut définir ses propres types par héritage ou agrégation. La plupart des langages de modélisation sont généraux, c'est-à-dire qu'ils permettent de représenter et simuler tous les systèmes physiques. Les langages *Omola* [Andersson, 99], *Ascend* [Allan, 97]) et *gProms* [Barton et Pantelides, 94] en sont des exemples. Actuellement, un groupe de chercheurs et d'industriels mettent au point un langage de modélisation unifié, *Modelica* [Elmqvist *et al.*,99], censé couvrir, à terme, de nombreux domaines physiques. Enfin, un dernier type d'outil repose sur une *approche orientée objet* et s'adresse au développeur pour qui les concepts orientés objet lui sont familiers. Ce dernier manipule des objets élémentaires et génériques qui ne sont pas nécessairement des appareils réels, mais qui peuvent également être des entités abstraites et génériques. La modélisation qui en résulte est alors intuitive et très flexible. Les objets sont en effet plus fins que les modules et le développeur reste capable de manipuler les variables et les équations qui les représentent. La philosophie objet permet également de générer des entités facilement réutilisables pour des développements futurs et conduit à long terme à des temps et coûts de développement réduits. Cette approche s'avère très efficace pour les sociétés d'ingénierie dont le travail est de créer des simulateurs dédiés à des unités spécifiques. En effet, les développeurs recodent souvent entièrement les unités [Mardon,89] et l'utilisation d'outils offrant une réutilisabilité maximale de l'existant est alors très intéressante. Enfin, soulignons que les objets déjà créés peuvent être manipulés sous forme d'une bibliothèque ou via une interface graphique, définissant alors un environnement de simulation.

C'est dans ce cadre que s'est inscrit le développement de la plate-forme *PrODHyS*. Elle s'appuie sur la philosophie objet et fournit un ensemble de composants généraux, extensibles et réutilisables afin de construire des éléments plus spécifiques.

4 CONCLUSION

Ce chapitre a proposé un tour d'horizon général sur les systèmes dynamiques hybrides. Il a, dans un premier temps, positionner ce domaine d'étude par rapport aux systèmes à dynamique continue et discrète. Il a ensuite introduit les aspects « modélisation » et a proposé, dans ce contexte, un ensemble de formalismes permettant de les représenter. Nous avons vu que ceux-ci résultaient de trois approches de modélisation : les extensions des modèles continus, les extensions des modèles discrets et enfin, les modèles mixtes. Ensuite, ce chapitre s'est focalisé sur les aspects « simulation » relatifs aux systèmes dynamiques hybrides. Il s'est attaché, dans un premier temps, à décrire le fonctionnement d'un noyau de simulation hybride. Il a ensuite poursuivi en dressant une liste d'outils de simulation couramment rencontrés; L'état de l'art sur les SDH proposé dans ce chapitre nous permet ainsi d'aborder plus aisément la problématique de notre étude.

Chapitre 3 : La plate forme de simulation PrODHyS

1 Introduction

PrODHyS (Process Object Dynamic Hybrid Simulator) résulte de l'unification de travaux de recherche réalisés depuis une dizaine d'années au sein du Laboratoire de Génie Chimique (LGC). De ce fait, son développement a conduit à l'intégration de différents modules dédiés à des fonctions particulières. Pour cette raison, ce chapitre se propose d'en décrire quelques-uns, tels qu'ils existaient avant le démarrage de ces travaux. Pour cela, les outils logiciels de conception et de développement utilisés sont tout d'abord introduits. Ensuite, une présentation générale de l'architecture de PrODHyS est proposée. Enfin, une description plus exhaustive des trois principaux modules est réalisée ; ils correspondent respectivement :

- au noyau numérique (avec une représentation objet des modèles mathématiques),
- à la base de données thermodynamiques (avec une représentation objet de la matière),
- et à une bibliothèque d'appareils décrits par des modèles dynamiques continus.

Il conclue avec un bilan de cette étude et met en évidence les principaux verrous qui ont conduit à la prise en compte des aspects hybrides.

2 Processus de développement logiciel

La création de l'environnement *PrODHyS* suit un processus de développement industriel unifié s'appuyant sur le formalisme *UML*. De façon générale, un tel processus répond à différentes caractéristiques:

- Il est incrémental. La définition d'incrément de réalisation est en effet la meilleure pratique de gestion des risques d'ordre à la fois technique et fonctionnel. Un projet qui ne produit rien d'exécutable dans les neuf mois court un risque majeur d'échec.
- Il est piloté par les risques. Dans ce cadre, les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité. Une première cause d'échec est liée à l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles ; une seconde est liée à l'inadéquation du développement aux besoins des utilisateurs.
- Il est construit autour de la création et de la maintenance d'un modèle, plutôt que de la production intensive de documents. Le volume d'informations de ce modèle nécessite une

organisation stricte qui présente les différents points de vue du logiciel à différents degrés d'abstraction.

- Il est itératif. Chaque itération porte sur des degrés d'abstraction de plus en plus précis et permet de produire des traces nécessaires au contrôle du changement.
- Il est orienté composants. Tant au niveau modélisation que production, c'est une garantie de souplesse pour le modèle lui-même et le logiciel qu'il représente.
- Il est orienté utilisateurs car la spécification et la conception sont construites à partir des modes d'utilisation attendus par les acteurs du système.

Au cours du cycle de vie itératif et incrémental, le système est développé par une série d'itérations jusqu'au système final. Chaque itération englobe une ou plusieurs phases du processus (modélisation, besoins, analyse, conception, implémentation, tests et déploiement). Ce type de cycle permet d'atténuer les risques, qui sont définis, évalués, classés par priorités et revus à chaque itération .

La gestion de la conception avec *UML* permet donc de faire des itérations entre la conception et le développement (figure 3.1). Dans le cas de PrODHyS, la conception et l'analyse sont réalisées à l'aide du langage *UML* et l'implémentation se fait en *C++*. Le passage *d'UML* au *C++* se fait par génération automatique de code et le passage inverse se fait par rétro ingénierie.

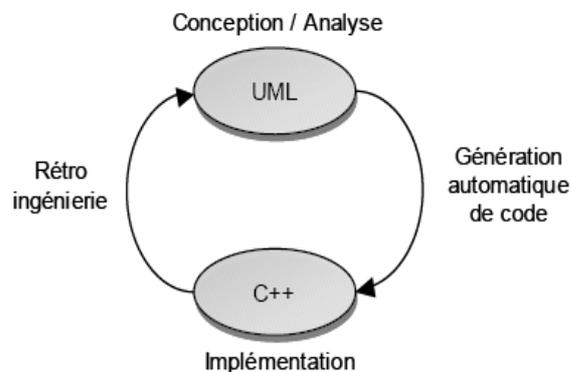


Figure 3.1 Conception itérative

Le développement de l'environnement de simulation a été réalisé à l'aide de deux logiciels, *Rational Rose* et *Visual Studio/C++*.

Rational Rose est un logiciel qui permet de mettre en œuvre le langage *UML* à l'aide d'une interface graphique.. En outre, un des intérêts majeurs de *Rational Rose* réside dans sa capacité à générer automatiquement du code *C++* à partir de cette description graphique.

Lorsque l'utilisateur se limite à concevoir conceptuellement ses classes, ce logiciel reste simple d'utilisation. En revanche, la génération de code nécessite une connaissance approfondie du logiciel et du langage. En effet, elle induit la prise en compte de nombreux paramètres (concernant le projet, le langage généré, les diagrammes, les relations, etc.) qui ne sont pas toujours évidents à définir. Toutefois, une fois ces paramètres fixés, *Rational Rose* facilite la gestion du projet en permettant un découpage du modèle complet en plusieurs sous-

modèles sur lesquels des utilisateurs différents peuvent travailler indépendamment et parallèlement. Tous les fichiers issus de *Rational Rose* sont ensuite traités grâce à *Visual Studio/C++*, qui permet la compilation, le débogage et l'exécution de programmes C++ au travers d'une interface conviviale.

3 Architecture globale de la plate-forme PrODHyS

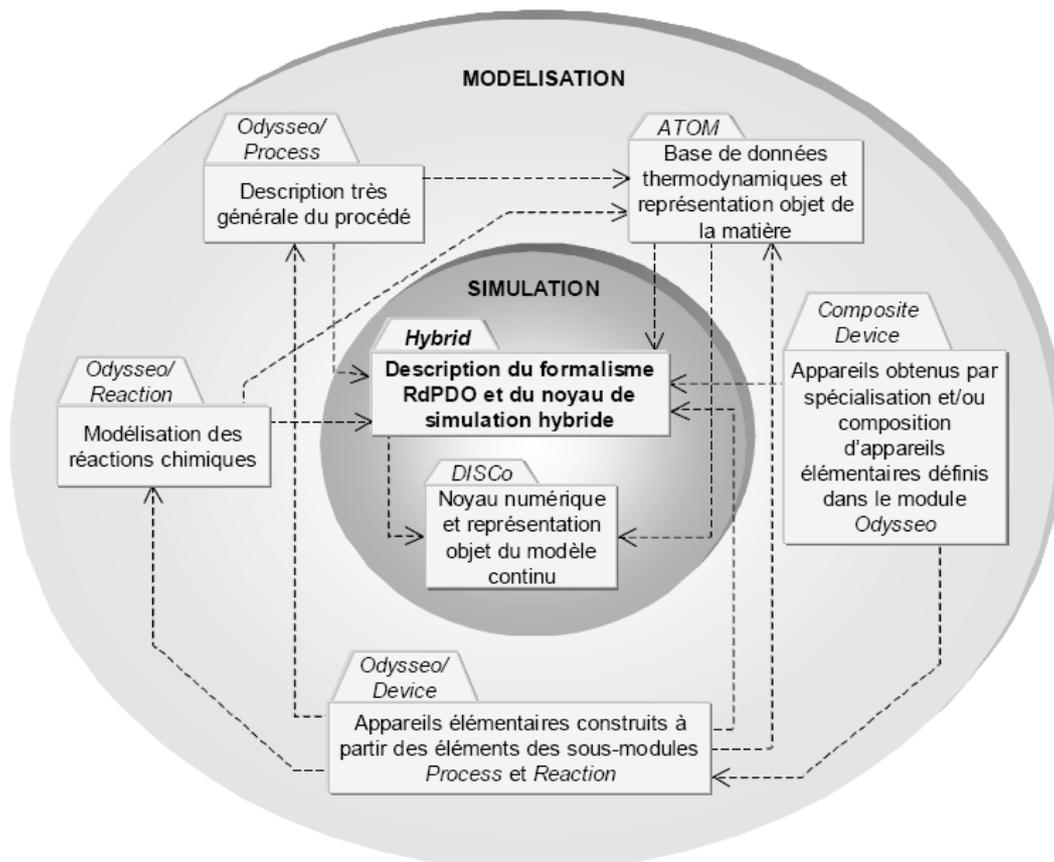


Figure 3.2 Structure logicielle de la plate-forme PrODHyS

La plate-forme PrODHyS résulte de l'unification de travaux de recherche réalisés depuis une dizaine d'années au sein du *Laboratoire de Génie Chimique*. Son développement a aussi profité de la longue expérience acquise au sein du département PSE dans le domaine de la modélisation et de la simulation des procédés. Avec l'ajout du module nommé *Hybrid*, PrODHyS est actuellement composé de sept modules. Ceux-ci sont présentés de façon schématique sur la figure 3.2. [Perret .03]

Cette figure met notamment en évidence les dépendances (flèches en pointillées) existant entre eux et fait apparaître une structure comprenant deux couches logicielles :

- La couche la plus interne correspond au *noyau de simulation* de la plate-forme. Elle fournit les éléments de base permettant de modéliser et simuler n'importe quel système dynamique. Aujourd'hui, cette couche comprend :

- Le module *DISCO* [Sargousse, 99 ; Le Lann, 99] qui constitue le noyau numérique du système ; il fournit un ensemble de solveurs et d'intégrateurs (EDA, EANL) et permet une représentation objet des modèles mathématiques continus.
- Le module *Hybrid* qui contient l'ensemble des classes nécessaires à la description du formalisme RdPDO ainsi que du noyau de simulation hybride.
- La couche supérieure correspond à un ensemble de classes dédiées à la modélisation des procédés. La couche «modélisation» s'appuie sur la couche «simulation» et fournit un ensemble d'entités «prêtes à l'emploi» qui peuvent être exploitées par un utilisateur désireux de construire son programme de simulation. Cette couche comprend :
 - Le module *ATOM* [Jourda, 96; Jourda *et al.*, 96] qui constitue la base de données thermodynamiques du système ; il s'appuie sur une représentation objet de la matière et permet le calcul des propriétés thermodynamiques.
 - Le module *Odysseo* [Moyse, 00] qui regroupe les entités élémentaires et génériques permettant de modéliser un procédé. Il se décompose sous la forme de trois sous-modules :
 - * Le sous-module *Process* qui regroupe un ensemble de classes, souvent abstraites, correspondant à une description très générale du procédé,
 - *Le sous-module *Reaction* qui permet la modélisation des réactions chimiques,
 - *Le sous-module *Device* qui rassemble les appareils élémentaires «concrets» du procédé construits à partir des éléments des deux sous-modules précédents.
 - Le module *CompositeDevice* qui regroupe des appareils obtenus par spécialisation et/ou composition d'appareils élémentaires définis dans le module *Odysseo*. Ils correspondent aux classes terminales figurant dans la hiérarchie des objets définis dans *PrODHyS*.

Le découplage des couches « simulation» et « modélisation» présente plusieurs intérêts qui sont:

- de créer des plate-formes dédiées à des domaines d'application divers (mécanique, électronique, etc.) en développant simplement la couche « modélisation» adéquate,
- de permettre l'accès à cette plate-forme à différents types d'utilisateurs:
 - Un développeur d'applications orientées objet utilisera directement la couche «simulation» pour créer les composants de base dans un domaine particulier (composants électroniques, pièces mécaniques, éléments de procédés, etc.),
 - Un modélisateur sera plutôt intéressé par mettre en place des systèmes complexes à partir des composants de base en faisant abstraction du comportement interne de ces derniers. Dans notre cas, cela correspond au développement du module *CompositeDevice* qui contient les appareils complexes spécifiques.
 - Enfin, une dernière catégorie d'utilisateur sera désireuse de réaliser une étude sans effectuer de développement informatique et en utilisant directement des éléments autonomes au travers d'une interface graphique. Cette dernière couche logicielle est en cours de développement dans un module nommé *PrODHySGUI*.

4 La couche modélisation

4.1 Modélisation de la matière

ATOM (Applied Thermodynamics Object-oriented Model) est issu d'un travail de thèse réalisé au sein du *Laboratoire de Génie Chimique* (Jourda, 1996). Cette bibliothèque a été améliorée et recodée en C++ au sein de la société *ProSim S.A.*. *ATOM* constitue la base de données thermodynamiques du système. Elle s'appuie sur une représentation objet de la matière.

La modélisation de la matière passe par la définition d'un univers thermodynamique, de constituants, de phases et de systèmes de phases. Le paragraphe suivant se propose d'illustrer leur utilisation à travers un exemple [Moyse, 2000].

Soit un système liquide/vapeur comportant trois constituants A, B et C. La modélisation de ce système nécessite la création d'objets. Ces derniers, associés aux relations qui les caractérisent, sont représentés sur la figure 3.3.

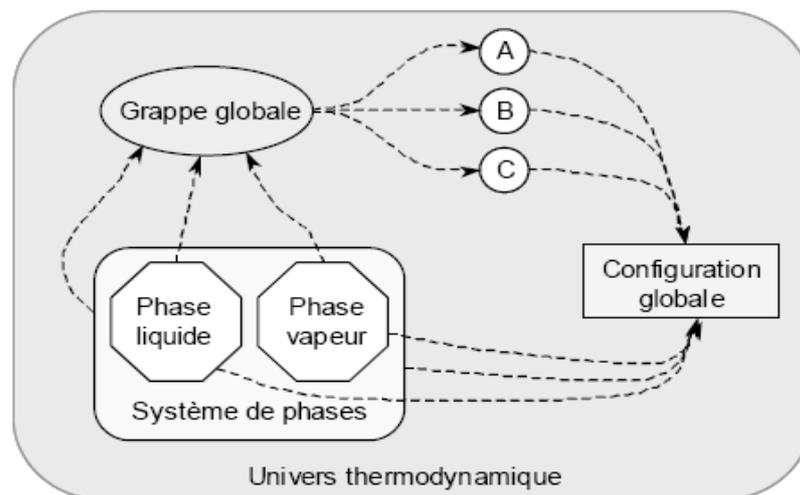


Figure 3.3 Création d'un univers thermodynamique

Dans un premier temps, l'univers thermodynamique, instancié en début de simulation, crée automatiquement une *grappe globale* de constituants et une *configuration globale*. L'utilisateur peut alors modifier les options par défaut de la configuration thermodynamique et choisir les modèles de calcul des propriétés thermodynamiques.

L'utilisateur crée ensuite trois objets A, B et C à travers l'interface de l'univers (A = éthanol, B = méthanol, C = eau, par exemple). La grappe globale possède alors automatiquement une référence vers ceux-ci. La configuration prise par défaut par les constituants est la configuration globale.

L'utilisateur crée enfin un système de phases à l'état liquide/vapeur (un calcul d'équilibre doit vérifier cet état maximal). Les deux objets Phase (un pour le liquide et l'autre pour la vapeur) sont alors automatiquement créés. Ces entités possèdent une référence d'une part, vers la grappe globale et d'autre part, vers la configuration globale.

4.2 Modélisation des appareils

La bibliothèque *Odysseo* (*Objected-oriented Dynamic Simulation Software EnvirOnment*) est issue d'un travail de thèse réalisée au sein du *Laboratoire de Génie Chimique* [Moyse,00]. *Odysseo* regroupe les entités élémentaires et génériques permettant de modéliser un procédé de génie chimique en vue de la *simulation dynamique continue*. *Odysseo* est divisé en trois grandes parties :

- Les classes générales, souvent abstraites, permettent une description générale des procédés afin d'en cerner les caractéristiques communes ; elles sont contenues dans le sous-module *Process*.
- Les classes qui permettent la modélisation et la simulation des réactions chimiques appartiennent au sous-module *Reaction*.
- Enfin, les classes plus spécifiques héritant des classes générales et permettant de décrire des appareils « concrets » sont contenues dans le sous-module *Device*.

Le procédé est décrit par le sous-module *Process*. Ce dernier regroupe les principaux concepts de description de la topologie qui étendent à la simulation dynamique, les idées développées par L. Jourda dans le cadre du régime permanent [Jourda, 1996]. La description de la topologie se fait en termes de *ports* et *d'appareils*.

4.2.1 Ports définis dans PrODHyS

Selon la fonctionnalité et la topologie des ports, six types de ports ont été identifiés : les ports de communication d'entrée, les ports de communication de sortie, les ports d'énergie d'entrée, les ports d'énergie de sortie, les ports de matière d'entrée et les ports de matière de sortie. Ils sont schématisés sur la figure 3.4.

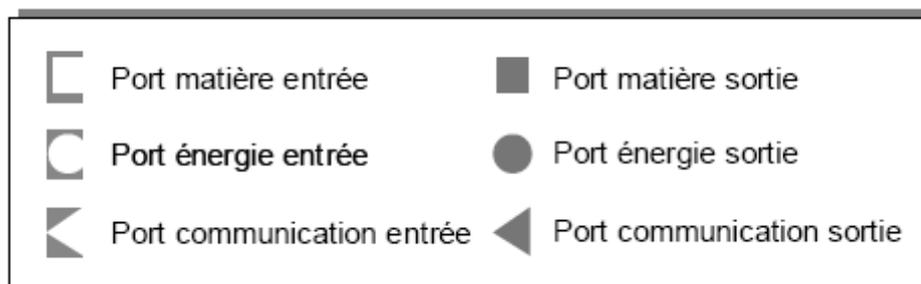


Figure 3.4 Représentation des ports définis dans PrODHyS

4.2.2 Les appareils élémentaires

Un appareil élémentaire est la plus petite entité possible dans le module *Odysseo*. Il définit la granularité de décomposition du procédé. Un appareil élémentaire dispose d'un ensemble de ports matière, énergie et communication entrants et sortants dont il est responsable. La matière qui circule dans le procédé est définie au niveau de l'appareil comme un objet ; il s'agit d'un système de phases concret défini dans *ATOM* mais auquel des fonctionnalités ont été ajoutées afin de décrire les réactions chimiques.

4.2.3 Les appareils composés

Un appareil plus complexe résulte de la réunion d'un certain nombre d'appareils élémentaires ou composés ; cela signifie qu'il peut contenir d'autres appareils composés. Le niveau de décomposition est donc infini. Un appareil composé n'est responsable ni de la création, ni de la destruction des appareils le constituant ; il s'agit simplement d'un regroupement d'appareils. Un appareil composé possède des ports mais ces ports ne lui appartiennent pas. Les ports d'un appareil composé sont les ports des appareils le constituant non connectés entre eux (cf. figure 3.5).

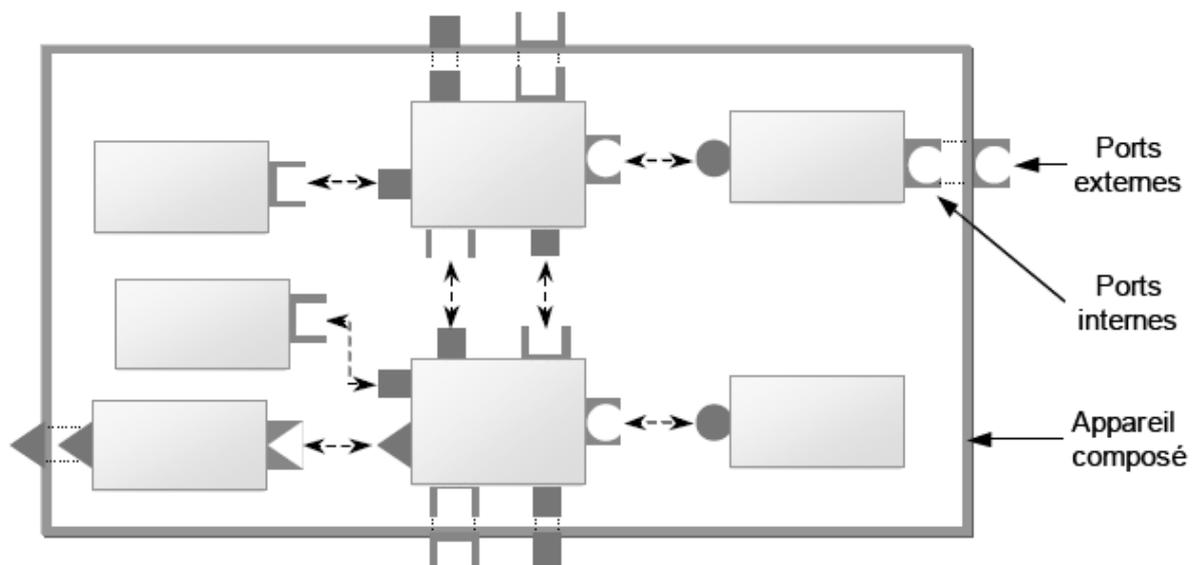


Figure 3.5 Ports d'un appareil composé

4.2.4 Les appareils spécifiques

Les classes développées dans le sous-module *Device* sont des classes concrètes qui représentent des appareils réels. Un des intérêts de la modélisation orientée objet est de définir un petit nombre de classes qui permettent, par leur combinaison, de représenter un grand nombre d'appareils. La réflexion sur les appareils s'est posée en terme de fonctionnalité au sein du procédé. Deux fonctionnalités principales ont été identifiées pour la simulation dynamique des procédés :

- Les *opérations unitaires* dont font partie les bacs, les colonnes, les échangeurs, etc. ; les simulateurs dédiés offrent souvent un large choix de ce type d'appareils.
- Les *systèmes hydrauliques*; ils jouent un rôle important dans la simulation dynamique car ils permettent de tenir compte de la mise en circulation des fluides et des phénomènes de pression trop souvent négligés et dont l'aspect transitoire est essentiel.

Un procédé doit associer ces deux types d'appareils. Les opérations unitaires sont le siège de transformations de matière et d'énergie et constituent des nœuds de pression ; ils subissent les flux à leurs frontières et les phénomènes hydrodynamiques internes fixent les

pressions. Les appareils hydrauliques imposent les flux entre les nœuds de pression. Ce sont des jonctions hydrauliques.

4.2.4.1 Les opérations unitaires

Les opérations désignées sous le terme d'opérations unitaires sont les appareils dans lesquels les transformations de matière ont lieu. Parmi celles-ci, on distingue:

- Les *appareils utilitaires* qui représentent des concepts plus que des appareils réels ; ils décrivent l'environnement et les alimentations de matière et d'énergie,
- Les *bacs* qui peuvent définir aussi bien les bacs élémentaires que les bacs composés (bacs avec système thermique intégré par exemple),
- Les *colonnes*, qui n'existent pas en tant que telles, mais dont les éléments les composant (bouilleur, plateau avec et sans garnissage, condenseur) peuvent être associés à loisir,
- Les *échangeurs* qui représentent des systèmes thermiques avec possibilité d'ajouter des réactions chimiques ; parmi eux, on distingue les échangeurs simples, les cellules de transfert thermique et les échangeurs tubulaires.

4.2.4.2 Les appareils hydraulique

Les appareils hydrauliques régissent la circulation des fluides entre les opérations unitaires ; ils définissent ainsi les débits aux ports de celles-ci. Dans ce contexte, la notion de jonction hydraulique a été introduite. Elle définit un système monophasique qui peut être élémentaire ou composé. Les jonctions élémentaires relient directement deux opérations unitaires et sont caractérisées par une configuration alors que les lignes hydrauliques sont des compositions de jonctions élémentaires. Plusieurs types de jonctions élémentaires ont été définis :

- Les *conduites* qui peuvent être droites ou caractérisées par des changements de section brusques ou par des coudes,
- Les *vannes* caractérisées par un coefficient d'ouverture,
- Les *pompes*.

5 La couche simulation

5.1 Le noyau numérique

La bibliothèque *DISCO* (*Do Integrate by Software Component*) résulte de nombreux travaux réalisés au sein de l'équipe *AFP* (*Analyse Fonctionnelle des Procédés*) du *Laboratoire de Génie Chimique* [Sargousse, 1999; Albet, 1992; Le Lann, 1999]. Comme l'indique la figure 3.2, *DISCO* constitue le noyau numérique du système ; il fournit un ensemble de solveurs et d'intégrateurs (EDA, EANL, etc.) et permet une représentation objet des modèles mathématiques continus.

5.1.1 Structure d'un modèle en dynamique

La modélisation et la simulation dynamique robuste des procédés conduisent souvent à un jeu mixte d'équations différentielles ordinaires (EDO) ou partielles (EDP) à un ordre

quelconque couplées avec des équations algébriques non linéaires (EANL). En effet, si l'on considère principalement les systèmes algébro-différentiels EDA (EDO+EANL), il apparaît que les équations différentielles ordinaires proviennent de l'application des lois de bilan de conservation de la matière, de l'énergie et de la quantité de mouvement établies en régime transitoire, tandis que les équations algébriques non-linéaires se dérivent des relations constitutives ou phénoménologiques qui traduisent, par exemple, la thermodynamique des systèmes, les équations d'équilibre physico-chimique, les conditions hydrodynamiques voire technologiques ou les équations de spécification de conception ou d'opérabilité.

La structure d'un modèle en dynamique peut être représentée sous la forme implicite générale suivante:

$$\begin{aligned} F(x^{(n)}, x, p, \theta) &= 0 \\ x_{\theta=0} &= x_0 \\ x_{\theta=0}^{(n)} &= x_0^{(n)} \end{aligned} \quad 3.1$$

Avec

F : jeu d'équations (différentielles et/ou algébriques),

x : ensemble des inconnues du système,

$x^{(n)}$: ensemble des dérivées à l'ordre n des inconnues par rapport à la variable indépendante,

p : ensemble des paramètres opératoires du système,

θ : variable indépendante (généralement le temps).

Le mode de résolution de tels systèmes s'appuie sur une stratégie de résolution globale nécessitant souvent deux informations supplémentaires :

$\frac{\partial F}{\partial x}$: matrice des dérivées des équations par rapport aux inconnues,

$\frac{\partial F}{\partial x^{(n)}}$: matrice des dérivées des équations par rapport à leurs dérivées.

5.1.2 Résolution

Une fois que le modèle mathématique a été mis en place, il doit faire l'objet d'une résolution mathématique. Pour cela, trois solveurs ont été implantés dans la bibliothèque *DISCO* :

- Un solveur linéaire pour résoudre les systèmes de type $AX=B$; ce solveur a été écrit en *Fortran 77* et encapsulé en *C++*. Le solveur linéaire a été choisi dans une bibliothèque de méthodes numériques commerciale (UMFPACK).

- Un solveur de systèmes d'équations algébriques non linéaires de la forme $F(X) = 0$; ce solveur, implémenté dans sa totalité en *C++*, est très important en génie chimique où les systèmes d'équations sont rarement linéaires.

- Un solveur de systèmes algébro-différentiels. Comme le solveur linéaire, ce solveur a été initialement développé en *Fortran 77* ; avant d'être introduit dans la bibliothèque, le code a

été remanié et traduit en C par le compilateur *F2C*. Il a été conçu dans le but de résoudre des systèmes algébro-différentiels, creux et de grande taille, en vue d'une simulation dynamique robuste. Le solveur est basé sur la méthode de *Gear* [Gear, 1971]. Ses principales qualités sont sa grande robustesse, sa capacité à passer les discontinuités, son aptitude à traiter des systèmes de grande taille, un algorithme de réduction d'index et un calcul automatique de conditions initiales cohérentes. Une autre de ses particularités est de posséder un système de détection d'événements. Ce système est basé sur l'algorithme de l'Illinois (Moler, 1998) et permet de stopper l'intégration dès qu'une fonction utilisant les inconnues du système atteint une valeur nulle [Le Lann ;99].

Actuellement, la bibliothèque *DISCO* gère les équations différentielles algébriques mais ne permet pas encore de traiter les équations différentielles partielles.

5.2 Le module hybride

Le module *Hybrid* se trouve au cœur du noyau de calcul de *PrODHyS* et contient l'ensemble des classes chargées de la gestion du déroulement de la simulation. Il repose sur un formalisme RdPDO (réseau de Petri Différentiel à Objets)

5.2.1 Sémantique et règles d'évolution du modèle RdPDO

Le terme « différentiel » indique la présence et l'association d'un système algébro-différentiel à certaines places du réseau. Le terme « objet » traduit l'utilisation de classes d'objets pour caractériser les jetons qui transitent sur le réseau ; il spécifie également le choix d'un jeu de jetons exprimé par des conditions associées aux transitions. Le réseau possède ainsi un ensemble d'éléments qui permet de décrire à la fois les aspects dynamiques et structurels du système à modéliser.

Ces éléments sont essentiellement:

- des classes d'objet représentant les différents composants du système à modéliser,
- des variables formelles typées par les classes,
- des jetons qui sont des instances de classes d'objet,
- des arcs typés dont certains peuvent être inhibiteurs,
- des places discrètes et différentielles,
- des transitions,
- des conditions et actions associées aux transitions.

La sémantique adoptée est représentée sur la figure 3.6. La suite de cette section a pour objectif de l'explicitier et de définir les règles d'évolution associées.

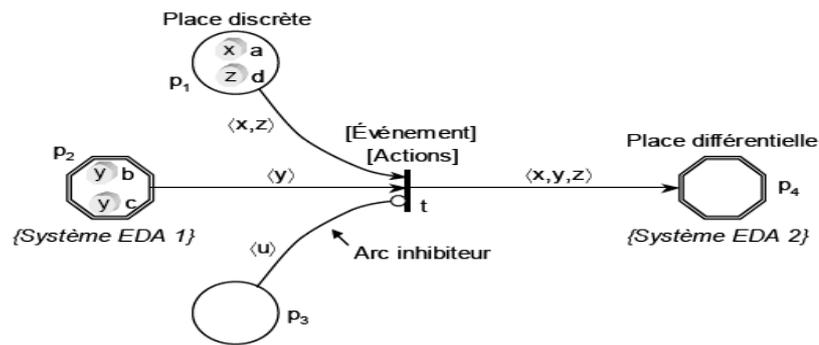


Figure 3.6 Sémantique du modèle

5.2.1.1 Les classes

Le but de l'intégration des concepts objet dans le modèle RdPDO est de faciliter la décomposition et la structuration du système à simuler, conduisant ainsi à une meilleure gestion de sa complexité. Par conséquent, un des premiers éléments caractéristique du modèle est l'ensemble $C = \{C_t, C_2, \dots, C_{n_c}\}$ de classes qui définit les objets constitutifs du système. L'établissement de cet ensemble de classes résulte généralement d'une analyse structurale du système considéré.

5.2.1.2 Les variables formelles

Les *variables formelles* sont définies dans l'ensemble $V = \{V_1, V_2, \dots, V_{n_v}\}$ et sont typées par des éléments de C . Le type de la variable formelle spécifie le type des jetons qui peuvent lui être substitués.

5.2.1.3 Les jetons

Le jeton correspond à une instance de classe d'objet constituée d'attributs (paramètres de l'objet, variables continues, réseau de Petri, etc.) et de méthodes. Ce mécanisme, avec individualisation et typage du jeton, permet de rendre le réseau plus compact, sans perte d'information.

5.2.1.4 Les arcs

L'arc, défini dans l'ensemble $A = \{A_1, A_2, \dots, A_{n_a}\}$ est le ligand entre la place et la transition. Afin d'assurer la cohérence du modèle, les arcs d'entrée et de sortie sont explicitement typés. En effet, ils portent des inscriptions permettant de spécifier les classes de jetons autorisés à transiter sur l'arc. S'agissant de classes, les propriétés relatives à l'héritage s'appliquent. En particulier, pourra aussi transiter par cet arc, toute classe de jeton héritière du type inscrit. Les inscriptions portées par les arcs sont des variables formelles. Ainsi, tout jeton autorisé à transiter par un arc se substitue à la variable formelle de même type.

Afin d'établir formellement les conditions qui permettent au jeton de transiter par un arc, la notion d'«*arc sensibilisé*» est introduite et la règle suivante est énoncée:

Règle 1 : Arc sensibilisé

Un arc est dit «sensibilisé» si et seulement si on peut substituer à chaque variable formelle, au moins un jeton distinct parmi les jetons de la place située en amont.

De plus, le réseau peut aussi contenir des arcs particuliers appelés *arcs inhibiteurs*. Au niveau de la représentation, ils se différencient par un cercle situé à l'une de ses extrémités (celle reliée à la transition). Les arcs inhibiteurs se définissent comme suit :

Un *arc inhibiteur* est un arc qui ne consomme pas et qui permet de tester l'absence de jeton sur la place située en amont.

L'arc inhibiteur est, par exemple, très bien adapté pour tester l'absence de stock dans une unité de production. Dans le cadre du formalisme RdPDO, l'arc inhibiteur permet de tester l'absence de jeton du (ou des) type(s) indiqué(s) sur l'arc.

Afin d'établir formellement les conditions qui permettent au jeton de transiter par un arc inhibiteur, la notion *d'arc inhibiteur sensibilisé* est introduite et la règle suivante est énoncée:

Règle .2 : Arc inhibiteur sensibilisé

Un *arc inhibiteur* est dit «sensibilisé» si et seulement si la place située en amont ne contient pas de jetons pouvant être substitués aux variables formelles associées à l'arc.

Afin d'illustrer les propos précédents, la figure 3.7 propose deux cas de figure illustrant les conditions de sensibilisation d'un arc « classique » et d'un arc inhibiteur,

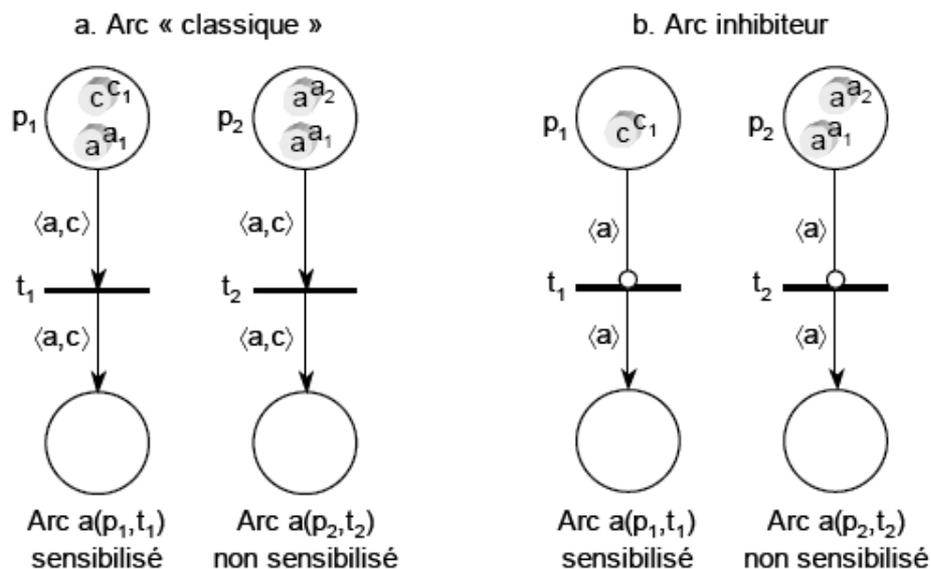


Figure 3.7 Sensibilisation des arcs « classiques » et des arcs inhibiteurs

Sur la figure 3.7.a, l'arc $a(p_1, t_1)$ est caractérisé par le n-uplet de variables formelles $\langle a, c \rangle$ et la place située en amont possède deux jetons a_1 et c_1 portant les mêmes inscriptions ; par conséquent, l'arc $a(p_1, t_1)$ est sensibilisé. En revanche, l'arc $a(p_2, t_2)$, caractérisé par le même n-uplet, n'est pas sensibilisé car la place p_2 ne contient pas de jeton de type c ; il est donc impossible de substituer un jeu de jetons au n-uplet de variables formelles $\langle a, c \rangle$.

Sur la figure 3.7.b, l'arc $a(p_1, t_1)$ est caractérisé par la variable formelle $\langle a \rangle$ et la place située en amont possède un seul jeton c_1 de type c : il est donc impossible de substituer un

jeton de la place p_1 à la variable formelle $\langle a \rangle$; l'arc inhibiteur $a(p_1, t_1)$ est par conséquent sensibilisé. Au contraire, la présence d'au moins un jeton de type a sur la place p_2 (jetons a_1 et a_2) rend l'arc inhibiteur $a(p_2, t_2)$ non sensibilisé.

5.2.1.5 Les places

Tout réseau de Petri est constitué d'un ensemble fini $P = \{P_1, P_2, \dots, P_n\}$ de places représentant les différents états du système. En outre, dans le modèle RdPDO, deux types d'états, correspondant à un fonctionnement spécifique, doivent être différenciés. De ce fait, deux types de places sont identifiés :

- des places dites *discrètes*,
- des places dites *différentielles*,

Par conséquent, l'ensemble P se décompose en deux sous-ensembles disjoints P_H et P_D tel que :

$$P = P_H \cup P_D$$

où

P_H est un ensemble fini de places différentielles,

P_D est un ensemble fini de places discrètes.

Dans le modèle RdPDO, les places sont représentées de la façon suivante :

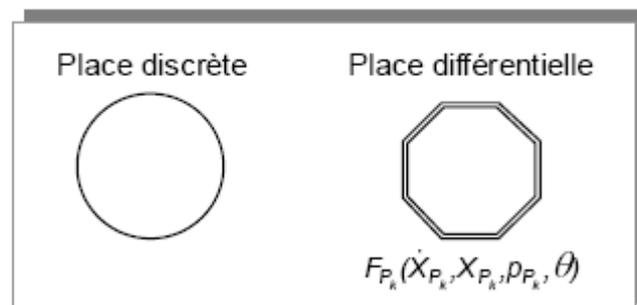


Figure 3.8 Représentation des places discrètes et différentielles

Une *place discrète* P_k représente un état dans lequel il n'y a pas d'évolution au cours du temps des variables continues du système. Elle est représentée par un simple cercle. Elle permet généralement de modéliser un contrôle, une commande, la disponibilité d'une ressource, etc.

Une *place différentielle* P_k est un état dans lequel il y a une évolution au cours du temps des variables continues du système. Cette évolution est régie par le système d'équations différentielles algébriques associé à la place P_k . Elle est représentée par deux octogones concentriques.

5.2.1.5.1 Les places temporisées

Il est souvent nécessaire de modéliser une temporisation. C'est le cas, par exemple, lorsque l'on souhaite imposer des temps d'alimentation (pour simuler des fautes dans notre cas), des durées réactionnelles ou des temps d'attente maximaux de certains produits (dans le domaine de l'agroalimentaire par exemple). Dans le formalisme RdPDO, ce type d'opération

se modélise par une place différentielle particulière nommée TimerPlace. Celle-ci contient d'une part, une temporisation d , et d'autre part, une équation différentielle algébrique du type :

$$\frac{d\theta_p}{d\theta} - 1 = 0 \quad (3.2)$$

où :

θ représente la variable indépendante, le temps,

θ_p représente le temps écoulé depuis le marquage de la place p (t_c).

Ce type de place rejoint la notion de *place temporisée* définie dans les réseaux de Petri p -temporisés. En effet, dans ces réseaux, une *durée* est associée à la place et le jeton est *indisponible* pendant toute la durée de la temporisation ; il est *réservé*.

Afin de clarifier ces propos, la figure 3.9 représente le modèle RdPDO utilisé pour représenter une place temporisée.

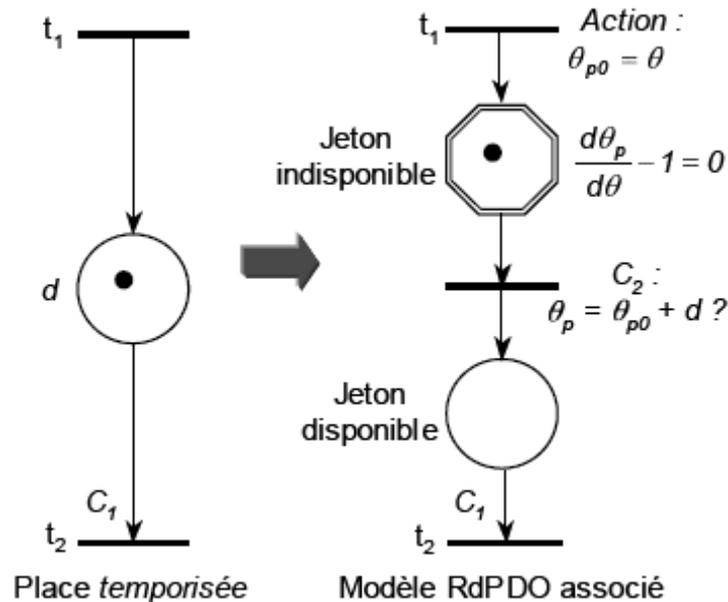


Figure 3.9 Modélisation des places temporisées selon le formalisme RDPDO

La figure 3.9 représente une place temporisée suivie d'une transition à laquelle est associée une condition particulière C_1 . Le modèle adopté pour décrire ce type de place consiste à la dissocier en une séquence place/transition/place. La première place est représentée par la classe TimerPlace et correspond à l'opération en cours, la transition située en aval marque la fin de l'opération et la dernière place correspond à une attente éventuelle (attente de l'occurrence de la condition C_1 par exemple). Un jeton mis dans la première place est indisponible pendant toute la durée de la temporisation. La condition C_2 détecte la fin de la temporisation et son occurrence libère le jeton. Ce n'est qu'à cet instant que la condition C_1 , associée à la transition t_2 , peut être alors testée.

5.3 LE GESTIONNAIRE DE SIMULATION

Afin d'implanter le formalisme RdPDO, le noyau du simulateur se décompose en trois modules : le *solveur discret*, le *solveur continu* et un *gestionnaire de simulation* qui gère les interactions entre les deux solveurs. Le gestionnaire de simulation est responsable de la simulation globale du système. Il donne à tour de rôle la main au joueur de réseau de Petri dont l'évolution conduit à l'obtention d'un état stable et au solveur EDA qui permet l'intégration du modèle continu. Le cycle de fonctionnement du simulateur est présenté sur la figure 3.10.

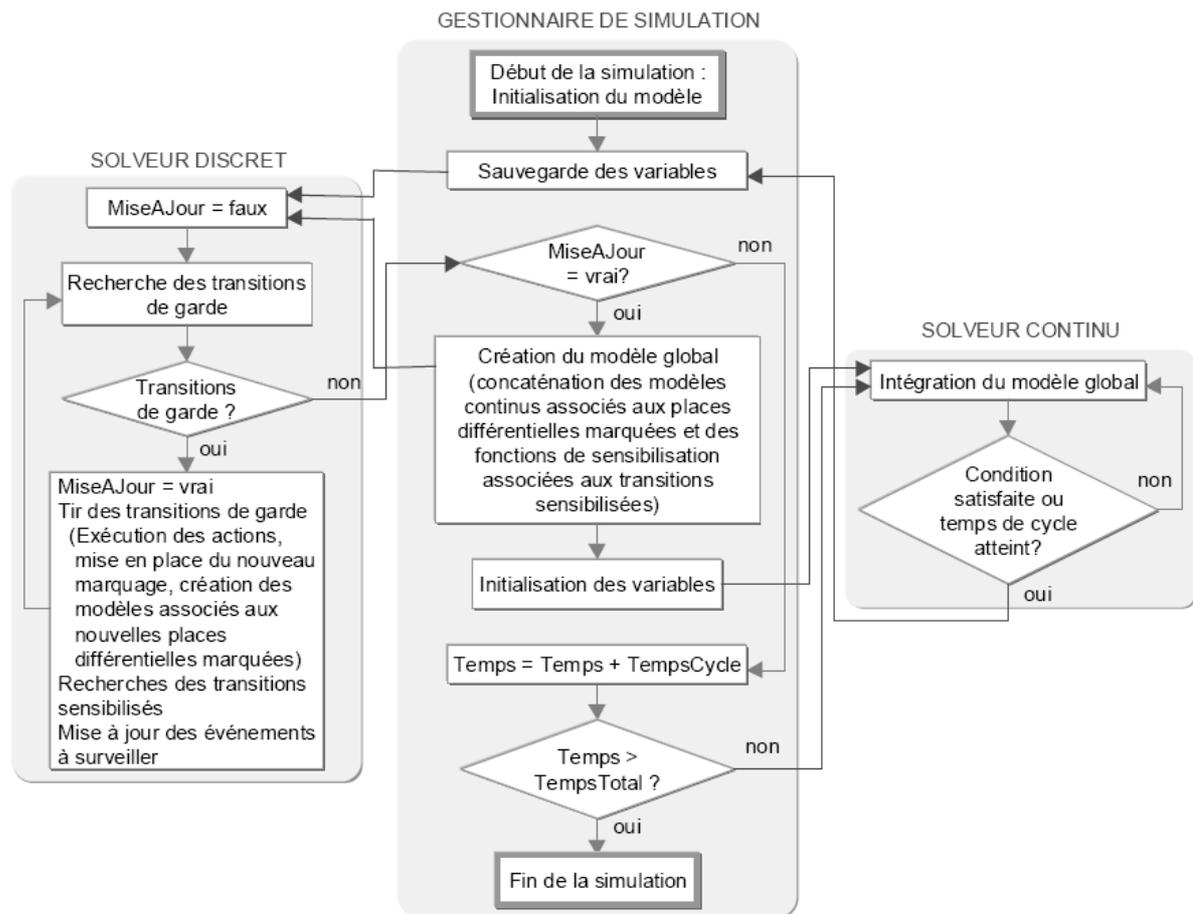


Figure 3.10 Noyau de simulation

1- Le gestionnaire de simulation se charge, dans un premier temps, de la construction et de l'initialisation du modèle continu global du système. Après avoir sauvegardé les variables d'état correspondantes, le gestionnaire donne la main au solveur discret.

2 - Le modèle discret est exécuté jusqu'à ce qu'une configuration stable soit atteinte, c'est à dire jusqu'à ce que plus aucune transition ne puisse être franchie. Les actions associées aux transitions franchissables sont exécutées et un nouveau marquage est établi.

3 - Ensuite, le modèle discret n'évoluant plus, la main est rendue au gestionnaire de simulation qui est responsable de la création du modèle continu global. Les modèles continus des places différentielles marquées des réseaux de Petri ayant été définis, le gestionnaire établit les

modèles globaux du réseau de Petri en cours et de ses esclaves. Ceux-ci résultent de la concaténation :

- du modèle continu du réseau en cours : il regroupe les systèmes algébro-différentiels associés aux places différentielles marquées et les fonctions de sensibilisation associées aux transitions sensibilisées,
- des modèles continus globaux intégrés dans les jetons marquant les places différentielles du réseau en cours.
- et enfin des modèles continus globaux des réseaux de Petri esclaves associés aux éléments constitutifs de l'appareil en cours.

Ainsi, le modèle global associé au réseau de Petri du niveau le plus haut, celui de la recette, constitue le modèle continu global du système à modéliser. Celui-ci, stocké dans l'attribut « model » du gestionnaire de simulation, est alors transmis au solveur continu afin d'être intégré. Il initialise ensuite toutes les variables en donnant momentanément la main au solveur *DISCO*. Celui-ci intègre en effet une procédure de calcul automatique de conditions initiales cohérentes. Le gestionnaire reprend ensuite le contrôle de la simulation qui impose alors une nouvelle mise à jour des réseaux de Petri. Si aucune évolution n'est constatée et si le temps final de la simulation n'est pas atteint, alors, le gestionnaire lance la simulation continue pendant une durée égale à un temps de cycle préalablement défini.

4. L'intégration du système algébro-différentiel global résultant est lancée, provoquant l'évolution continue au cours du temps des variables d'état. Celle-ci est stoppée dès l'occurrence d'un événement ou l'apparition de la fin du cycle.
5. Le gestionnaire de simulation enregistre les variables et redonne la main au solveur discret (--> retour en 2).

6 Conclusion

Ce chapitre a détaillé la plate-forme de simulation proprement dite en présentant tout d'abord sa structure générale puis en décrivant de façon plus détaillée les deux couches qui la composent. Cette description reste cependant assez générale et le lecteur pourra consulter les documents référencés pour plus d'informations.

Il a aussi permis d'illustrer la mise en œuvre du modèle RdPDO au sein de la plate-forme *PrODHyS*. Dans ce cadre, il a proposé une description détaillée des éléments le composant (transitions, places, arcs, etc.) et de son fonctionnement. Il a également mis l'accent sur le noyau de simulation et les interactions entre les composantes continues et discrètes du modèle.

Chapitre 4 Méthodologie pour le diagnostic

Introduction

La séparation entre le monde des systèmes continus et celui des systèmes à événements discrets, n'est plus aussi nette depuis une bonne dizaine d'années. Une des raisons de ce rapprochement entre les deux communautés se trouve dans la prise de conscience que les systèmes réels présentent à la fois des aspects continus et discrets. La création de groupes de travail tel celui sur les Systèmes Dynamiques Hybrides (GDR Automatique/SEE/Club EEA) a permis de favoriser les échanges entre les différents spécialistes, qu'ils soient issus du monde de l'automatique continue ou de celui de l'automatique événementielle. Les premiers résultats de ces groupes portent essentiellement sur les problèmes de modélisation et de commande des Systèmes Dynamiques Hybrides. Or, une des évolutions essentielles dans la conception des systèmes automatisés, et ceci indépendamment de leur complexité, concerne la prise en compte, dès les premières phases de conception, des préoccupations relatives à leur sûreté de fonctionnement. La sûreté de fonctionnement caractérise les performances d'un système en ceci qu'elle rend compte de son aptitude à remplir sa mission. Elle est liée à sa capacité à résister aux défaillances matérielles, logicielles et humaines, et aux agressions de son environnement. Cette problématique de Sûreté de Fonctionnement des systèmes, a fait l'objet d'un intérêt croissant de la communauté scientifique au cours de ces dernières années (voir activité du Groupement S3). Il nous semble que l'enjeu actuel, au niveau des Systèmes Dynamiques Hybrides, est maintenant celui de

l'utilisation des travaux passés, qui ont permis une meilleure compréhension de la modélisation des systèmes hybrides, pour la simulation, la supervision, la surveillance et le diagnostic de systèmes hybrides. En effet, un des principes de la supervision (à base de modèles) est de comparer en temps réel le comportement du système supervisé avec celui, simulé, d'un modèle afin de détecter toute déviation. Ce modèle peut également servir de support au diagnostic et à la reconfiguration du système. De nombreux travaux ont montré l'intérêt que porte la communauté automatique internationale au problème du diagnostic des Systèmes Dynamiques Hybrides.

1 Schéma et méthodologie du diagnostic

Globalement, l'objectif scientifique de notre travail est la recherche d'une méthodologie de diagnostic des Systèmes Dynamiques Hybrides. De manière générale, lorsque l'on parle de Diagnostic, on sous-entend la réalisation d'une architecture applicative qui, outre les fonctions nominales du système, met en œuvre des fonctions de détection, de localisation et de diagnostic des défaillances, de détection des changements de mode de fonctionnement (en particulier liés à des changements de comportement de l'environnement), ainsi que des fonctions de pronostic, d'accommodation des défaillances ou des agressions, de reconfiguration de la commande ou des objectifs.

Un premier objectif du travail proposé est de mettre en évidence les différents scénarii de fautes qui peuvent exister dans le système étudié. La finalité est de restreindre les chemins à explorer pour le diagnostic en utilisant une classification floue par la méthode *LAMDA* (Learning Algorithm for Multivariate Data Analysis). Un deuxième objectif, plus centré sur l'aspect Surveillance (détection, localisation, diagnostic) est de mettre en évidence des concepts cohérents permettant la formulation du problème du diagnostic par la plate-forme de simulation PrODHyS. Le formalisme adopté dans ce travail pour modéliser ces systèmes s'appuie sur le modèle « Réseau de Petri Différentiel à Objet » (RdPDO) qui est intégré dans la plate-forme de simulation PrODHyS (*Process Object Dynamic Hybrid Simulator*). Il possède l'avantage de prendre en compte le comportement hybride d'une part, en associant les variables continues aux jetons et d'autre part, en associant un système algèbro-différentiel aux places permettant de faire évoluer les variables d'état continues. Donc notre travail s'inscrit dans le domaine du diagnostic des systèmes hybrides et est basé sur l'utilisation d'un modèle. Il a pour objectif de diagnostiquer les fautes à partir de la connaissance structurelle, comportementale ou fonctionnelle du système en représentant ces connaissances (modèle du système) séparément de la connaissance sur la tâche de diagnostic. Deux raisonnements ont été suivis pour le diagnostic, le premier qualifié de raisonnement avant et le second de raisonnement arrière.

1.1 Méthode de diagnostic par raisonnement avant

La méthodologie de diagnostic proposée s'effectue en deux étapes. La première étape consiste à détecter à l'aide d'une classification floue, des fautes qui présentent les mêmes symptômes - à partir d'une connaissance préalable des états de défaillance obtenue par apprentissage - afin de réduire les chemins ou les scénarios à explorer lors de la seconde phase. Cette dernière sert à diagnostiquer la faute parmi celles détectées à l'étape précédente en levant l'ambiguïté. Pour ce faire, un raisonnement avant a été introduit, qui consiste à former pour chaque faute incriminée, un critère d'écart entre les mesures effectuées sur le système et celles émanant du modèle avec la faute simulée, sur une fenêtre temporelle et

d'isoler ainsi la faute aboutissant au critère le plus faible. La figure 4. 1 nous résume la méthodologie adoptée pour le diagnostic par raisonnement avant.

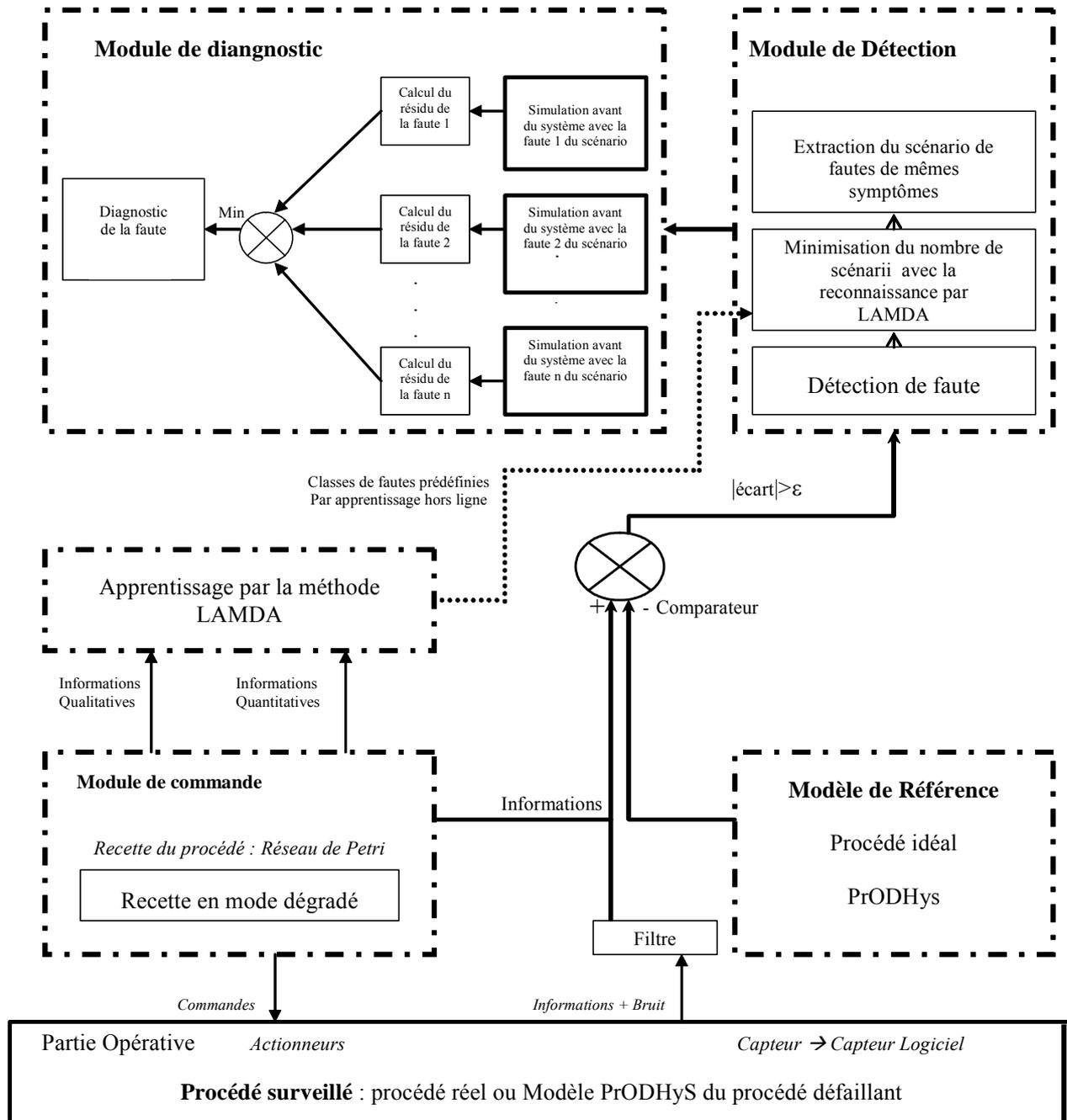


Figure 4.1 Schéma général de la méthode de diagnostic par raisonnement Avant

1.1.1 Modélisation de la recette du système en mode dégradé

Pour bien illustrer cette modélisation, prenons l'exemple de la figure 4. 2 . [Mokhtari et al.,06 a] Il est basé sur un système de régulation de volume de deux réservoirs. Il est

constitué d'un ordinateur, de deux pompes, de trois électrovannes (tout ou rien), de deux capteurs de volume (ou niveau) et des deux réservoirs régulés (Réservoir 1, Réservoir 2) et d'un troisième réservoir de vidange. Les deux réservoirs régulés alimentent des utilisateurs selon un besoin prédéfini (fonction du temps). Le volume dans chaque réservoir (1 ou 2) doit rester dans un intervalle donné $[V_{imin}, V_{imax}]$ ($i= 1$ ou 2). Le contrôle s'opère à l'aide du ordinateur qui décide, selon la valeur du volume (délivrée par le capteur), d'approvisionner (ou non) le réservoir en question en alimentant (ou non) l'électrovanne concernée. Pour chaque réservoir, on distingue donc deux phases de fonctionnement suivant que l'électrovanne alimentant ce réservoir est ouverte ou fermée :

- une phase de remplissage lorsque l'électrovanne est ouverte. Le volume dans le réservoir est croissant durant cette phase, et cela quelle que soit la valeur du débit de sortie vers l'utilisateur (le débit d'alimentation de l'électrovanne est bien supérieur, par hypothèse, au débit de sortie),
- une phase de vidange lorsque l'électrovanne est fermée. Le volume dans le réservoir est par conséquent décroissant.

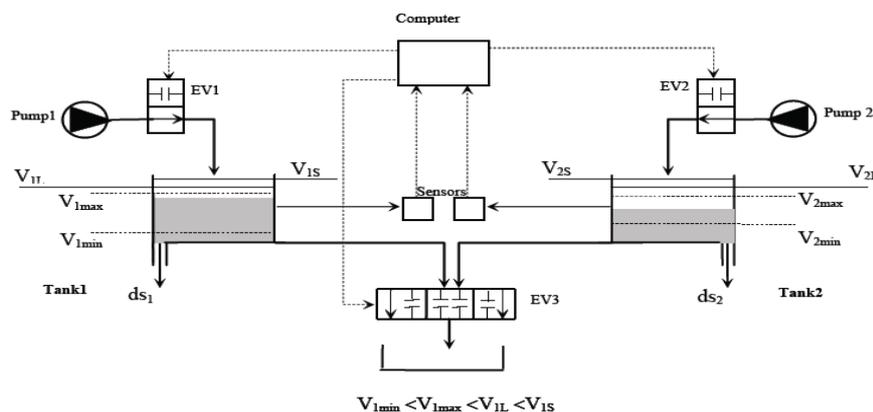


Figure 4.2 Système de régulation de deux réservoirs

La loi de commande en provenance du ordinateur pour chaque réservoir est telle que lorsque le volume dépasse la limite supérieure de commande V_{imax} pendant la phase de remplissage, alors le ordinateur commande la fermeture de l'électrovanne. Lorsque le volume devient inférieur à min (limite inférieure de la commande) durant la phase de vidange alors le ordinateur commande à l'électrovanne de s'ouvrir et on change par conséquent de phase de fonctionnement. Ce système doit assurer l'approvisionnement des utilisateurs tout en évitant le débordement de l'un des réservoirs. Une troisième électrovanne de secours est prévue pour cet effet. Elle est partagée entre les deux réservoirs et assure leur vidange quand ils débordent. Elle ne peut être utilisée que par un seul réservoir à la fois. Quand le volume dans l'un des réservoirs dépasse la limite supérieure de sécurité (V_{iL}), alors le ordinateur commande l'ouverture de cette électrovanne du côté du réservoir qui risque de déborder, et ce jusqu'à ce que le volume devienne inférieur à V_{imin} . Pour simplifier dans un premier temps, nous supposons que seules les électrovannes peuvent subir des défaillances. Les électrovannes 1 et 2 (prévues pour l'alimentation des réservoirs) peuvent être bloquées en ouverture. En cas de

défaillance de l'électrovanne 3 (de secours), celle-ci est mise hors service (on ne peut pas l'ouvrir pour faire la vidange). Maintenant, supposons que l'une des électrovannes (ex EV1) est bloquée en ouverture, c'est-à-dire qu'on ne peut pas la fermer, alors le volume dans le réservoir 1 continue de croître jusqu'à atteindre V1L. Dans ce cas, on doit ouvrir l'électrovanne EV3 de secours pour vidanger le réservoir1. Si l'électrovanne EV3 est aussi hors service (on ne peut pas l'ouvrir) ou si elle est occupée par la vidange du réservoir 2 (EV1 bloquée et EV3 hors service ou occupée), alors le volume dans le réservoir 1 dépassera V1L pour atteindre V1S qui est l'événement redouté (débordement de réservoir 1).

1.1.1.1 Modélisation du fonctionnement normal

Le réseau de Petri représenté sur la figure 4. 3 modélise le fonctionnement normal d'une cuve (un autre réseau similaire représenterait la deuxième cuve) avec la place *M_OpenAlim* qui modélise l'ouverture et la fermeture de l'alimentation, les places *FeedPL* et *IdlePL* représentent l'état du système en fonctionnement respectivement normal en alimentation+vidange et en vidange seulement ; la vidange infinie est représentée par la place *Open vanne vidange*. Remarquons que dans le Réseau de Petri de la figure 4. 3, on ne représente que la recette du fonctionnement du système, les appareils comme la vanne et l'alimentation sont dans une couche interne à la plate-forme PRODHyS, nous avons seulement fait figurer les places de commande de ces appareils qui sont *M_OpenAlim* et *Open Vanne Vidange*.

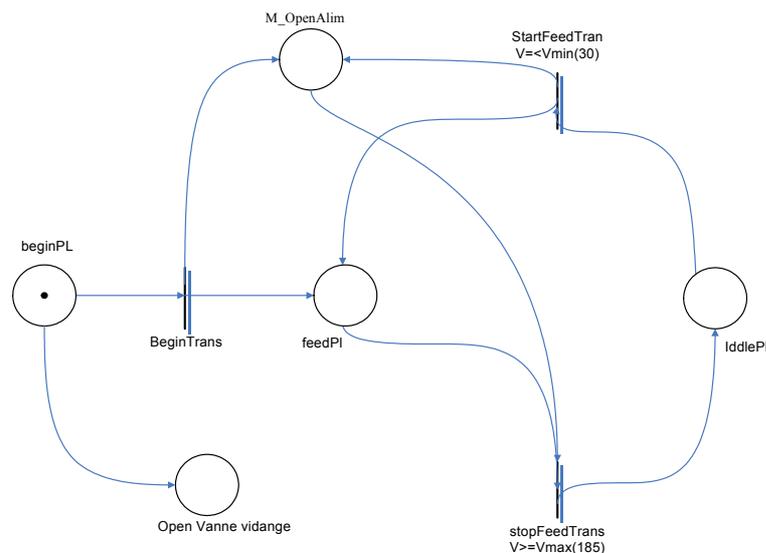


Figure 4.3 RdP du fonctionnement normal du système

1.1.1.2 Simulation du modèle de dysfonctionnement

Afin de modéliser le système en mode de dysfonctionnement, nous avons rajouté au RDP de la figure 4. 3 des places différentielles qui permettent le passage d'un état de bon fonctionnement à un état de dysfonctionnement comme le montre la figure 4. 4[Mokhtari et al.,06 b] . En fonctionnement normal, un jeton est placé dans les places *OkPL* et *TempoOkPL*. Le passage dans l'état de défaillance se fait suivant le franchissement de la transition *DefTran* qui place un jeton dans les places *DefPL* et *TempoDefPL*. Ceci engendre un blocage de l'alimentation en ouverture et vu que le débit de la vanne de vidange est inférieur au débit de l'alimentation, le niveau de l'eau croit, dépassant le seuil Vmax. Le retour à l'état normal se fait aussi par le franchissement de la transition *RepTran* qui replace

un jeton dans les places *OkPL* et *TempoOkPL*. En général, l'introduction de places temporisées nous permet de simuler plusieurs scénarii de fautes dans des intervalles finis ou infinis. [Mokhtari et al.,06 d]

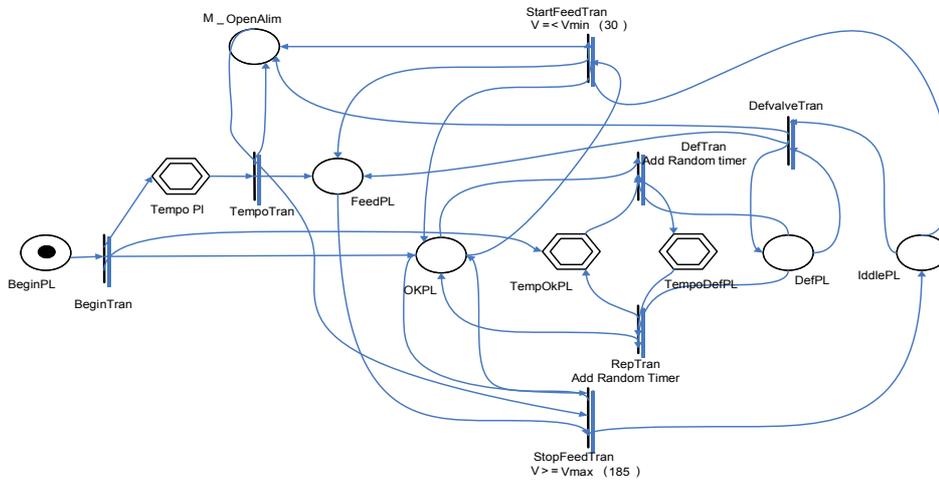


Figure 4.4 RdP du modèle avec dysfonctionnement

1.1.2 Notion de scénario redouté

Un scénario sous-entend un début, une fin et une histoire qui décrit l'évolution d'un système. Dans le contexte de la sûreté de fonctionnement, un scénario redouté mène à un état catastrophique ou dangereux : c'est l'état final (dit état redouté). L'état initial est un état de bon fonctionnement du système. Le scénario redouté décrit de manière précise (ce qui est nécessaire pour la compréhension) et concis (le juste nécessaire : causalité) comment le système quitte le bon fonctionnement pour évoluer vers un fonctionnement jugé dangereux. C'est en effet une description du système sous la forme de changements d'états et de suites d'événements qui mènent vers l'état redouté. C'est une explication claire des raisons pour lesquelles le système s'est trouvé ou risque de se trouver dans un état redouté donné. Ce n'est donc pas simplement une suite d'événements sans liens de causalité entre eux mais, un ensemble de relations d'ordre qui peuvent être interprétées comme des relations de causalité entre certains événements, ceux qui ne sont pas reliés par une relation d'ordre étant considérés comme des événements non causalement liés.

En résumé, un scénario redouté est une description de l'évolution de certains composants du système global à partir d'un état de bon fonctionnement jusqu'à l'occurrence de l'événement redouté. Ce scénario doit donc ne faire intervenir que les composants ayant un lien de causalité avec l'occurrence de l'événement redouté.

Les combinaisons de défaillances des composants élémentaires d'un système forment la majeure partie des causes possibles pouvant provoquer les scénarios redoutés. Ces combinaisons peuvent être identifiées par un raisonnement en parcourant l'ensemble des combinaisons possibles des défaillances élémentaires. Dans certains cas, ces défaillances élémentaires sont bénignes pour le fonctionnement du système mais, combinées avec des

interactions entre certains sous-systèmes (partage de ressource ou partage de variable continue), ces défaillances peuvent être à l'origine des scénarios redoutés.

1.1.2.1 Exemple de constellation de satellites

L'exemple choisi pour illustrer les différentes caractérisations de scénario est un système spatial. Il s'agit d'une constellation composée de deux satellites (S1, S2) et d'une station sol (SS). Nous avons choisi ce système à la fois pour sa simplicité et parce qu'il contient des scénarios qui permettent de voir comment dans une même séquence d'événements avec des ordres d'occurrence différents on peut ou non atteindre un état redouté. Le système est défaillant lorsque les deux satellites sont défaillants ou lorsque la station sol est en panne. Dans la figure 4. 5 qui représente la modélisation de ce système en réseau de Petri, la défaillance des satellites S1 et S2 est représentée par le franchissement des transitions f_1 et f_2 (marquage des place Ko_1 et Ko_2). La défaillance de la station sol est représentée par le franchissement la transition f_s (marquage de la place Ko_s).

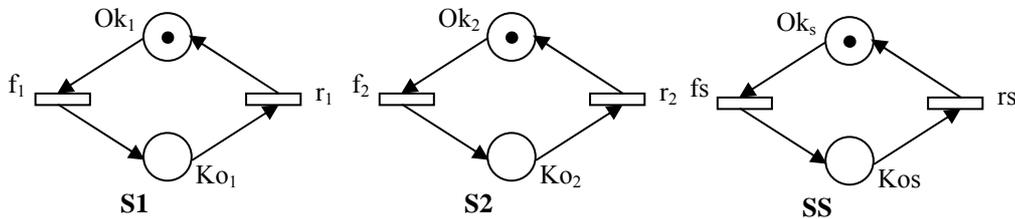


Figure 4.5 Modèle de réseau de Petri de la constellation

Chaque composant du système (un des deux satellites ou la station sol) peut être dans un état nominal (bon fonctionnement "Ok") ou en panne ("Ko"). Tous les composants sont en silence sur défaillance. Quand ils sont dans un état nominal, les satellites et la station sol ont des comportements légèrement différents. Un satellite actif "i" envoie sans interruption une télémessure "Out_i" à la station sol ; quand la station sol est dans un état nominal, elle fournit les données prévues "Out_s" si au moins une télémessure est disponible. On suppose qu'initialement les composants sont dans un état de bon fonctionnement (les place Ok₁, Ok₂ et Ok_s sont marquées) ; ils peuvent alors subir des défaillances et être réparés à tout instant (franchissement des transitions r₁, r₂ et r_s). L'arbre de défaillance de la constellation est donné sur la figure 4. 6

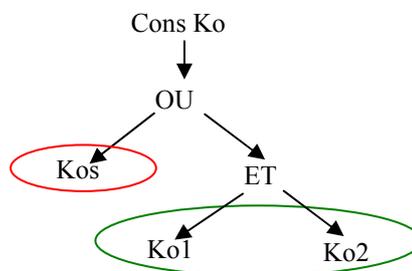


Figure 4.6 Arbre de défaillance de la constellation

1.1.3 Travaux au LAAS sur les scénarii redoutés

1.1.3.1 Travaux de Sarhane Khalfaoui

La thèse de [Khalifaoui 03] réalisée au LAAS en collaboration avec PSA (Peugeot Citroën) se focalise sur l'analyse qualitative de la sécurité des systèmes mécatroniques en vue de l'obtention des scénarios redoutés. La connaissance de ces scénarios permet d'évaluer leurs probabilités d'occurrence et de valider les lois de reconfiguration pour orienter le choix des concepteurs. Ces travaux ont permis de développer une méthode de recherche de scénarios redoutés basée sur la logique linéaire et le formalisme de réseau de Petri et d'un ensemble d'équations différentielles pour modéliser le système. Les scénarios sont ainsi extraits directement de modèle Réseau de Petri (RdP) sans générer le graphe d'accessibilité associé. Cette méthode ne se réduit pas à l'utilisation d'un joueur de réseau de Petri, elle est équivalente à une recherche systématique des causalités qui serait bien sûr des processus finis obtenus par dépliage de réseaux de Petri. Mais dans le cas de l'analyse de la sécurité des systèmes mécatroniques, seul l'état final du scénario (l'état redouté) est connu. L'état initial (dernier état de fonctionnement normal) et la liste des événements conduisant à l'état redouté doivent être découverts. La méthode mise au point permet, en partant de l'état redouté, de revenir en arrière à travers la chaîne des relations de cause à effet et d'extraire tous les scénarios possibles menant vers cet état. Ainsi, les événements constitutifs des scénarios sont découverts progressivement et seuls les événements nécessaires pour arriver à l'état redouté sont pris en considération.

Ce travail était une première approche et l'algorithme de construction des scénarios ne prend en compte que la vue discrète du système. Or il s'avère, même sur des exemples assez simples, que de nombreux scénarios sont produits et en plus ne vérifient pas les contraintes provenant de la vue continue du système hybride. Ils sont contradictoires avec la dynamique de la partie énergétique du système. Il semble plus efficace d'en éliminer, au moins un certain nombre, dès la construction des scénarios. Une amélioration de la méthode a été proposée par [medjoudj, 06] en prenant en compte l'aspect continu du système.

1.1.3.2 Travaux de Malika Medjoudj

La nouvelle méthode proposée prend en compte les conditions associées au franchissement de certaines transitions qui sont des seuils d'évolution des variables continues en transformant ces seuils en intervalles temporels correspondant au temps que met le système pour les atteindre quand les transitions sont sensibilisées. Il s'agit donc d'une «abstraction temporelle». Cette abstraction temporelle transforme le modèle de départ, un Réseau de Petri Prédicats Transitions Différentiels Stochastiques, en un Réseau de Petri arc-pt-temporel. Le principe consiste ensuite à interdire le franchissement d'une transition sensibilisée si elle est en conflit avec une autre transition sensibilisée de seuil temporel inférieur. Par exemple si dans un RdP nous avons deux transitions t_1 et t_2 en conflit et telles que le seuil associé à t_2 est inférieur à celui associé à t_1 , le franchissement de t_2 est prioritaire. La transition t_1 ne sera franchie que si t_2 n'est pas franchissable (non sensibilisée), nous avons alors une relation de causalité indirecte entre t_2 et t_1 qui induit une relation de précédence entre l'événement qui désensibilise t_2 et celui qui est associé au franchissement de t_1 . Dans cette nouvelle approche, le scénario correspondant au franchissement de t_1 suivi de t_2 sera éliminé car il est incohérent vis-à-vis de la dynamique continue du système. Malgré cette amélioration, le nombre de scénarii explorés reste très important. C'est dans le but de réduire ce nombre de scénarii que

nous avons adopté la méthode de classification LAMDA permettant d'utiliser les informations qualitatives et quantitatives issues du système pour effectuer une première étape de détection.

1.1.4 Diagnostic par reconnaissance de formes

La reconnaissance des formes regroupe l'ensemble des méthodes permettant la classification automatique d'objets, suivant sa ressemblance par rapport à un objet de référence. Une forme est définie à l'aide de n paramètres, appelés individus, qui sont les composantes d'un vecteur forme $X_i = [x_1, x_2, \dots, x_n]^T$. Une forme peut donc être représentée par un point d'un espace à n dimensions. Dans la suite, C_1, C_2, \dots, C_k seront les k différentes classes, ou formes types, d'un problème de reconnaissance de formes. L'objectif est alors, étant donnée une forme X , de décider si elle doit être affectée à la classe C_1 , ou C_2, \dots , ou C_k . Chaque classe occupe une zone géométrique de l'espace à n dimensions, le problème consiste alors, connaissant les différentes classes, à définir les frontières les séparant. La résolution d'un problème de reconnaissance de formes se ramène finalement à la détermination des frontières entre classes. Comme montre la figure 4.7, la connaissance des frontières entre classes permet l'affectation d'une nouvelle observation à l'une d'entre elles, c'est l'opération de classification.

La résolution d'un problème de reconnaissance de formes nécessite : (a) la définition précise des k classes entre lesquelles va s'opérer la décision, (b) Le choix d'un jeu de caractères pertinents pour la discrimination des vecteurs des formes et (c) l'élaboration d'un classificateur permettant l'affectation d'une forme observée à l'une des classes.

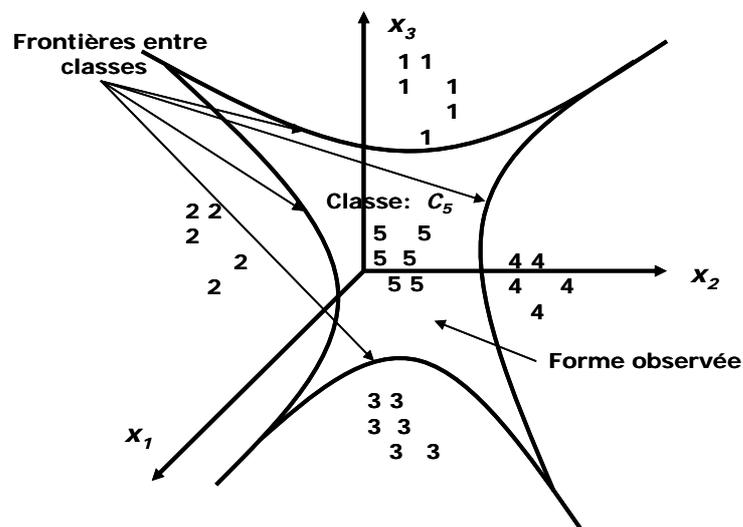


Figure 4.7 La forme observée X est ici associée à la classe C_5 .

La démarche exposée précédemment, peut être appliquée avec profit au diagnostic d'une installation industrielle. Dans ce cas, les n paramètres du vecteur forme résultent de mesures réalisées sur le système à surveiller et des observations réalisées par les opérateurs en charge de l'installation. Une bonne connaissance de l'installation permettra de choisir les paramètres le plus adaptés. Ces paramètres, une fois choisis, sont mesurés en permanence sur l'installation à surveiller. Par suite des bruits de mesure et des diverses perturbations auxquelles le système est inévitablement soumis, une suite d'observations du vecteur forme X ,

résultant d'un même état de fonctionnement du système, ne va pas se retrouver en un seul point, mais occupe une zone de l'espace à n dimensions. Si les n paramètres ont été bien choisis, une forme correspondant à un fonctionnement normal appartiendra à une certaine zone ou classe, alors qu'une forme correspondant à un autre mode de fonctionnement appartiendra à une autre classe. Ainsi, chaque mode de fonctionnement peut être représenté au moyen d'une classe de l'espace de représentation.

La figure 4. 8 présente la structure simplifiée d'un système de diagnostic par reconnaissance des formes. La fonction d'observation a pour rôle d'élaborer le vecteur forme à partir des mesures et observations réalisées sur l'installation. La forme ainsi générée est appliquée au bloc de classification permettant de réaliser son affectation à l'une des classes connues et au mode de fonctionnement correspondant.

La construction d'un dispositif de diagnostic par reconnaissance de formes se déroule en trois étapes principales : (a) la création d'une base d'apprentissage qui regroupe un certain nombre de classes, chacune correspondant à un mode de fonctionnement particulier du système, (b) le choix d'un classificateur que permettra de décider de l'appartenance d'une nouvelle observation à l'une des classes existantes, et (c) l'utilisation effective du classificateur en phase d'exploitation qui consiste à implémenter le système afin de proposer une décision pour toute forme n'appartenant pas à une classe déjà définie (mise en œuvre du classificateur destiné à un fonctionnement en ligne).

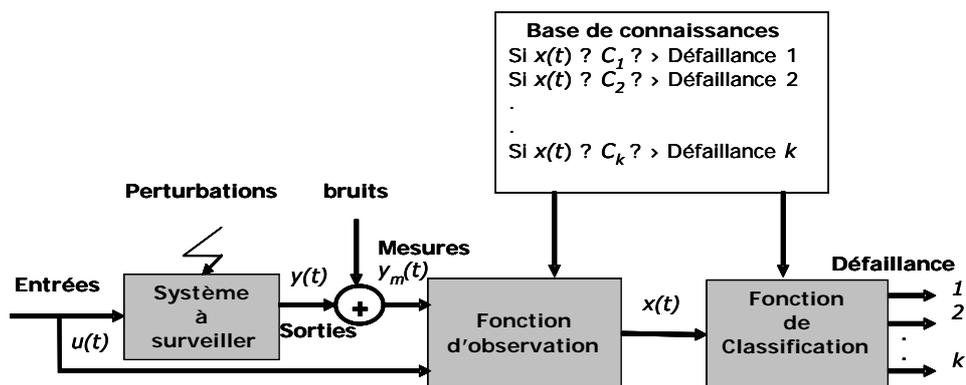


Figure 4.8 Structure d'un système de diagnostic par reconnaissance des formes

Nous allons dans la suite présenter sommairement les méthodes de classification floue pour le diagnostic dans le cadre de systèmes complexes et s'attarder plus particulièrement sur la méthode de classification *LAMDA* qui peut être considérée comme une variante de ce type de méthode.

1.1.4.1 L'approche floue

Dans la théorie des ensembles classiques, la notion d'appartenance est fondamentale, mais elle est de type tout ou rien, un élément appartient ou n'appartient pas à un ensemble [TOSCANO, 2005]. Un tel outil s'avère alors difficilement utilisable lorsqu'il s'agit de manipuler des données vagues, imprécises, contradictoires ou lorsqu'il s'agit de classer des informations suivant des catégories aux frontières mal définies. La théorie des ensembles flous, par un assouplissement de la notion d'appartenance, permet d'atteindre de tels objectifs. Elle s'avère

alors plus adaptée pour la représentation des connaissances qualitatives. Les applications floues sont nombreuses, on peut citer la gestion financière, la médecine, le diagnostic, la commande automatique de processus et bien d'autres.

L'idée de l'approche floue est de construire un dispositif, appelé système d'inférences floues, capable d'imiter les prises de décision d'un opérateur humain à partir des règles verbales traduisant ses connaissances relatives à un processus donné.

La relation mathématique existant entre un défaut et ses symptômes est le plus souvent difficile à obtenir. Toutefois, les opérateurs humains ayant en charge la maintenance et la conduite du système sont souvent capables, de par leur expérience, de déterminer, sur la base de leurs observations, l'élément défaillant qui est à l'origine d'un comportement qu'ils ont jugé anormal. Ce type de savoir peut être exprimé à l'aide de règles de la forme :

SI condition ALORS conclusion

où la partie *condition* comporte les symptômes observés et la partie *conclusion* l'élément défaillant. Ce type de connaissances peut alors être utilisé pour construire un système d'aide au diagnostic de l'installation. La notion de sous-ensemble flou introduite par Zadeh en 1965, est fondée sur le degré d'appartenance, qui généralise les fonctions caractéristiques.

Chaque classe en relation avec les modes de fonctionnement du système peut être interprétée comme un sous-ensemble d'un espace multidimensionnel, la classification se résume alors à la recherche des propriétés caractéristiques de ces ensembles. Finalement, chaque classe peut être modélisée au moyen d'un sous-ensemble flou caractérisé par une fonction d'appartenance multidimensionnelle qu'il s'agit de déterminer. Dans la figure 4. 9 le vecteur des symptômes x du classificateur, élaboré à partir des grandeurs mesurées sur le système, peut être vu comme une forme, qu'il s'agit de classer parmi l'ensemble des formes correspondant à un fonctionnement normal ou non.

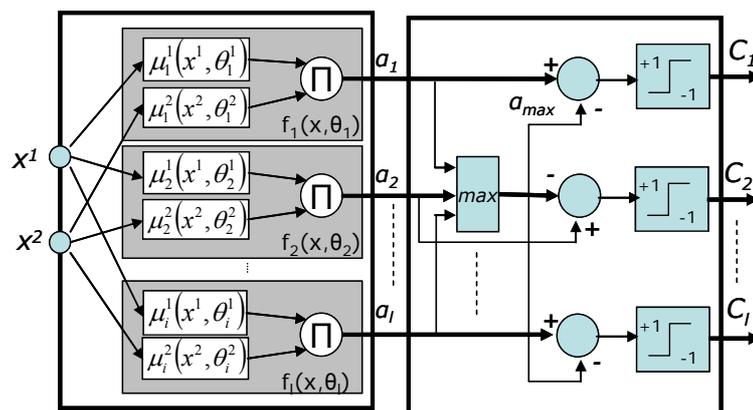


Figure 4.9 Structure générale du classificateur

La classification est réalisée au moyen des fonctions de vérité et de décision, dont le paramétrage réalise la frontière entre les classes. L'ensemble d'apprentissage permet, d'une part de générer la base de règles et d'autre part de réaliser le paramétrage des fonctions d'appartenance.

Cette approche consiste à rechercher les projections de la fonction d'appartenance à une classe, sur chacune des dimensions de l'espace de représentation. La fonction

d'appartenance au sous-ensemble flou global est alors obtenue au moyen d'un opérateur de conjonction, ce qui se traduit symboliquement par l'écriture d'une règle d'appartenance à la classe considérée. Le classificateur ainsi comporte un nombre de règles égales au nombre de classes. La base de règles du classificateur est composée d'une liste de propositions conditionnelles, de la forme,

$$R_i : Si(x^1 \text{ est } A_i^1) \text{ et } (x^2 \text{ est } A_i^2) \text{ et } \dots \text{ et } (x^q \text{ est } A_i^q), x \in C_i; i = 1 \dots l \quad 4.1$$

où l est le nombre total de règles, les A_i^j sont des sous-ensembles flous définis par des fonctions d'appartenance $\mu_i^j(x^j, \theta_i^j)$ dont θ_i^j représente le vecteur des paramètres, et les C_i représentent les différentes classes de la classification.

Soit α_i le degré d'appartenance de l'observation x à la classe C_i ,

$$\alpha_i(x) = \prod_{j=1}^{n_x} \mu_i^j(x^j, \theta_i^j) \quad 4.2$$

Soit d'autre part α_{\max} le plus grand degré d'appartenance correspondant à l'observation x ,

$$\alpha_{\max} = \max_{i=1 \dots l} \alpha_i(x) \quad 4.3$$

L'observateur x doit être affectée à la classe C_i permettant d'obtenir le plus grand degré d'appartenance, d'où la règle de décision :

L'observation x est affectée à la classe C_i telle que $\alpha_i(x) - \alpha_{\max} \geq 0$

Les sorties délivrées par ce classificateur sont :

$$\begin{aligned} C_i &= +1 \text{ Si } x \in C_i \\ C_i &= -1 \text{ Si non} \end{aligned} \quad 4.4$$

La règle de décision doit être complétée afin d'offrir la possibilité de rejets d'ambiguïté et d'utilisation de distance. Il y a ambiguïté lorsqu'une observation appartient à un domaine commun à plusieurs classes. Dans ses conditions, les degrés d'appartenance $\alpha_i(x)$ correspondants sont assez peu différents, ce qui peut conduire à une mauvaise classification.

1.1.5 La méthode de classification LAMDA

LAMDA (Learning Algorithm for Multivariate Data Analysis) est une stratégie de classification avec apprentissage proposée par Joseph Aguilar-Martin [Aguilar-Martin et al.,80]. La méthode a été développée par plusieurs chercheurs [Aguilar-Martin et al.,82; Desroches,87; Piera et al.,89]. C'est un algorithme d'analyse de données multidimensionnelles par apprentissage et reconnaissance de formes. La formation et la reconnaissance de classes dans cette méthode sont basées sur l'attribution d'un objet à une classe à partir de la règle heuristique appelée adéquation maximale. *LAMDA* a été utilisé dans des domaines très diverses : en analyse biomédicale [Chan et al., 89], pour les bio-procédés [Aguilar-Martin et al., 99], pour l'étude des processus de dépollution des eaux usées [Waissman-Vilanova,00], pour la psychologie [Galindo,02]. Plus récemment, dans les travaux de thèse de Kempowsky [Kempowsky,04a] (procédés industriels), Orantes (placement ds capteurs) [Orantes,05], [Orantes et al.,06] et [Atine,05] (segmentation

d'images biologiques). *LAMDA* a été mise en œuvre pour la première fois dans le logiciel SYCLARE [Desroches,87], puis plus récemment dans le logiciel SALSA [Kempowsky,04b].

1.1.5.1 Les caractéristiques principales de *LAMDA*

➤ **L'adéquation :**

LAMDA ne considère pas la similarité ou la distance entre éléments pour la classification, mais il introduit la notion de *degré d'adéquation* de l'élément aux classes déjà formées,

➤ **L'attribution:**

On affecte chaque élément à la classe dont le degré d'adéquation est maximal, cependant on conserve les degrés d'adéquation à toutes les classes, ce qui constitue une partition floue.

➤ **Entropie maximale :**

Dans l'univers d'où proviennent tous les éléments, le concept d'entropie maximale est à la base de la modélisation de l'homogénéité qui correspond à l'absence d'information, il correspond à une classe qui accepte tous les éléments avec le même degré d'adéquation. Cette classe est très importante pour le processus de formation de nouvelles classes : on l'appelle « Classe non-informative » (*NIC*). L'existence de cette classe agit comme une limitation ou un seuil : aucun élément ne sera assigné à une classe si son degré d'adéquation globale n'est pas supérieur à celui de la classe non informative.

➤ **Degré d'adéquation avec connectifs :**

Le degré d'appartenance à une classe est calculé à partir des valeurs de ses descripteurs. Ces valeurs contribuent au calcul de l'adéquation à chaque classe au moyen de degrés d'adéquations marginales fournis par des fonctions de distribution floue.

➤ **Connectifs :**

L'agrégation des adéquations marginales se fait à partir de connectifs de la Logique Floue, c'est-à-dire d'une t-norme et de son dual la t-conorme ou s-norme.

1.1.5.2 Les principales propriétés de *LAMDA*

- On peut choisir les fonctions d'appartenance de la Logique Floue, qu'elles soient associées à certaines distributions probabilistes, (Binomiale, gaussienne, ...).
- On peut choisir les connectifs parmi des familles de t-normes (produit probabiliste, min-max de Zadeh, t-normes de Frank, t-normes de Yager).
- On peut ajuster le degré d'exigence par l'introduction de connectifs mixtes linéairement compensés : une grande exigence considérera plus d'éléments non reconnus, et en cas d'auto-apprentissage créera plus de classes. Il est possible d'obtenir des classifications différentes du même groupe d'objets ordonnées par rapport au concept d'"exigence".

- On peut gérer des variables qualitatives et quantitatives simultanément par le choix de fonctions d'adéquation marginale tenant compte des modalités.
- Il peut s'adapter à une situation évoluant au cours du temps en raison d'un apprentissage séquentiel.
- A la fois des apprentissages supervisés et non supervisés peuvent être effectués, et aussi compléter un apprentissage dirigé, par la création de nouvelles classes (apprentissage supervisé).

1.1.5.3 Les défauts de *LAMDA*

- Il n'y a pas de garantie d'obtenir la meilleure partition. La qualité de la partition obtenue est laissée à l'appréciation de l'expert.
- Il n'y a pas, pour l'instant, de procédure automatique permettant de choisir les connectifs et l'indice d'exigence, mais des travaux récents (thèse de Claudia Isaza) ont permis de développer une méthodologie visant à optimiser la partition obtenue en termes de compacité et de séparation des classes en utilisant les degrés d'appartenance d'une classification floue et les concepts de similitude entre ensembles floues.

1.1.5.4 Méthodologie générale

On considère qu'un objet ou situation x est décrit par un nombre fini et fixé d'attributs notés x_1, x_2, \dots, x_n . Afin d'obtenir une confrontation entre x et les différentes classes C_j , une fonction d'adéquation $M_{i,j} : D_i \times C \rightarrow [0,1]$ nommée Degré d'Adéquation Marginale (*DAM*) est calculée pour chaque attribut x_i et la forme dans laquelle l'espace de description correspondant est représenté de façon générale dans la classe C_j .

- Le *DAM* est une fonction d'appartenance issue de la Logique Floue. Cette fonction peut exprimer un degré entre l'adéquation d'un attribut à une classe et l'inadéquation de l'attribut à cette classe. Entre ces deux valeurs extrêmes, il existe une valeur de l'attribut telle que, si on se limitait à cette unique information, il serait impossible de décider de l'appartenance de cet objet à une classe. Ceci est équivalent à une adéquation neutre. Le concept d'adéquation neutre est nécessaire dans la représentation d'information insuffisante pour la classification. L'expression d'une adéquation neutre, pour toute valeur dans l'espace de description, est équivalente à « l'indistingibilité » d'une certaine classe. L'une des spécificités importantes de la méthode *LAMDA* réside dans la prise en compte de ce manque d'information au moyen d'une classe non informative *NIC*. La classe *NIC* équivaut donc à considérer « indistingables » tous les attributs.

1.1.5.4.1 Degré d'adéquation Marginale (*DAM*) et Degré d'adéquation Globale (*DAG*)

Pour un élément donné, les caractéristiques par rapport à chaque descripteur interviennent dans le calcul du degré d'appartenance de cet élément à une classe par ce qu'on a appelé « le degré d'adéquation marginale *DAM* ». Pour chaque élément, on détermine un vecteur des degrés d'appartenance marginale.

L'information de ces degrés devra être agrégée afin d'obtenir un indicateur qui permettra de savoir comment un objet satisfait les conditions propres à la classe C_j . Cet

indicateur est modélisé par un opérateur logique d'agrégation $L : [0,1]^n \rightarrow [0,1]$ Le résultat est appelé le Degré d'adéquation Globale (*DAG*) qui est fonction des appartenances marginales. $DAG(X/C)$ est le degré d'appartenance globale d'un élément X à une classe C , $\mu_j = DAM(x_j/C)$ est le degré d'appartenance marginale (ou partielle) par rapport au descripteur j , et $[\mu_1, \dots, \mu_j \dots, \mu_p]$ est le vecteur des appartenances marginales.

Dans ce qui suit, nous donnons de façon détaillée, les différentes étapes de l'algorithme de classification.

Soit un élément X et les classes C_0, C_1, \dots, C_K , une classification se déroule de la façon suivante.

L'objectif premier est de calculer les degrés d'appartenance globale *DAG* de l'élément X à chacune des classes C_1, \dots, C_K et C_0 la classe vide ou résiduelle, notés $[\rho_1, \dots, \rho_p]$. Pour ce faire, on calcule des degrés d'appartenance marginale (μ_j) par rapport à chaque descripteur.

Le calcul du degré d'appartenance marginale ou partielle dépend du type de descripteur correspondant. Dans la logique propositionnelle, la façon d'agréger les informations se fait par le biais d'opérateurs de conjonction. Si nous utilisons l'opérateur d'intersection (conjonction logique), un objet aura une adéquation élevée à une classe seulement si tous les attributs de l'objet ont un degré d'adéquation élevé pour cette classe. Au contraire, dans le cas de l'opérateur d'union (disjonction logique), le fait que l'un des attributs présente un degré d'adéquation marginale élevé sera suffisant pour considérer cet objet adéquat à la classe correspondante. Cependant, il est normal de rencontrer des situations où nous ne pouvons pas être assez exigeant pour utiliser l'opérateur d'intersection, mais pour lesquelles l'opérateur d'union serait trop permissif. Pour ces situations, des opérateurs mixtes d'agrégation linéairement compensés ont été proposés. Ces opérateurs ont un comportement réglable allant de l'union à l'intersection. Nous sommes alors capables d'ajuster l'exigence de la méthode. Le schéma général du calcul de l'adéquation d'un objet à une classe est représenté sur la figure 4. 10.

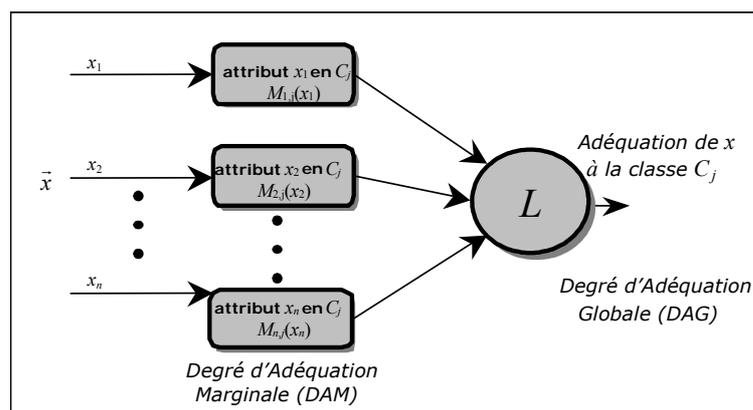


Figure 4.10 Schéma général du calcul de l'adéquation d'un objet à une classe

Bien qu'on soit obligé en pratique d'assigner une seule classe à chaque élément, le résultat final de la classification n'est pas une partition classique de l'univers de description

mais une partition floue, où chaque objet dans l'univers de description a une valeur d'appartenance à chacune des classes existantes. Afin d'obtenir une partition classique de l'univers de description, une fois le *DAG* calculé pour toutes les classes, x sera attribué, selon le critère d'adéquation maximale, à la classe où la valeur du *DAG* est maximale. Pour des raisons algorithmiques, et pour assurer que l'assignation d'un objet à une classe soit unique, une deuxième règle de décision *FF* (first found) est prise en considération : lorsque la valeur maximale du *DAG* est trouvée dans plus d'une classe, l'objet sera placé dans la première classe à laquelle il a été confronté et qui présente une valeur d'appartenance maximale. On peut aussi envisager une variante appelée *LF* (last found) dans laquelle c'est la dernière classe qui est retenue.

L'apprentissage consiste à extraire, à partir de l'information contenue dans une base de données connue d'apprentissage, les caractéristiques qui décrivent le mieux chaque concept. Dans *LAMDA*, ceci est traduit par l'estimation des paramètres définissant les fonctions d'appartenance des *DAM*. Dans notre approche, les fonctions d'appartenance de chaque attribut sont estimées indépendamment de l'information disponible sur les autres attributs. Les paramètres représentant une classe sont estimés à partir uniquement des données de l'ensemble d'apprentissage appartenant à cette classe. Ceci implique que, dans *LAMDA*, l'apprentissage d'un concept (ou classe) est réalisé à partir uniquement de l'information dont nous disposons sur celui-ci, et non par opposition aux autres concepts établis.

Après la présentation des principes et des bases de la méthodologie de classification *LAMDA*, nous explicitons dans ce qui suit les deux parties essentielles de cette méthodologie : les fonctions d'appartenance qui définissent le Degré d'Adéquation Marginale (*DAM*) et les opérateurs logiques d'agrégation qui déterminent le Degré d'adéquation Globale (*DAG*). Déterminer les fonctions d'appartenance à partir de données est une opération très importante de l'application de la logique floue à des situations réelles. Toutefois, il n'existe pas de guide ou règle qui puissent être utilisés afin de choisir la meilleure méthode pour obtenir ces fonctions. De plus, il n'existe pas de mesure pour évaluer la qualité d'une fonction d'appartenance.

1.1.5.4.2 Fonctions d'adéquation (appartenances floues)

Pour bien délimiter les types de fonctions d'appartenance qui sont adaptés à la méthode *LAMDA*, on établit les contraintes suivantes :

- *LAMDA* étant une méthode conceptuelle, les fonctions d'appartenance des *DAM* dépendent des paramètres représentant les données d'apprentissage.
- Les fonctions d'appartenance utilisées pour le *DAM* doivent refléter l'adéquation de la valeur d'un attribut à une classe par rapport à l'inadéquation. Dans *LAMDA*, les valeurs, minimale et maximale, possibles d'une fonction (0 et 1) signifient une totale inadéquation de l'attribut à la classe et une totale adéquation, respectivement. Parmi les valeurs extrêmes de la fonction d'appartenance, une valeur d'adéquation doit être représentée par un degré d'appartenance bien défini et connu. C'est-à-dire, les fonctions d'appartenance utilisées dans *LAMDA*, sont plutôt une généralisation floue d'une logique à trois valeurs (0,1, ?) que d'une logique binaire.
- Afin de modéliser la classe *NIC*, pour des paramètres précis, la fonction d'appartenance doit montrer une adéquation neutre dans tout l'espace de description.

1.1.5.4.2.1 Cas des descripteurs qualitatifs

Un descripteur qualitatif est caractérisé par un ensemble non ordonné de modalités. Lors de la classification, on procède au calcul des fréquences de chaque modalité à l'intérieur d'une classe. Le calcul de la fonction d'appartenance marginale d'un élément est la fréquence de la modalité observée dans cette classe.

1.1.5.4.2.2 Cas des descripteurs quantitatifs

Les descripteurs quantitatifs sont tels que les valeurs associées peuvent se mettre dans un ensemble ordinal discret ou continu. Cet ensemble se présente donc comme un intervalle $[x_{min}, x_{max}]$ et peut être réduit à l'intervalle $[0, 1]$ par la formule de normalisation suivante :

$$x_j = \frac{x_j - x_{min}}{x_{max} - x_{min}} \quad 4.5$$

Il existe plusieurs fonctions pour représenter l'appartenance d'un descripteur. Dans ce qui suit, nous donnons les 3 fonctions que nous avons utilisées dans notre travail.

Binomiale floue. C'est une extension floue de la fonction binomiale [AGUILAR-MARTIN, 1980]:

$$\mu(x_j / C_i) = \rho_{i,j}^{x_j} (1 - \rho_{i,j})^{(1-x_j)} \quad 4.6$$

Binomiale floue Centrée. Cette fonction permet une partition autour des centres des classes [WAISSMAN-VILANOVA, 2000]. Le *DAM* est calculé alors par la proximité entre la valeur x_j observée pour le descripteur j et le centre c_{ij} du même descripteur pour la classe i :

$$\begin{aligned} par &= \rho_{i,j}^{x_j} (1 - \rho_{i,j})^{(1-x_j)} \\ des &= x_j^{x_j} (1 - x_j)^{(1-x_j)} \end{aligned} \quad 4.7$$

$$\mu(x_j / C_i) = \frac{par}{des}$$

Gauss: Dans ce cas, les relations utilisées sont à rapprocher de celles donnant la moyenne et l'écart type d'une distribution gaussienne non normalisée:

$$\mu(x_j / C_{ij}) = e^{-\frac{1}{2\sigma_{ij}^2}(x_j - \mu_{ij})^2} \quad 4.8$$

où μ_{ij} et σ_{ij} correspondent, respectivement, à la valeur moyenne et à la variance du descripteur j pour la classe i .

Une fois que les *DAMs* ont été obtenus pour une classe, le *DAG* doit être calculé. Il est obtenu par l'agrégation des *DAMs* en utilisant les connectifs choisis, [PIERA et al., 1991].

L'étape suivante consiste, à l'aide du connectif, à déterminer le degré d'appartenance globale *DAG* de l'élément X à la classe C_i . Les connectifs mixtes linéairement compensés que nous avons cités précédemment effectuent une interpolation entre l'opérateur logique d'intersection (T-norme) et celui de l'union (T-conorme), par le biais du paramètre α , par la formule :

$$DAG_{\alpha}(DAM_1, \dots, DAM_d) = \alpha T(DAM_1, \dots, DAM_d) + (1 - \alpha) S(DAM_1, \dots, DAM_d)$$

Les appartenances marginales pour chaque descripteur, permettent de calculer l'appartenance d'un élément à chacune des classes. Cet élément est assigné à la classe dont le degré d'appartenance globale correspondant est maximal. Le paramètre α est l'indice d'exigence et $\alpha \in [0,1]$, Pour $\alpha = 0$, la classification est peu exigeante dans l'attribution d'un individu à une classe. L'exigence la plus forte est obtenue pour $\alpha = 1$.

L'organigramme général de l'algorithme de classification *LAMDA* est donné sur la figure 4. 11. Ce schéma illustre l'algorithme *LAMDA* dans le cas de l'auto-apprentissage ou bien dans le cas de la reconnaissance. Dans le cas de l'apprentissage, sachant qu'il s'effectue de façon séquentielle, la représentation d'une classe varie après qu'un élément ait été attribué. La mise à jour de la classe s'effectue en prenant en compte les caractéristiques du nouvel élément ainsi que la description de la classe à l'instant précédent.

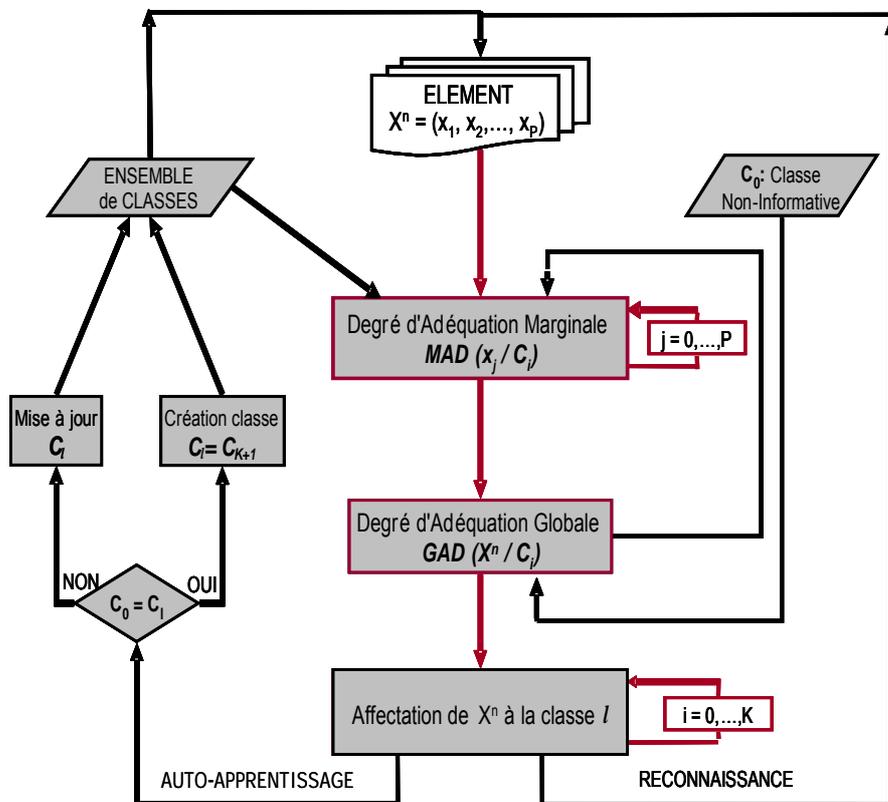


Figure 4.11 Algorithme général de LAMDA

1.1.5.4.2.3 Actualisation des paramètres

L'actualisation des paramètres associés aux descripteurs quantitatifs se fait de la façon suivante:

$$\rho_{i,j} = \rho_{i,j} + \frac{x_j - \rho_{i,j}}{N + 1} \quad 4.10$$

où N est le nombre d'objets attribués à cette classe. Pour procéder séquentiellement, il est nécessaire de connaître le nombre d'éléments ayant servi au calcul des paramètres de la classe correspondante.

On peut créer aussi, à l'aide de la classe (NIC) une nouvelle classe qui va être caractérisée par l'affectation d'un élément à cette classe. X est le premier élément d'une nouvelle classe C_{K+1} et la représentation de cette nouvelle classe dépendra de cet élément. On prendra un paramètre fictif N_0 correspondant au «nombre d'éléments de la classe « NIC ».

$$\rho_{i0} = \rho_{i0} + \frac{x_i - \rho_{i0}}{N_0 + 1} \quad 4.11$$

Dans le cas de l'auto-apprentissage, toute classe a dû être initialisée par la classe NIC , c'est pourquoi la formule de la mise à jour doit contenir ce paramètre fictif N_0 et elle devient:

$$\rho_{ik} = \rho_{ik} + \frac{x_i - \rho_{ik}}{N_0 + N + 1} \quad 4.12$$

Le paramètre $N_0 > 0$ détermine l'initialisation de l'apprentissage. Sa valeur peut être choisie arbitrairement mais elle influe sur le pouvoir d'absorption de chaque classe nouvelle, plus N_0 est grand, moins la classe nouvelle sera influencée par le premier élément; par contre en apprentissage dirigé ce paramètre n'a pas d'influence sur le résultat de la classification.

Dans le cas de la reconnaissance de formes, l'élément est attribué à une classe significative ou rejetée dans la classe résiduelle NIC . Dans le cas de l'apprentissage, s'il est affecté à une classe significative, il y a modification des paramètres de cette classe. En revanche, si la classe vide a la plus grande adéquation, une nouvelle classe doit être créée pour contenir cet élément. Il y a rejet si la classe vide est la plus proche et qu'il n'y a pas possibilité de création de nouvelle classe parce que le nombre maximum des classes créées est atteint.

1.1.5.5 L'outil SALSA [Kempowsky, 04b]

L'outil Salsa a été développé sur la base de la méthode *LAMDA* dans le cadre du projet européen CHEM (Advanced Decision Support Systems for Chemical and Petrochemical Manufacturing Processes) dont l'objectif principal a été le développement d'une plate-forme générique d'outils intégrés basés sur des méthodologies avancées pour la surveillance, la supervision, la détection de défauts et le diagnostic des procédés [CHEM, 2006]. Il a été utilisé pour déterminer le comportement de la station SMAPA de production d'eau potable par traitement des données issues des capteurs [HERNANDEZ, 2006]. Il a aussi permis d'élaborer une méthodologie d'aide au placement de capteurs [ORANTES et al., 2007].

Les caractéristiques principales qui ont guidé notre choix vers SALSA, sont celles de LAMDA : information aussi bien qualitative que quantitative, algorithme séquentiel, auxquels il faut rajouter la facilité d'installation et de configuration, ainsi que l'aide au dialogue avec l'opérateur. Il permet indifféremment l'apprentissage non supervisé et l'apprentissage supervisé, nécessite un nombre minimum de paramètres à régler par l'opérateur. Cependant, la méthodologie développée lors de nos travaux est générique et peut donc se satisfaire de tout autre algorithme de classification ou méthode d'isolation de fautes dans la mesure où cette étape consiste à détecter l'ensemble des fautes conduisant aux mêmes symptômes.

1.2 Méthode de diagnostic par raisonnement arrière

La méthodologie de diagnostic par raisonnement arrière est très similaire à celle par raisonnement avant, ce qui change dans le module de diagnostic c'est le fait d'effectuer une évolution temporelle passée du système par une simulation arrière effectuée avec ProDHys, offrant la possibilité supplémentaire par rapport au premier raisonnement de remonter à l'instant de la défaillance[Mokhtari et al.,06 c]. Le schéma de diagnostic devient alors :

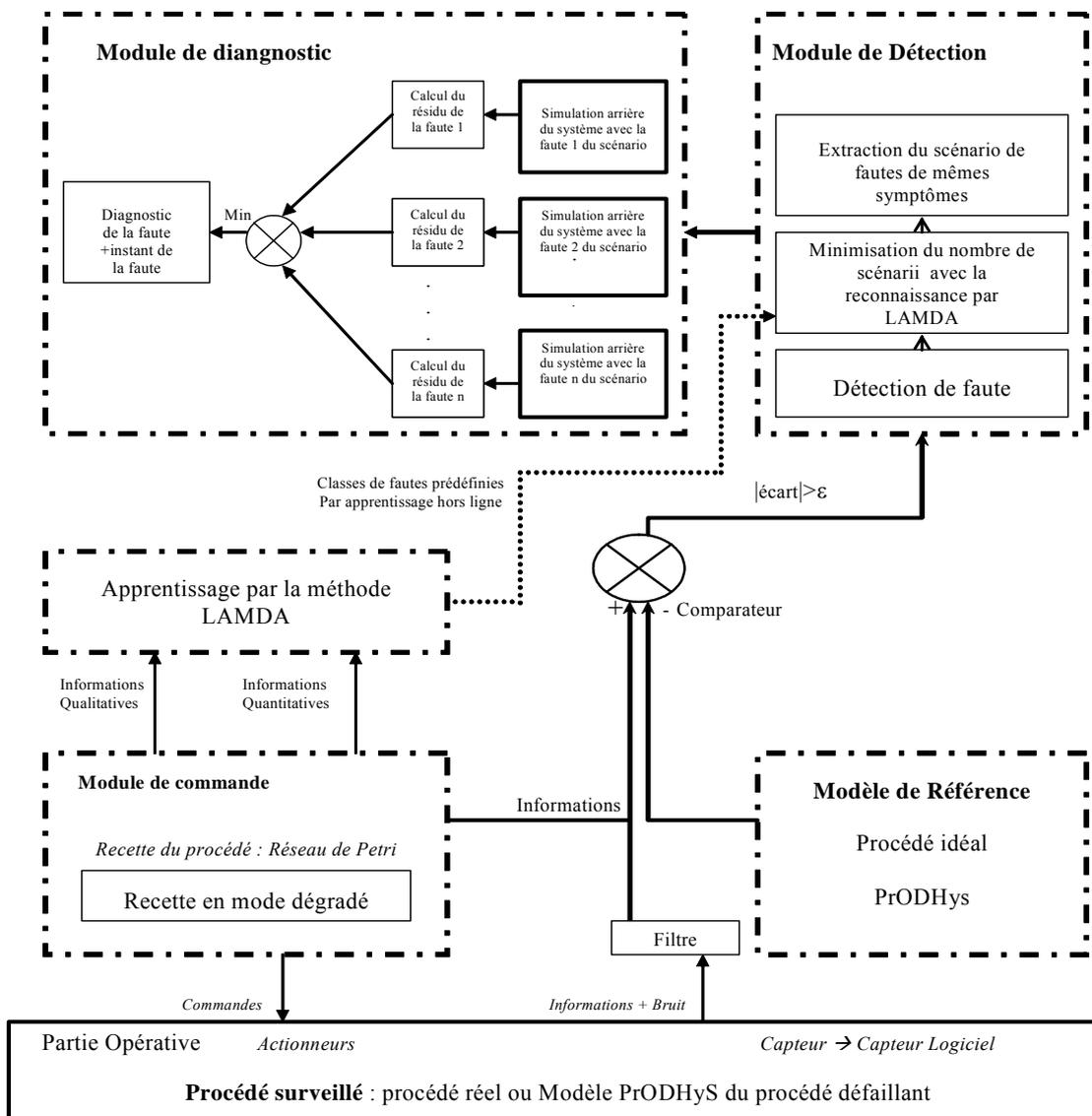


Figure 4.12 Méthode de diagnostic par raisonnement arrière

1.2.1 Réseau de Petri inversé

1.2.1.1 Définition informelle

Soit R un réseau de Petri. On appelle réseau de Petri inversé le réseau de Petri obtenu une fois que l'on a inversé tous les arcs du réseau R . Notons R^{-1} ce nouveau réseau de Petri.

1.2.1.2 Définition formelle

Soit un réseau de Petri $R = \langle P, T, Pre, Post \rangle$ où :

- P est un ensemble fini de places,
- T est un ensemble fini de transitions,
- Pre est l'application incidence avant (places précédentes),
- $Post$ est l'application incidence arrière (places suivantes).

Nous appelons réseau de Petri inversé le réseau R^{-1} tel que $R^{-1} = \langle P', T', Pre', Post' \rangle$

où :

- $P' = P$,
- $T' = T$,
- Les applications Pre' et $Post'$ sont définies telles que :

$$\forall p \in P, et \forall t \in T \begin{cases} Pre'(p, t) = Post(p, t) \\ Post'(p, t) = Pre(p, t) \end{cases}$$

1.2.2 Accessibilité entre deux marquages : deux approches duales pour les systèmes à événements discrets

L'accessibilité entre deux marquages peut être vue sous deux angles : le premier consiste à partir du marquage initial et à construire les états successeurs en tirant une à une les transitions franchissables de la liste pré-établie jusqu'à atteindre le marquage final, c'est **l'accessibilité avant**. Le marquage initial représente l'état présent alors que le marquage final représente un état futur. Quant à **l'accessibilité arrière**, le marquage final est considéré comme l'état présent et le marquage initial est vu comme un état du passé. On construit, à partir de l'état présent, les états prédécesseurs en supposant avoir franchi une à une les transitions de la liste pré-établie. Commençons par la formalisation de l'accessibilité avant en logique Linéaire.

1.2.2.1 Accessibilité avant

C'est l'accessibilité du marquage M_f par les franchissements de transitions de la liste l dans d'un réseau de Petri R muni d'un marquage initial M_0 . A partir du marquage initial et au fur et à mesure des applications successives des règles de franchissement, on consomme et on produit des jetons jusqu'à l'obtention du marquage final. On part effectivement du marquage initial (une liste de marquages initialement disponibles) pour aboutir au marquage final (une liste de marquages initialement indisponibles). Cette démarche est la plus naturelle dans les problèmes d'accessibilité. Toutefois, il est possible de procéder autrement et ce, en partant non plus du marquage initial M_0 mais du marquage final M_f et de construire ses prédécesseurs jusqu'à obtenir M_0 . C'est le principe de l'accessibilité arrière.

1.2.2.2 Accessibilité arrière

Au lieu de chercher les transitions franchissables à partir d'un marquage donné du réseau de Petri et avancer en les tirant une à une, on cherche celles qui ont dû être tirées par une démarche abductive, et on remonte dans le passé en cherchant les causes de ces franchissements.

Intuitivement, on pourrait penser que l'accessibilité arrière de M_f vers M_0 est finalement la même que l'accessibilité avant de M_0 vers M_f , c'est à dire que les marquages et les séquences explorées sont identiques. Nous allons voir que ce n'est pas le cas. Nous présenterons par la suite une formalisation de l'accessibilité arrière basée sur l'inversion du réseau de Petri ainsi qu'un résultat d'équivalence entre les deux types d'approches d'accessibilité.

1.2.2.2.1 Définition de l'accessibilité arrière

L'accessibilité arrière d'un marquage est l'ensemble des marquages M_i tels qu'il existe une séquence σ_i telle que $M_i \rightarrow M_f$. Illustrons cette notion au travers d'un exemple.

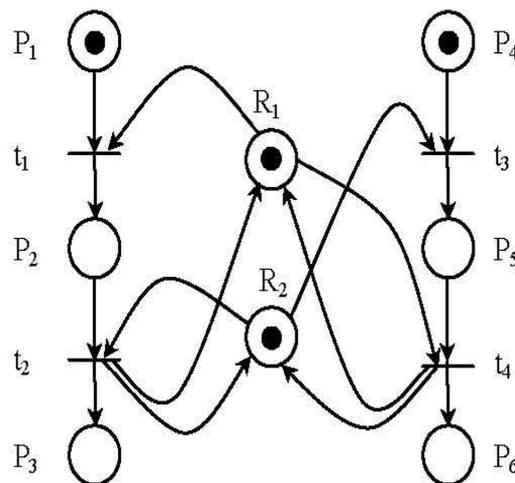


Figure 4.13 Exemple de réseau de Petri marqué

Nous voulons souligner au travers de cet exemple de réseau de Petri la différence entre l'accessibilité avant et l'accessibilité arrière. Nous étudierons pour cela le problème de l'accessibilité entre un marquage initial M_0 et un marquage final M_f en franchissant une fois chacune des transitions du réseau de Petri de la figure 4.13. M_0 est constitué d'un jeton dans chacune des places P1, P4, R1 et R2 du réseau. Quant à M_f , il est constitué d'un jeton dans chacune des places P3, P6, R1 et R2.

En partant de M_0 on trouve deux séquences menant à M_f : $M_0 \xrightarrow{t1;t2;t3;t4} M_f$ et $M_0 \xrightarrow{t3;t4;t1;t2} M_f$. On trouve aussi deux séquences menant au blocage mortel M_b (un jeton dans P2 et un jeton dans P5) $M_0 \xrightarrow{t1;t3} M_b$ et $M_0 \xrightarrow{t3;t1} M_b$. Par contre, si l'on cherche tous les prédécesseurs de M_f , on ne rencontre pas M_b (bien évidemment puisque M_b n'a pas de successeur) et seules les séquences $M_0 \xrightarrow{t3;t4;t1;t2} M_f$ et $M_0 \xrightarrow{t1;t2;t3;t4} M_f$ sont obtenues.

Bien évidemment, les deux séquences menant avec succès de M_0 vers M_f sont trouvées aussi bien par l'accessibilité avant que par l'accessibilité arrière, mais les processus de recherche ne sont pas les mêmes. De manière générale, l'accessibilité avant permet de mettre en évidence non seulement les scénarios menant du marquage initial au marquage final, mais également ceux qui ne mènent pas vers ce marquage final. Réciproquement, l'accessibilité arrière identifie les scénarios qui mènent du marquage initial au marquage final, et aussi ceux qui mène au marquage final sans passer par le marquage initial.

1.2.2.3 Equivalence entre accessibilité arrière et accessibilité avant sur le RdP inversé

Nous allons maintenant prouver que l'accessibilité arrière entre deux marquages M_0 et M_f est équivalente à l'accessibilité avant entre M_f et M_0 sur le réseau de Petri inversé. Ainsi, il n'est pas nécessaire de développer deux algorithmes d'accessibilité (avant et arrière). Il suffira de travailler sur le réseau de Petri inversé chaque fois qu'il y a un problème d'accessibilité arrière.

Nous voulons montrer que l'ensemble des séquences de franchissement et des marquages obtenus lors de l'analyse de l'accessibilité arrière sur R est le même que celui obtenu pour l'accessibilité avant sur R^{-1} (après inversion des séquences). Comme l'analyse de l'accessibilité est un processus itératif, il suffit de faire la preuve pour une itération de base entre un marquage M_i et un marquage M_{i+1} tel que $M_i \xrightarrow{t_j} M_{i+1}$

Dans R , le fait que M_{i+1} soit accessible à partir de M_i par le franchissement de t_j implique que $M_i = M_{i+1} - \text{Post}(t_j) + \text{Pre}(t_j)$ et $M_{i+1} \geq \text{Pre}(t_j)$. Comme $M_i - \text{Pre}(t_j) = M_{i+1} - \text{Post}(t_j)$, $M_i \geq \text{Pre}(t_j)$ implique $M_{i+1} \geq \text{Post}(t_j)$.

Dans R^{-1} , nous avons donc $M_{i+1} \geq \text{Pre}'(t_j)$ et $M_i = M_{i+1} - \text{Pre}'(t_j) + \text{Post}'(t_j)$, ce qui exprime l'accessibilité avant de M_i à partir de M_{i+1} par t_j dans R^{-1} .

Réciproquement, si nous avons M_i accessible à partir de M_{i+1} par t_j dans R^{-1} , alors $M_i = M_{i+1} - \text{Pre}'(t_j) + \text{Post}'(t_j)$ et $M_{i+1} \geq \text{Pre}'(t_j)$.

Nous avons donc :

$M_i = M_{i+1} - \text{Post}(t_j) + \text{Pre}(t_j)$ et $M_i \geq \text{Pre}(t_j)$ qui expriment l'accessibilité de M_i à partir de M_{i+1} par t_j dans R .

1.2.3 Extention par ProHDyS du raisonnement arrière aux systèmes hybrides

Si l'utilisation du raisonnement arrière des RdP pour effectuer le diagnostic dans le cas de systèmes purement discrets commence à être connue, il n'en est pas de même dans le cas de systèmes hybrides. Nos travaux ont montré la possibilité par les RdPDO tels qu'ils sont utilisés dans ProDHyS d'effectuer la résolution de systèmes algèbro-différentiels en remontant dans le temps. Cette propriété va donc permettre à terme d'étendre l'utilisation du chaînage arrière pour le diagnostic des systèmes hybrides.

Le principe de la simulation arrière réside dans l'inversion des directions de tous les arcs du RdP en prenant comme condition initiale l'état du système à la fin de la simulation directe (moment de la détection par l'algorithme de classification d'une défaillance, ou tout du moins d'un symptôme). Le Réseau de Petri décrit sur la figure 4.14 est le réseau inverse de

la figure 4.4 vu dans la première partie. Il permettra d'isoler et de diagnostiquer la faute parmi celles pouvant affecter le réservoir 2 par comparaison des évolutions.

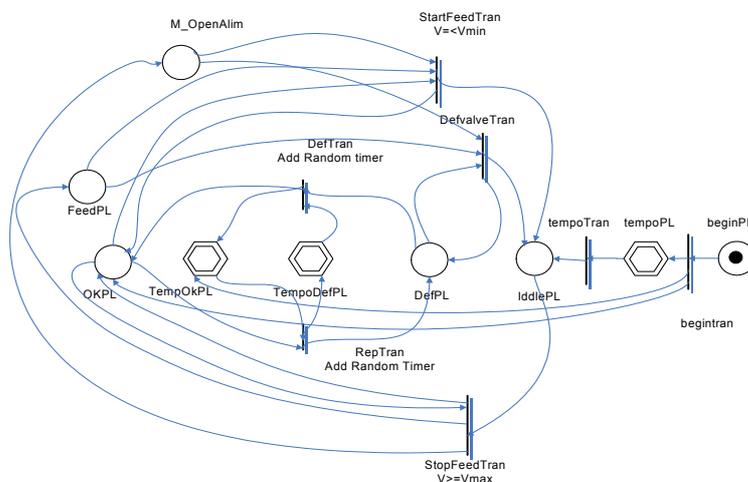


Figure 4.14 Réseau de Petri inverse

Les résultats obtenus sont donnés sur la figure 4.15

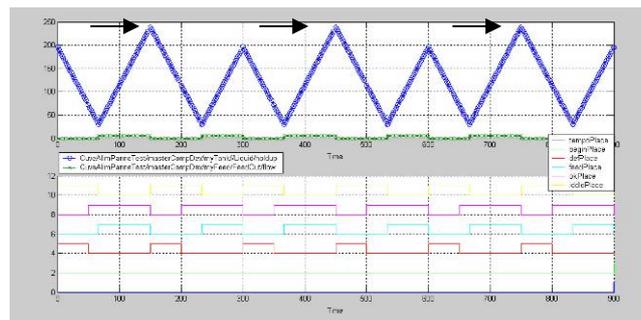


Figure 4.15 Résultats de la simulation arrière

Le graphe du bas représente les états de marquage des places du Réseau de Petri inverse. Le graphe du haut donne l'évolution du niveau d'eau dans la cuve obtenue par résolution du système d'équations algèbro-différentielles avec comme valeurs initiales les valeurs finales observées sur le processus. Le graphe de la simulation arrière nous renseigne bien sur l'état du blocage de l'alimentation en ouverture par l'apparition des pics (dépassement du niveau 185 initial) aux différents instants de défaillance.

Comme précédemment, lorsqu'il y a plusieurs possibilités de défaillances, cette technique se heurte à explorer toutes ces possibilités. Pour éviter l'explosion des possibilités à examiner par simulation arrière, nous pouvons de même utiliser la classification des données (mesures issues des capteurs ou prétraitement des mesures) pour restreindre l'ensemble des possibilités à celles donnant les mêmes symptômes, la simulation arrière venant identifier la faute parmi celles isolées à l'étape précédente ainsi que l'instant d'apparition de la faute.

Durant la phase d'implémentation de la méthode sur la plate-forme PrODHys nous nous sommes heurtés à plusieurs problèmes tels que l'initialisation des variables, la diminution du pas d'intégration, mais le plus important concernait le traitement des systèmes à

dynamique ultra rapide qui surviennent lors d'utilisation d'appareils à pression ou à inversion de flux ; ce problème n'a pas pu être résolu dans notre méthode par raisonnement arrière car il réside dans le noyau interne de la plate-forme PRODHYS auquel nous n'avons pas accès. Nous donnons dans ce qui suit un complément sur la difficulté du traitement des équations différentielles à dynamiques ultra-rapides qui expliquerait le problème rencontré puisqu'il n'apparaît que lorsque les équations résolues sont de type bilan de quantité de mouvement incluant donc des équations de pression.

1.2.3.1 Le traitement des systèmes à dynamiques ultra rapides (fast transient)

Dans la communication [Jarvis et Pantelides,92], un exemple de modèle d'un problème de génie chimique tout à fait banal est présenté : il s'agit d'une cuve remplie de gaz à une pression supérieure à l'extérieur. Cette cuve se vide progressivement vers l'extérieur au travers d'une vanne tandis que la pression externe augmente linéairement. A un temps donné, la pression externe devient supérieure à la pression interne, le débit s'inverse et la cuve se remplit de nouveau la figure 4. 16 présente cet exemple.

$$\left\{ \begin{array}{l} \frac{dM}{dt} = -F \\ p.V = M.R.T \\ F = k.\sqrt{|p - p^0|}.\text{signe}(p - p^0) \\ p^0 = 0.9 + 0.1.t \end{array} \right. \quad 4.13$$

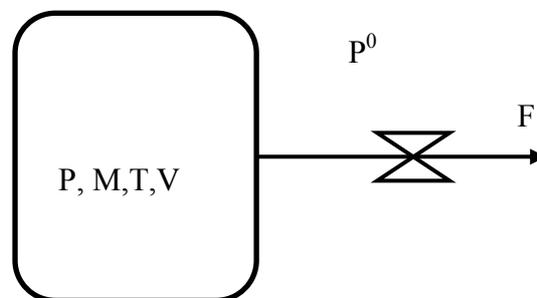


Figure 4.16 Présentation de l'exemple de la cuve

A t^* où le débit s'inverse et où dans notre cas le pas d'intégration est négatif, la dérivée du débit par rapport au temps devient infinie :

$$\frac{\partial F}{\partial t} \rightarrow -\infty$$

Les auteurs précisent que ce type de comportement entraîne des difficultés d'intégration :

- Le pas d'intégration est diminué de façon drastique pour assurer la tolérance sur les variables,
- Des valeurs importantes peuvent apparaître dans l'opérateur dynamique conduisant à des erreurs lors de la factorisation ou la résolution du système linéaire,

- Enfin, l'opérateur dynamique peut devenir singulier à l'approche de t^* provoquant également un échec de l'intégration.

Pour éviter ce genre de problèmes, les auteurs proposent de changer de variable indépendante à l'approche de ce genre de problème. Ainsi ici, la variable F deviendrait variable indépendante entre t^- et t^+ alors que t deviendrait une nouvelle inconnue du système,, ce changement de variable indépendante devant se faire de manière automatique et transparente.

Le solveur actuel Disco de la plate-forme PrODHyS ne permet pas actuellement ce changement de variable indépendante automatique.

La méthode de changement suivante a été adoptée :

Si l'on pose θ la variable indépendante interne au solveur, on peut écrire le système 4.13 sous la forme :

$$\left\{ \begin{array}{l} \frac{\partial t}{\partial \theta} = 1 \\ \frac{\partial M}{\partial \theta} \cdot \frac{\partial \theta}{\partial t} = -F \text{ soit } \frac{\partial M}{\partial \theta} = -F \\ p.V = M.R.T \\ F = k.\sqrt{|p - p^0|}.\text{signe}(p - p^0) \\ p^0 = 0.9 + 0.1.t \end{array} \right. \quad 4.14$$

Avec pour inconnues t, M, F, P, P^0

Pour intégrer ce système avec F pour variable indépendante, il suffit de changer la première équation par $\frac{\partial F}{\partial \theta} = 1$, le reste du système restant inchangé.

Un problème persiste dans ce cas en particulier au tout début de l'intégration et n'a pas encore été résolu.

2 Conclusion

Dans ce chapitre, nous avons proposé une méthodologie pour le diagnostic basée sur l'alliance et non l'opposition de deux approches en montrant le lien entre la classification exploitant les données issues de la simulation et la modélisation par réseaux de Petri. Notre objectif était de proposer une démarche de diagnostic plus affinée afin d'être plus précis dans la détection de fautes en plus dans un temps de calcul réduit.

La méthode de diagnostic, s'appuie sur le pouvoir de modélisation de la plate-forme PrODHy, plate-forme de simulation générique pouvant toucher plusieurs domaines d'application. Nous avons présenté aussi les différents paramètres de la méthode LAMDA à régler pour avoir une classification acceptable. Les résultats de cette classification sont primordiaux pour la conception du modèle de comportement du système sur lequel est basé le

diagnostic. Des travaux (thèse de Claudia ISAZA) ont permis de faciliter cette tâche en élaborant des indices de mesure de la qualité d'une partition.

Le chapitre suivant va nous permettre d'illustrer la méthode au travers d'un exemple d'application. Notons que dans le chapitre suivant la méthode détaillée est celle par raisonnement avant, car pour l'instant et malgré l'énorme travail et le temps passé dans la simulation arrière, le problème persiste encore. Cependant, si ce problème est résolu, la transposition des résultats que nous avons obtenus avec la simulation avant est immédiate et on gardera ce raisonnement comme perspective directe à notre travail.

Chapitre 5 Application à un système hydraulique

Ce chapitre a pour but de montrer un exemple illustratif complet de l'application de la méthodologie de diagnostic proposée dans le chapitre précédent, il détaille aussi le processus de modélisation et simulation du procédé à partir de la bibliothèque d'appareils proposée par PrODHyS, et aussi l'exploitation des résultats de cette simulation par la classification. Le système hydraulique considéré est inspiré d'un benchmark défini par l'AS 193 « Diagnostic des systèmes hybrides » (www.univ-lille1.fr/lail/AS193).

1 Description du procédé considéré

Le système considéré (figure 5.1) est constitué de 2 réservoirs cylindriques **R1** et **R2**, reliés entre eux par 2 conduites **C1** et **C2**, munies respectivement des vannes TOR (tout ou rien) **V3** et **V4**. La conduite **C1** est située au fond des réservoirs (référence des niveaux d'eau), la conduite **C2** est située à une hauteur de **50 cm**. Les 2 pompes **P1** et **P2** sont identiques. Elles sont contrôlées (système de commande ou utilisateur) et sont commandées en tout ou rien. La dynamique des pompes est rapide, le transitoire peut être négligé. On peut donc supposer que l'on a :

$$\text{Débit P1} = \text{QP1} = 0 \text{ (pompe arrêtée)}$$

ou

$$\text{débit P1} = \text{QP1} = D \text{ (pompe alimentée)}.$$

Les vannes TOR : **V2**, **V3** et **V4** sont contrôlées (soit par un système de commande, soit manuellement par l'utilisateur). **V1** est une vanne qui reste toujours fermée en fonctionnement normal. Cette vanne permet de simuler une fuite du réservoir R1 (fonctionnement défaillant).

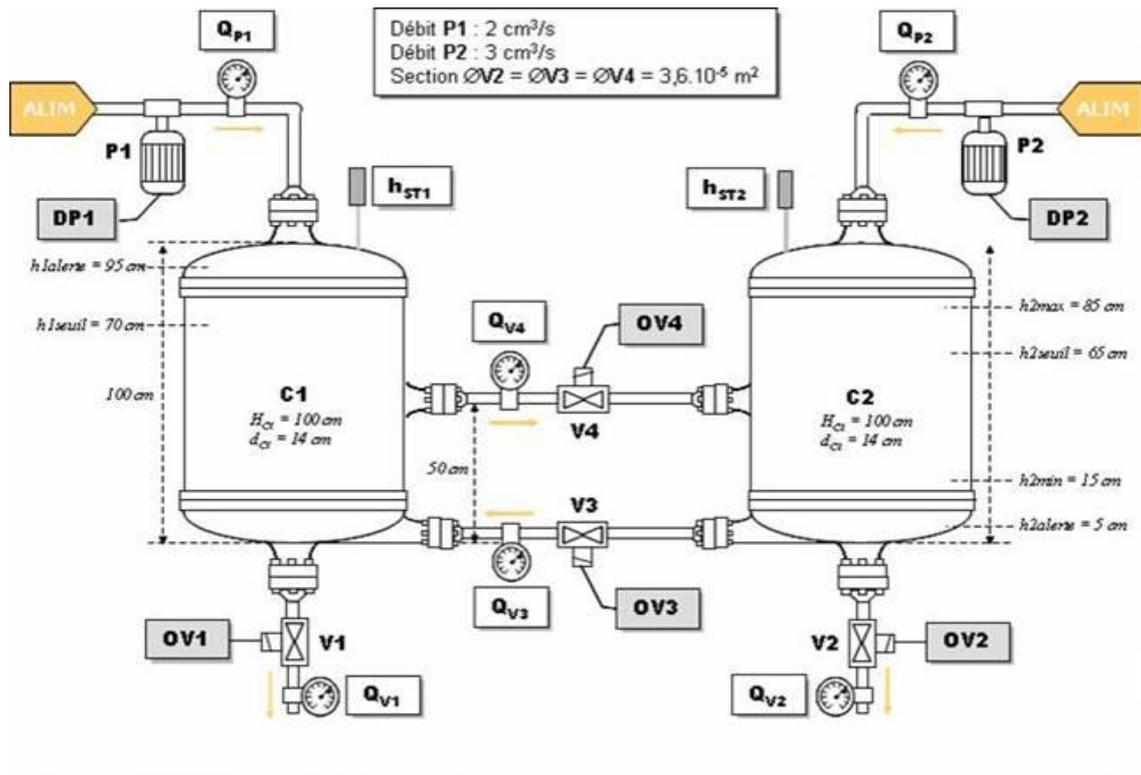


Figure 5.1 Le système hydraulique à deux réservoirs

Le système de commande a pour objectif principal de maintenir le niveau de liquide h_2 dans C2 entre les hauteurs h_{2min} et h_{2max} par action sur la vanne V4. Toutefois, la vanne V3 peut être ouverte pour éviter une rupture au niveau de C2 ($h_2 \leq h_{2\text{ alerte}}$).

1.1 Schémas d'instrumentation

Plusieurs schémas d'instrumentation (capteurs) peuvent être considérés :

1) **Cas 1** : Instrumentation maximale (7 capteurs) :

Débit QA en sortie de P1

Débit QB en sortie de P2

Débit Q2 traversant V2

Débit Q3 traversant V3

Débit Q4 traversant V4

Hauteurs h_1 et h_2 d'eau dans les réservoirs R1 et R2

2) **Cas 2** : 2 capteurs

Hauteurs h_1 et h_2 d'eau dans les réservoirs R1 et R2

3) **Cas 3** : 2 capteursDébit Q_2 traversant V_2 Hauteur h_1 (ou exclusif h_2)4) **Cas 4** : 1 capteur1 capteur : au choix, Hauteur h_1 , h_2 , débit Q_2

Dans notre cas, nous avons opté pour le cas 2 avec 2 capteurs et nous nous intéressons seulement aux hauteurs h_1 et h_2 dans les deux réservoirs. Sachant que toutes les séquences d'états discrets (ouvertures de vannes, et commandes de pompes) peuvent donc être envisagées.

1.2 Conditions initiales

Initialement, on peut considérer que les deux pompes sont à l'arrêt et toutes les vannes sont fermées. Quatre conditions initiales peuvent être envisagées en ce qui concerne les niveaux d'eau. Ces conditions sont représentées ci-dessous :

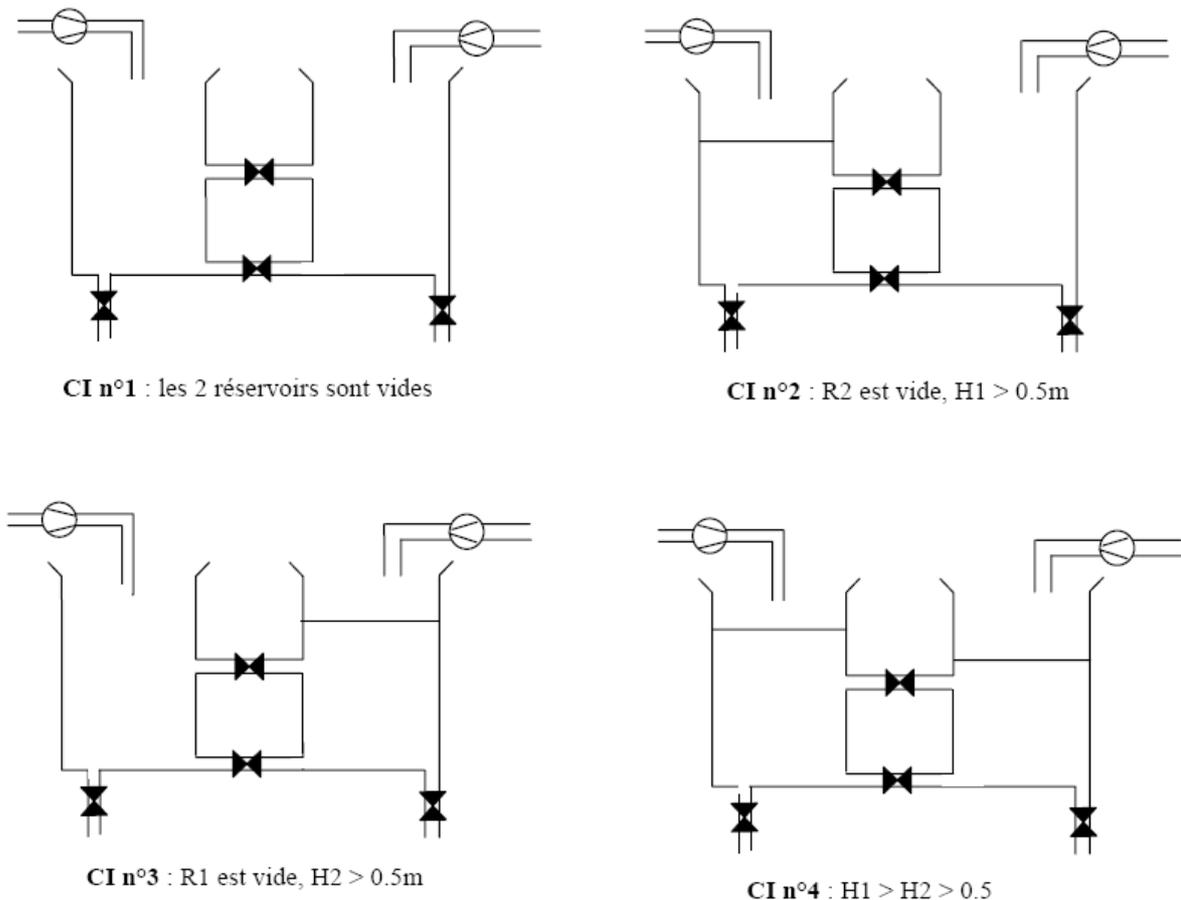


Figure 5.2 Les quatre cas des conditions initiales

1.3 Les défaillances considérées

- Défaillance de la pompe P1 ou P2: la pompe s'arrête alors qu'elle devrait fonctionner (Éventuellement l'inverse aussi).
- Défaillance des vannes : vanne bloquée ouverte ou bloquée fermée
- Défaillance procédé : fuite réservoir 1 (vanne V1 ouverte)

1.4 Les équations dynamiques

Sous certaines hypothèses (écoulement laminaire, non compressibilité du liquide) deux équations permettent de décrire le comportement dynamique du système (évolution des variables continues).

Le débit Q_{12} circulant dans la conduite située entre les deux réservoirs $R1$ et $R2$ (au travers d'une vanne ouverte) est donné par la loi de Toricelli :

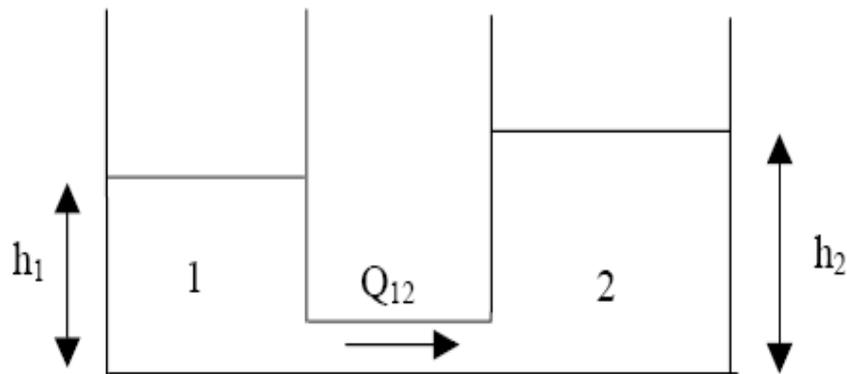


Figure 5.3 Débit traversant les deux réservoirs

$$Q_{12} = A \cdot \text{sign}(h_1 - h_2) \sqrt{2 \cdot g \cdot |h_1 - h_2|} = -Q_{21} \quad 5.1$$

A: section du conduit

g : constante de gravité : $g = 9.81 \text{ m/s}^2$

h_1 et h_2 : hauteur d'eau dans les réservoirs 1 et 2

Remarque : La loi de Toricelli reste valable dans la configuration suivante :

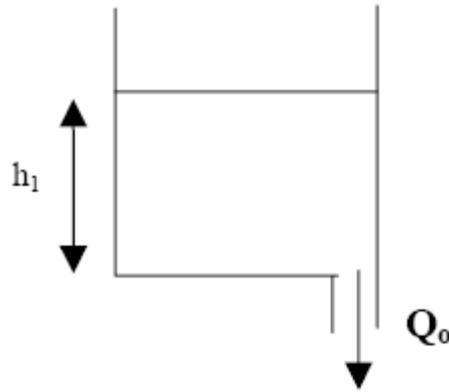


Figure 5.4 Débit dans un seul réservoir

Dans ce cas, on prend $h_2 = 0$. Ce qui donne :

$$Q_0 = A\sqrt{2.g.h_1}$$

L'équation différentielle décrivant l'évolution du volume dans un réservoir est donnée par :

$$\dot{V} = S.\dot{h} = \sum Q_{in} - \sum Q_{out}$$

$\sum Q_{in}$ = somme des débits entrant.

$\sum Q_{out}$ = somme des débits sortant.

Exemple :

Soit la configuration suivante :

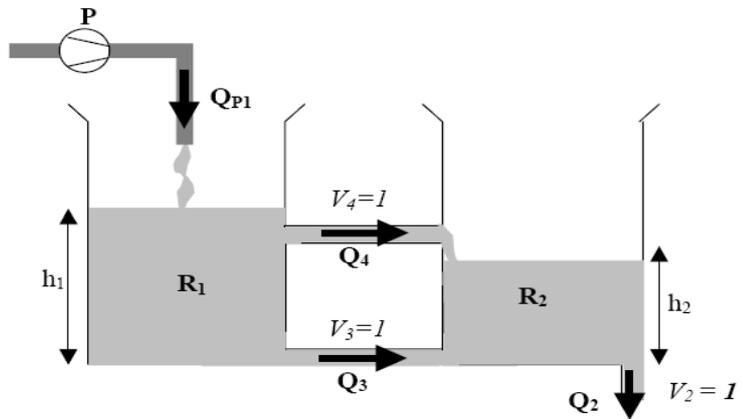


Figure 5.5 Exemple de configuration

Les équations sont :

$$S.\dot{h} = Q_{P1} - Q_4 - Q_3 \quad 5.2$$

$$S.\dot{h}_2 = Q_4 - Q_3 - Q_2 \quad 5.3$$

$$Q_3 = A.\sqrt{2.g.\sqrt{(h_1 - h_2)}} \quad 5.4$$

$$Q_4 = A.\sqrt{2.g.\sqrt{(h_1 - 0.5)}} \quad 5.5$$

$$Q_2 = A.\sqrt{2.g.\sqrt{(h_2)}} \quad 5.6$$

2 Processus de modélisation du procédé

La modélisation d'un procédé complet au sein de la plate-forme *PrODHyS* requiert plusieurs opérations de la part de l'utilisateur. Il doit :

- définir la recette afin de spécifier la séquence opératoire à suivre. La recette est décrite selon le formalisme RdPDO.
- décrire le procédé en choisissant les appareils qui le composent et en précisant leur configuration et leur topologie ainsi que les connexions qui les relient. Pour ce faire, une procédure générale et systématique doit être suivie ; elle comprend les étapes suivantes :

1. Déclarations des objets
2. Création du lien avec l'univers thermodynamique
3. Création du procédé
4. Création de la recette
5. Définition des constituants intervenant dans le procédé
6. Définition du modèle thermodynamique
7. Définition des réactions chimiques éventuelles
8. Sélection des appareils et définition de leur configuration (au fur et à mesure qu'ils sont définis, les appareils et leur réseau de Petri sont ajoutés respectivement à l'appareil composé de simulation et à la recette)
9. Établissement des connexions
10. Description de la recette
11. Création du solveur
12. Application de la configuration globale de la matière à tous les éléments du procédé
13. Initialisation des appareils (pression, température, composition, flux, etc.)
14. Mise en place du marquage initial des RdP
15. Simulation
16. Destruction des objets créés

Actuellement, la description de la recette, du procédé et de la matière se fait entièrement en *C++*. Cependant, une interface graphique dédiée à la description de la recette est actuellement en cours de développement; elle est supposée s'étendre, à terme, au procédé et à la matière. Elle devrait ainsi permettre une utilisation plus conviviale de la plate-forme de simulation.

2.1 Modélisation du procédé

Le modèle adopté pour représenter le procédé décrit ci-dessus apparaît sur la figure 5.6. Il comporte:

- deux bacs (de type StorageTank) ;
- deux alimentations matière (de type MaterialFeed);
- deux actionneurs de port (de type PortActuator) pour contrôler le soutirage des bac 1 et 2;
- deux vannes à inversion de flux (de type ValveTwoWay) ;
- 4 détecteurs de niveau (de type LevelDetector) indiquant respectivement un dépassement du niveau haut et bas dans les bacs 1 et 2 ;
- l'environnement (de type surroundings).

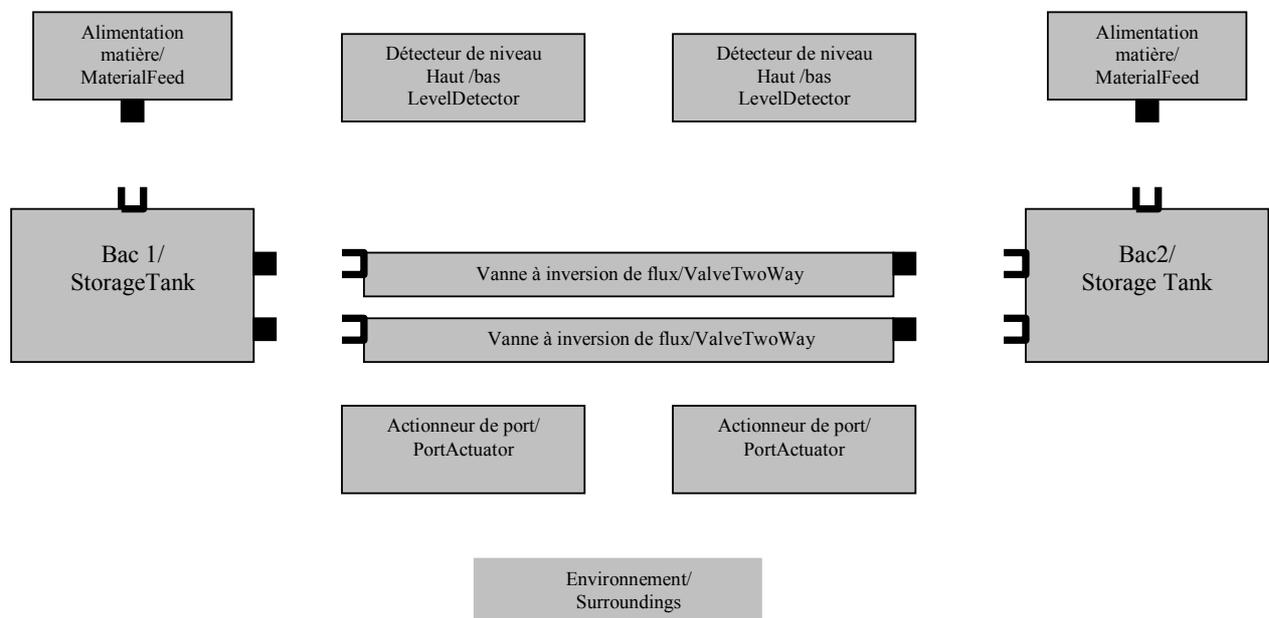


Figure 5.6 Modèle associé à l'exemple d'application

2.2 Modélisation de la recette

Dans le cadre de cette application, la séquence opératoire à réaliser est décrite sur la figure 5.7

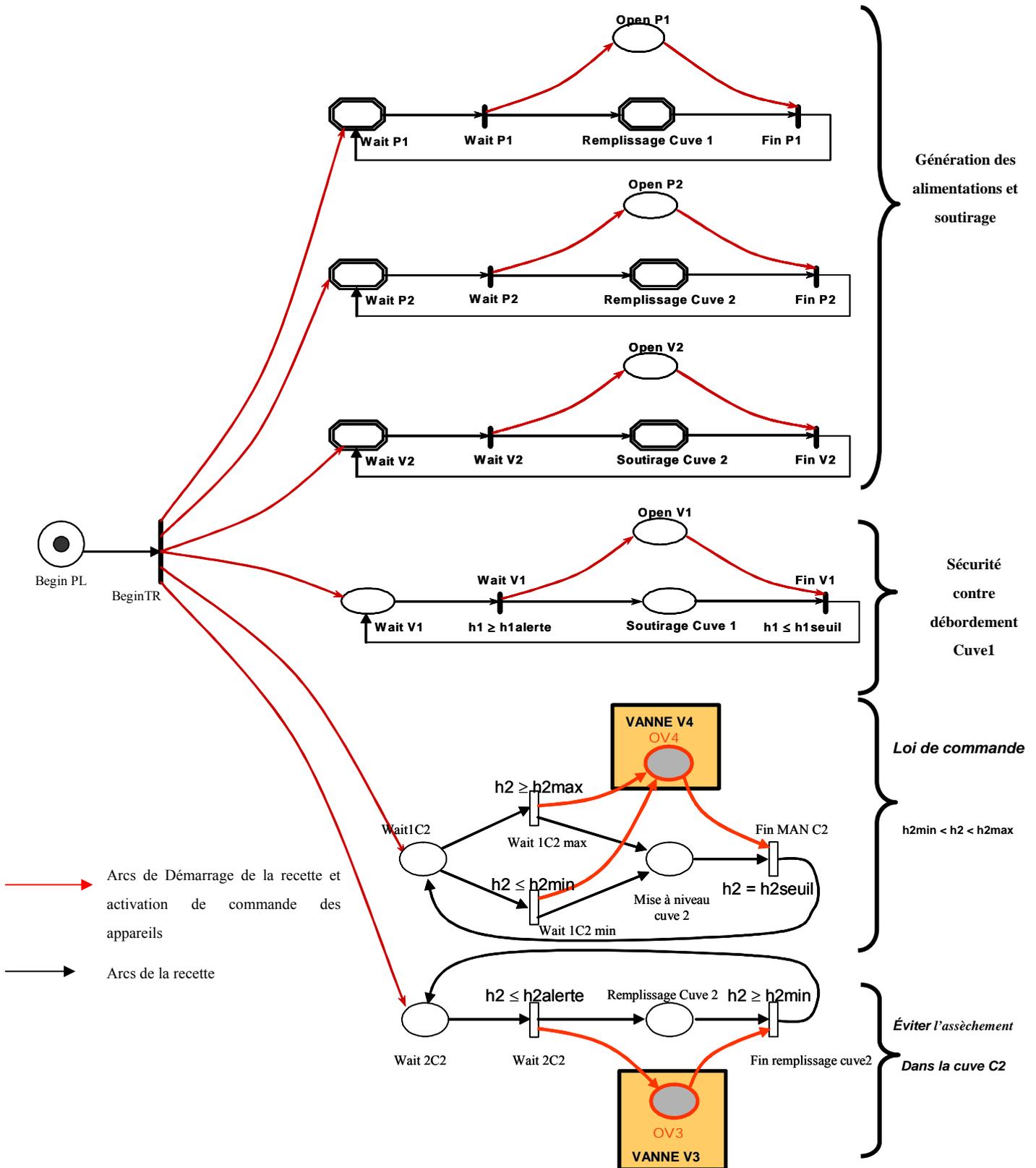


Figure 5.7 Recette associée au système

2.2.1 Génération des alimentations et soutirages

La gestion des alimentations de pompe P1 et P2 ainsi que le soutirage de la vanne V2 se fait par le réseau de Petri de la figure 5.8 :

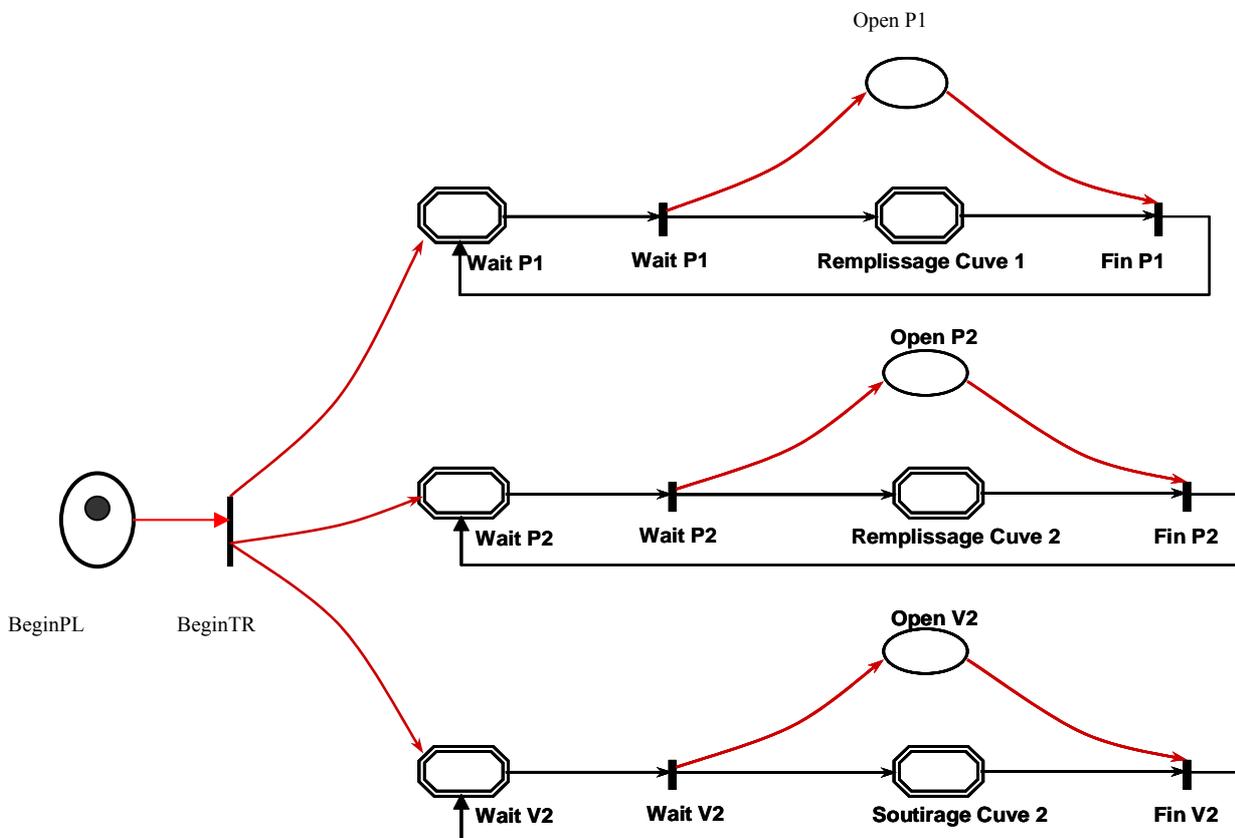


Figure 5.8 Génération des alimentations et soutirage

L'ouverture des pompes P1 et P2 ainsi que la vanne V2 se fait par le marquage des places de commande *OpenP1*, *OpenP2* et enfin *OpenV2*. La durée d'ouverture ou de fermeture de ces pompes et vanne est gérée par les places temporisées représentées sous forme d'hexagone. Par exemple tant que la durée de temporisation dans la place *WaitP1* n'est pas écoulée la pompe P1 ne sera pas ouverte, il en est de même pour la place *remplissage Cuve1* qui gère la fermeture de la pompe1. Ces places temporisées nous permettent de modéliser les fautes sur les pompes et les vannes en supposant que la temporisation de ces places est infinie pour la simulation des blocages en ouverture ou en fermeture ; si par exemple la temporisation de la place *WaitP1* est infinie, la place d'ouverture de la pompe P1 ne sera jamais marquée ce qui signifie qu'elle est bloquée en fermeture et si par exemple la temporisation de la place *Remplissage Cuve1* est infinie, la place *OpenP1* reste toujours marquée ce qui signifie que la pompe P1 reste bloquée en ouverture ; le même raisonnement est établi pour la simulation des fautes sur P2 et sur V2.

2.2.2 Sécurité contre le débordement de la cuve 1

Afin d'assurer la sécurité contre le débordement de la cuve1 nous avons introduit le réseau de Petri de la figure 5.9 :

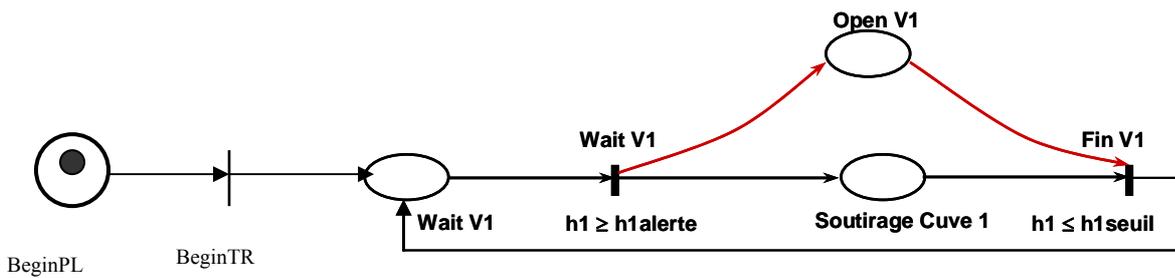


Figure 5.9 RdP de sécurité contre le débordement de la cuve 1

L'ouverture de la vanne V1 est gérée par la détection de seuil sur la transition *WaitV1* : si la hauteur du liquide n'a pas atteint *h1alerte* la vanne V1 ne sera pas ouverte et une fois ouverte elle ne sera fermée que si la hauteur de *h1* est inférieure ou égale à *h1seuil*. On peut aussi modéliser des fautes sur V1 en supposant que *h1alerte* est infini ce qui va engendrer le blocage de la vanne V1 en ouverture.

2.2.3 Loi de commande $h2min < h2 < h2max$

Afin de maintenir le niveau du liquide *h2* dans la cuve 2, nous avons introduit dans notre système le réseau de Petri de la figure 5.10 :

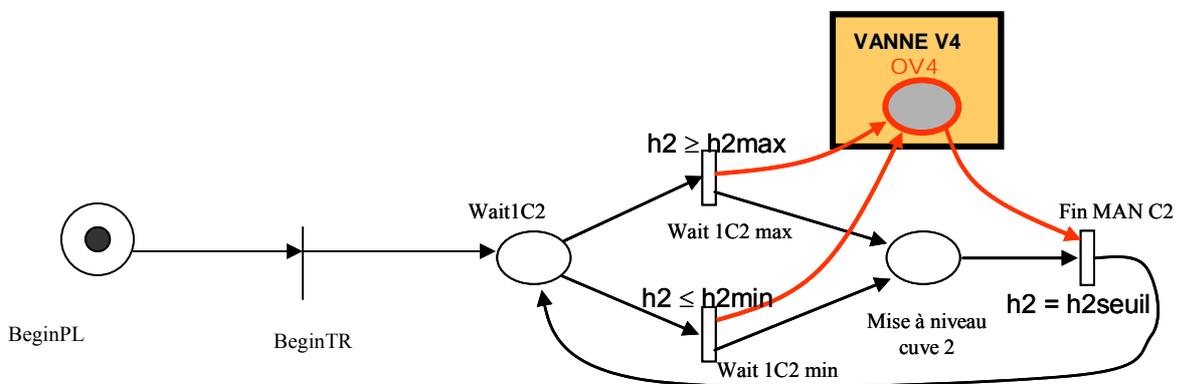


Figure 5.10 Loi de commande $h2min < h2 < h2max$

L'ouverture de la vanne 4 est gérée par la détection des seuils *h2max* et *h2min* sur les transitions *Wait 1C2max* et *Wait1C2min* respectivement. La fermeture se fait par la détection du niveau *h2seuil* sur la transition *Fin MAN C2* ce qui va permettre de fixer le niveau du liquide entre *h2min* et *h2max*.

2.2.4 Éviter l'assèchement dans la cuve C2

La figure 5.11 nous montre le réseau de Petri utilisé afin d'éviter l'assèchement dans la cuve 2 :

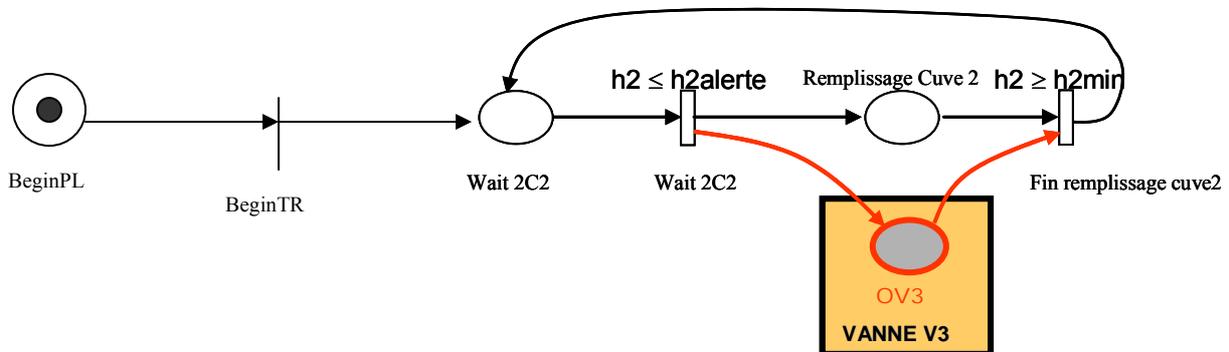


Figure 5.11 Rdp pour éviter l'assèchement dans la cuve 2

Si le seuil $h2_{alerte}$ est détecté la vanne V3 s'ouvre pour éviter la l'assèchement du liquide et se ferme une fois que la hauteur atteint $h2_{min}$.

2.3 Simulation

Différents scénarii peuvent être simulés par action sur les pompes P1 et P2 et la vanne V2 ; nous allons voir tout d'abord les résultats de la simulation sans prendre en compte les fautes comme nous le montre la figure 5.12 :

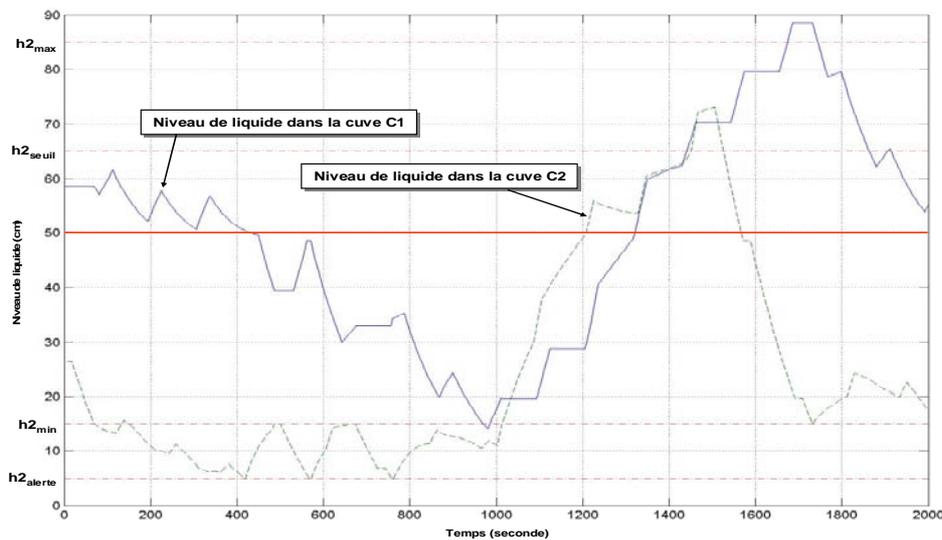


Figure 5.12 Niveaux du liquide dans le procédé en fonctionnement normal

Nous remarquons que la loi de commande est bien vérifiée : il n'y a pas de dépassement de seuil dans les deux cuves et dans l'intervalle de temps entre 150s et 1000s il y a eu intervention du réseau de Petri pour éviter la rupture dans la cuve 2 (figure 5.11) d'où l'oscillation du niveau $h2$ entre $h2_{alerte}$ et $h2_{min}$.

Comme nous l'avons présenté précédemment, nous pouvons simuler plusieurs fautes sur le système par action sur les temporisations des places ou sur les seuils fixés, nous détaillerons dans ce qui suit les différentes fautes présent en compte sur notre système.

2.3.1 Simulation de fautes

Nous avons pris en compte 5 fautes majeures dans le système qui sont :

- Pompe P1 bloquée ouverte (P1BO).
- Pompe P2 bloquée ouverte (P2BO).
- Pompe P2 bloquée fermée (P2BF).
- Vanne V2 bloquée ouverte (V2BO).
- Vanne V2 bloquée fermée (V2BF).

Le choix de ces fautes est justifié par la similitude des symptômes engendrés ce qui va nous permettre par la suite d'étudier la capacité de la méthodologie proposée à détecter et discriminer des fautes présentant des symptômes de nature similaire.

D'autres fautes peuvent d'être simulées mais il n'était pas nécessaire de les inclure vu la facilité avec lesquelles elles peuvent être détectées.

2.3.1.1 Pompe P1 bloquée ouverte (P1BO).

Pour simuler la pompe P1 bloquée en ouverture, nous avons mis la temporisation de la place *Remplissage cuve 1* (figure 5.8) à l'infini et cela après franchissement de la transition *Wait P1* ce qui signifie que la place de commande d'ouverture de la pompe P1 *Open P1* reste toujours marquée d'où le blocage de la pompe P1 en ouverture comme nous le montre la figure 5.13.

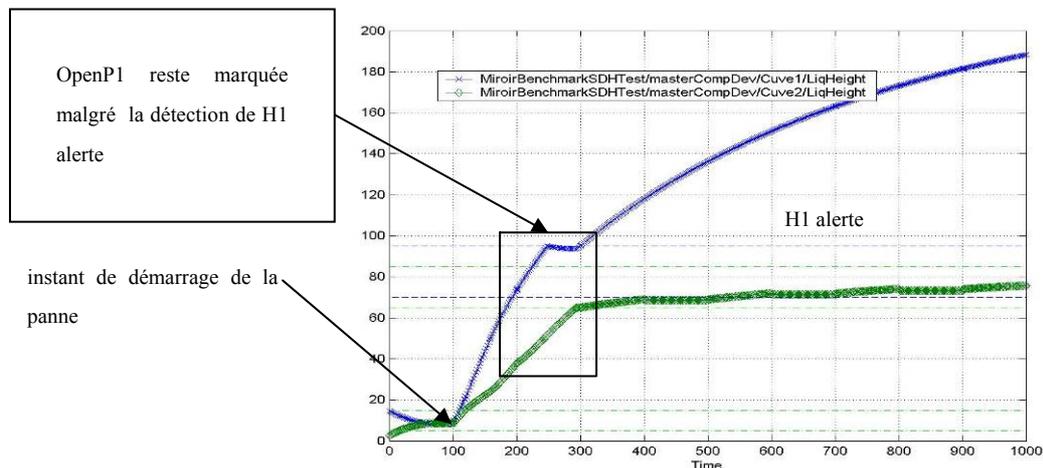


Figure 5.13 Simulation de la pompe P1 bloquée en ouverture

Nous avons choisi un temps de simulation assez grand pour bien montrer l'élévation du niveau du liquide mais la détection de la faute proprement dite sera effectuée à partir de la fenêtre temporaire du moment de détection de la faute comme nous le montre le rectangle sur la figure 5.13, on fera de même pour toutes les autres fautes qui suivent.

2.3.1.2 Pompe P2 bloquée ouverte (P2BO)

Comme pour la pompe P1, pour bloquer la pompe P2 en ouverture il suffit de mettre à l'infini la temporisation de la place *Remplissage cuve 2* du réseau de Petri de la figure 5.8. Les résultats obtenus sont illustrés dans la figure 5.14.

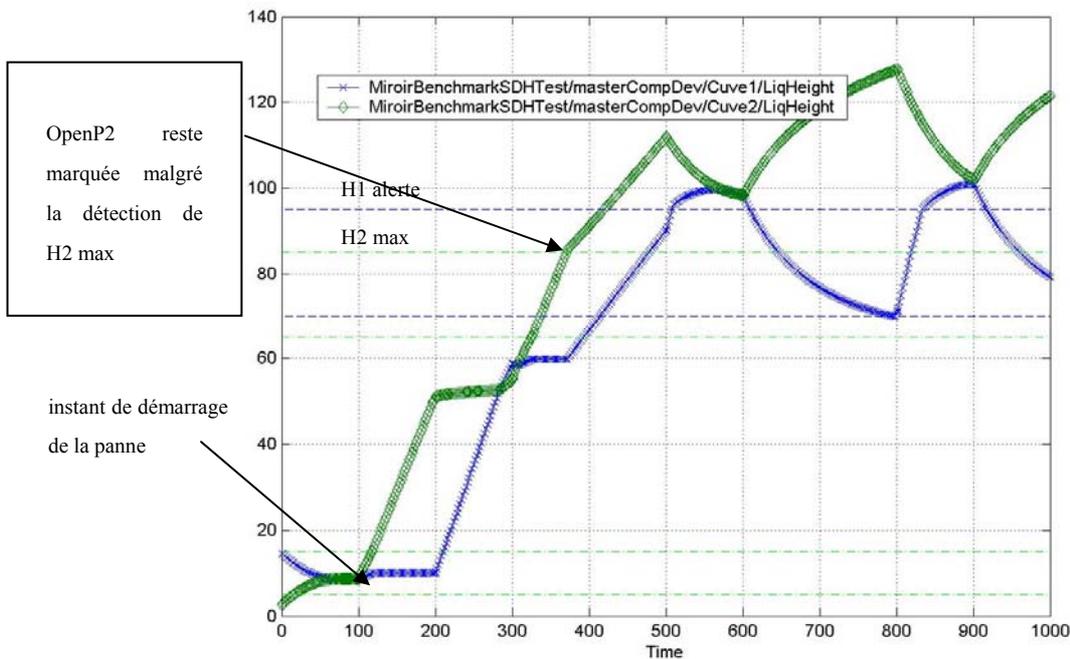


Figure 5.14 Simulation de la pompe P2 bloquée en ouverture

Nous remarquons que les deux seuils $h1_{alerte}$ et $h2_{max}$ sont largement dépassés ce qui est cohérent vu l'inversion de flux qui va s'effectuer par l'intermédiaire de la vanne 4 une fois $h2$ supérieur à $h1$ d'où le dépassement dans la cuve 1.

2.3.1.3 Pompe P2 bloquée fermée (P2BF).

Pour bloquer la pompe P2 en fermeture, il faut mettre la temporisation de la place *Wait P2* à l'infini donc la transition *WaitP2* ne sera jamais franchie et la place de commande d'ouverture de la pompe P2 jamais marquée, ce qui signifie que la pompe 2 reste toujours fermée.

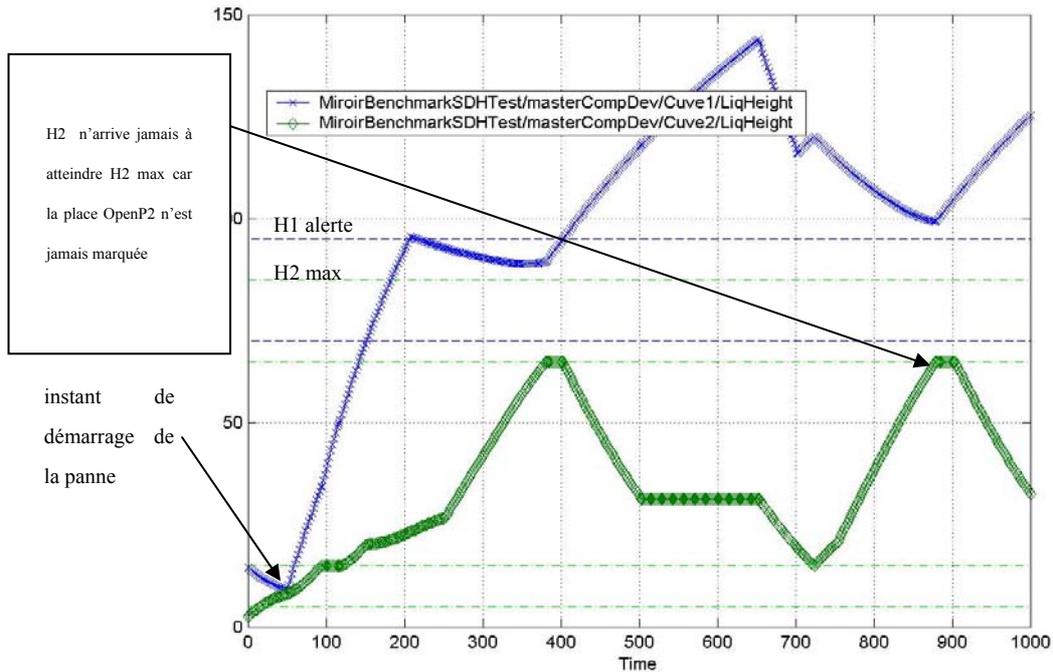


Figure 5.15 Simulation de la pompe P2 bloquée fermée

2.3.1.4 Vanne V2 bloquée ouverte (V2BO).

C'est le même principe que celui du blocage des pompes en ouverture ; cette fois-ci, il faut mettre la temporisation de la place *Soutirage Cuve 2* du réseau de Petri de la figure 5.8 à l'infini. Les résultats de la simulation sont décrits sur la figure 5.16. Nous remarquons un assèchement de la cuve 2 à cause de l'ouverture constante de la vanne V2

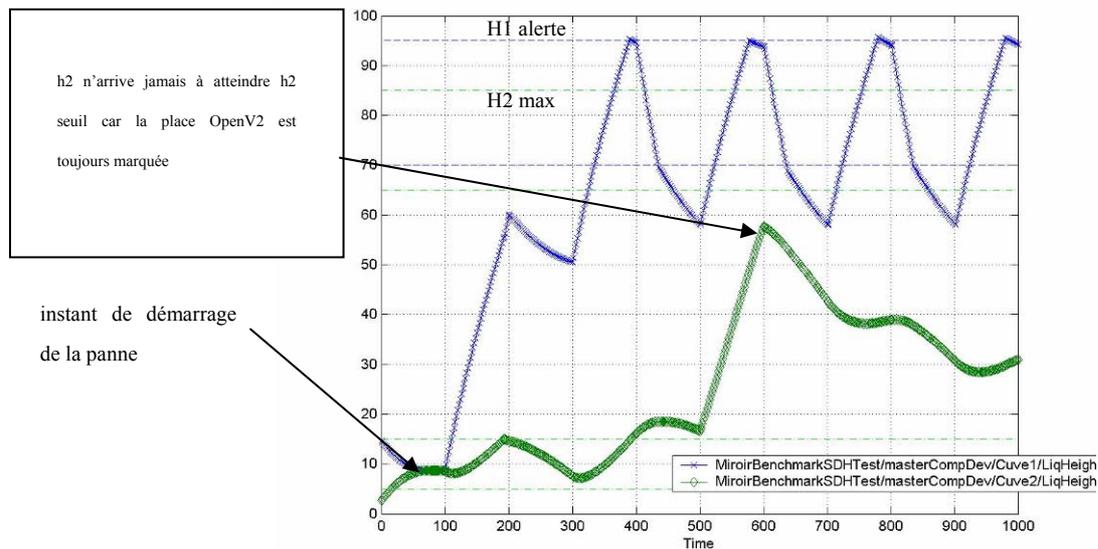


Figure 5.16 Simulation de la vanne V2 bloquée ouverte

2.3.1.5 Vanne V2 bloquée fermée (V2BF)

Il s'agit de fixer la temporisation de la place *Wait V2* à l'infini (figure 5.8) et de ce fait empêcher le franchissement de la transition *Wait V2* ce qui va bloquer la vanne V2 en fermeture comme nous le montre la figure 5.17.

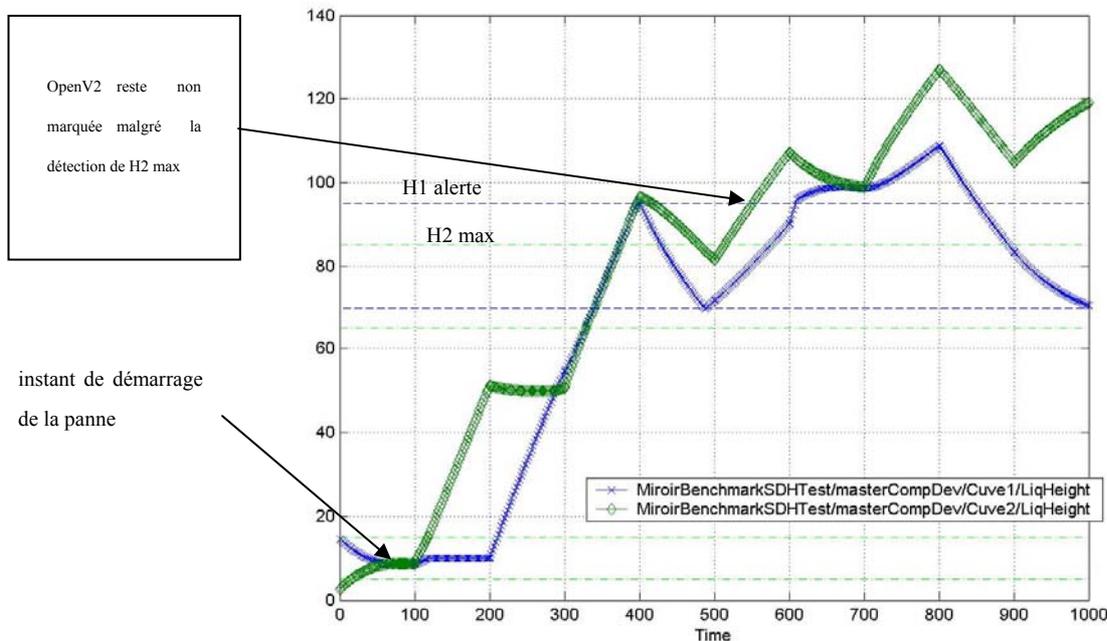


Figure 5.17 Simulation de la vanne V2 bloquée fermée

Le blocage de la vanne V2 en fermeture a engendré un dépassement de seuil dans la cuve2 mais aussi dans la cuve1 puisqu'elles sont connectées entre elles par les vannes V4 et V3.

Nous remarquons que dans les fenêtres temporaires des fautes simulées, on trouve beaucoup de similitudes. Ce sont bien des fautes qui engendrent les mêmes symptômes et qui sont donc particulièrement difficiles à discriminer. C'est pour cela qu'on va établir une première classification des fautes qui ont les mêmes symptômes à partir d'une connaissance préalable des états de défaillance en utilisant l'outil SALSA basé sur la méthode LAMDA (chapitre 4), afin de restreindre l'ensemble des diverses fautes possibles à explorer à un sous-ensemble ne comportant que des fautes très semblables.

3 La classification

3.1 L'outil SALSA

SALSA (*Situation Assesment using Lamda claSsification Algorithm*) est un outil développé sous LabWindows par [Kempowsky, 2004b] pour déterminer l'état fonctionnel (situation) d'un processus en traitant des données en ligne et hors ligne. L'objectif de la phase hors ligne est la conception d'un modèle de comportement du processus à partir du résultat de la classification effectuée sur des données historiques ou expérimentales. La phase en ligne est effectuée pour reconnaître et déterminer l'état fonctionnel actuel d'un processus en temps

réel, en utilisant le modèle de comportement obtenu durant la phase hors ligne. La Figure 5.18 et la Figure 5.19 schématisent ces deux phases d'utilisation :

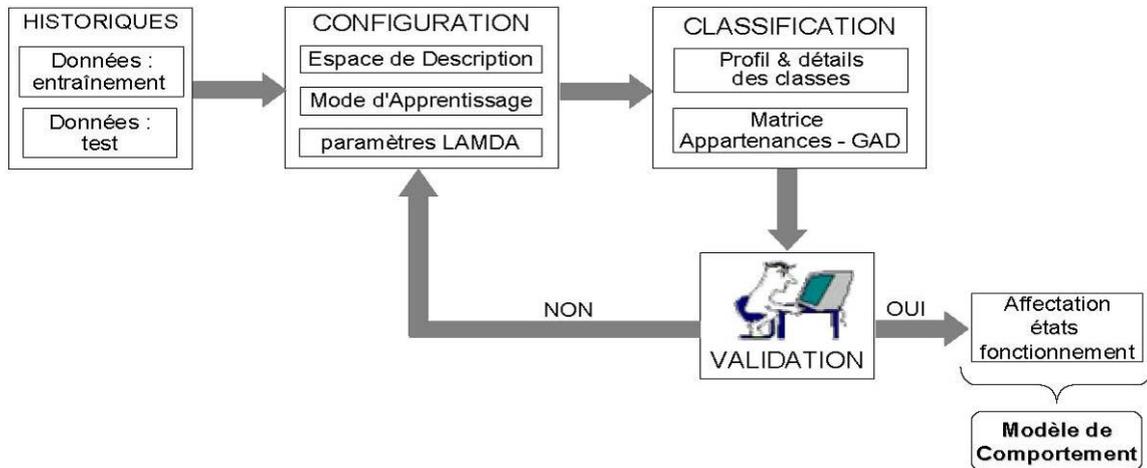


Figure 5.18 Structure de SALSA hors ligne

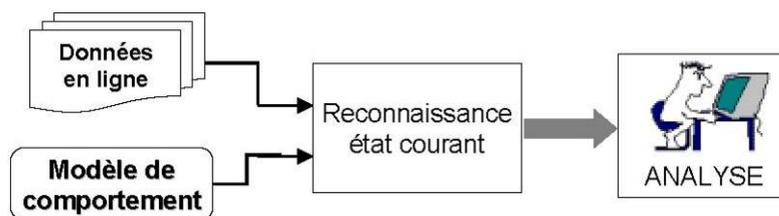


Figure 5.19 Structure de SALSA en ligne

La première phase de la méthodologie de diagnostic comme nous l'avons présentée au chapitre précédemment requiert la possession d'un modèle de comportement issu d'un apprentissage hors-ligne effectué par la méthode de classification SALSA.

3.2 Apprentissage hors ligne

Il s'agit de la construction à partir des données historiques issues du processus, d'un classificateur caractérisé par un ensemble de classes qui permettent d'identifier un ensemble de situations. Ceci correspond à l'obtention d'un modèle de comportement. Il faut posséder des données historiques ou d'observations de différentes situations (normal et anormales). Grâce à des simulateurs, nous pouvons disposer d'un nombre suffisant de données pour chaque état de défaillance et chaque état de fonctionnement normal du processus. En revanche, il est fort probable que tous les défauts n'aient pas été simulés a priori, à cause de la difficulté d'appréhender l'exhaustivité des défauts dans un processus complexe ou dans le cas où deux ou plusieurs défaillances combinées provoquent des effets qu'aucune d'elles, prise isolément, ne pourrait produire (synergie).

Une fois le modèle construit, le suivi des situations attendues, identifié comme étant la phase de reconnaissance, vise à associer toute nouvelle observation à l'une des classes déterminées. Dans notre cas et en se basant sur les simulations de fautes vues précédemment, nous avons construit un modèle de comportement qui est schématisé sur la figure 5.20.

Nous remarquons l'existence de trois classes :

- La première qui définit l'état normal du système.
- La seconde est un état de défaillance associé soit à P1BO ou P2BF ou bien V2BO.

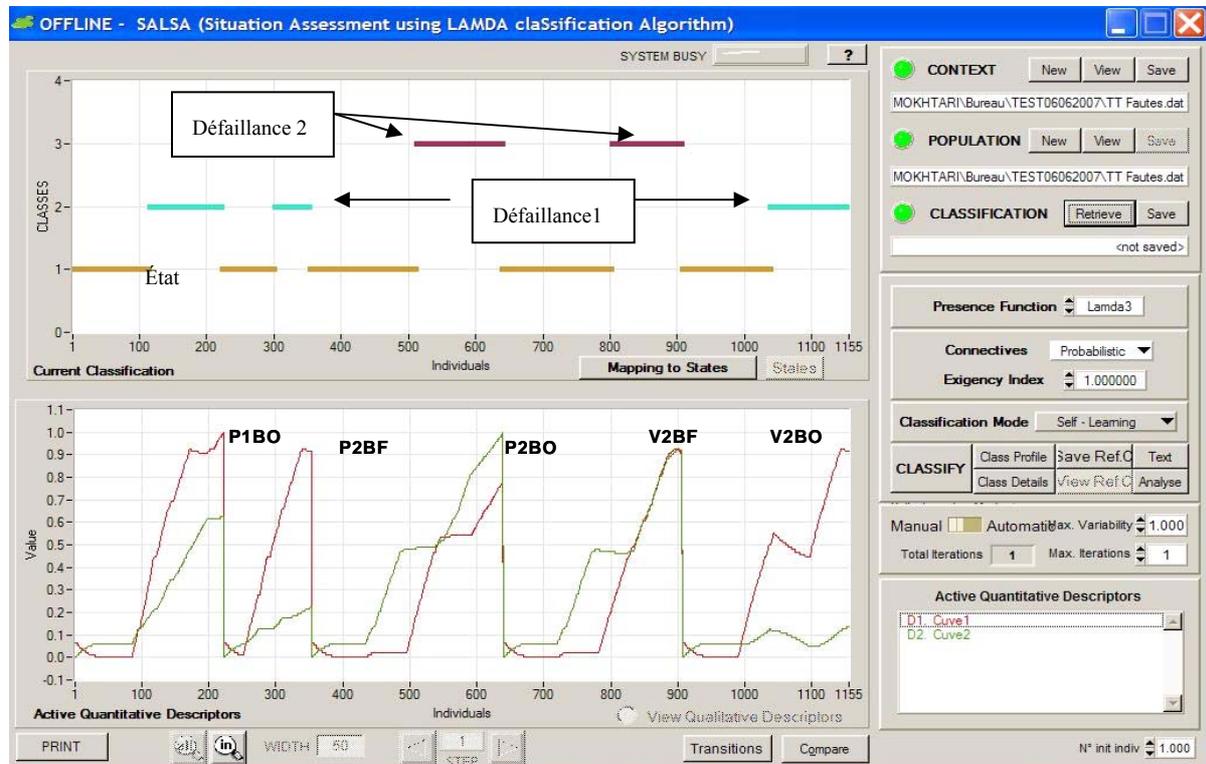


Figure 5.20 Classification des fautes

Nous voyons donc que l'utilisation de la classification dans une étape préalable va nous permettre de réduire les hypothèses de fautes à explorer dans la phase de diagnostic.

Il faut choisir et ajuster certains paramètres de façon à obtenir une partition satisfaisante. Concernant la méthode LAMDA, il y a trois paramètres à régler : la fonction d'adéquation marginale (ou d'appartenance), les connectifs mixtes d'association pour l'agrégation des contributions de chaque descripteur et l'indice d'exigence.

Nous avons utilisé comme fonction d'appartenance la fonction Lamda3 avec la somme probabiliste comme connectif avec un indice d'exigence de 1, ce choix est le fruit de plusieurs tests avec différentes fonctions d'appartenance, différents connectifs et indices d'exigence. Le tableau suivant résume le nombre de classes obtenues avec les différents tests

	EXIGENCY	LAMDA 1	LAMDA 3	GAUSS 1
	0.2	10	2	6
	0.4	10	2	10
MIN-MAX	0.6	12	3	16
	0.8	13	5	19
	1.0	34	6	22
	0.2	12	2	7
	0.4	12	2	12
PROBABILISTIC	0.6	12	3	15
	0.8	12	3	15
	1.0	12	3	16

Figure 5.21 Nombre de classes selon les fonctions d'appartenance

3.3 Diagnostic en ligne

Comme nous l'avons mentionné dans la section précédente, rien ne permet d'assurer l'exhaustivité de la connaissance du comportement du processus car les données historiques ne peuvent pas couvrir la vie entière du processus. De plus, au fur et à mesure de la présentation de nouvelles observations, de nouvelles situations peuvent apparaître dans la structure initiale. Pour cette raison, il est nécessaire que le système de surveillance présente un caractère adaptatif au moment de l'identification des nouvelles situations. Pour cela, deux principes d'apprentissage sont prévus : un apprentissage hors ligne et une reconnaissance en ligne. L'apprentissage hors ligne a été expliqué dans la section précédente, lequel est utilisé dans l'étape de création du modèle de comportement, mais aussi pour caractériser de nouvelles situations. La reconnaissance en ligne permet de détecter en continu un défaut en se basant sur le modèle de comportement, afin de détecter, localiser et identifier une défaillance. La Figure 5.22 montre le diagramme de blocs associé au diagnostic en ligne.

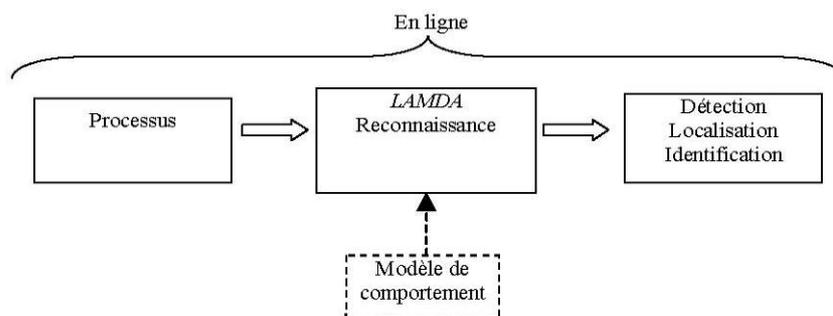


Figure 5.22 Schéma du diagnostic en ligne du procédé

Afin d'expliquer de manière générique l'algorithme de diagnostic en ligne, nous avons divisé la phase de diagnostic en trois parties : l'espace de mesure, l'espace de reconnaissance et l'espace de diagnostic. Dans ce qui suit, nous détaillons chacune de ces étapes. (Figure 23)

a) Espace de mesure : C'est l'ensemble des descripteurs d_1, d_2, \dots, d_p . Ce sont les entrées du système de diagnostic en ligne.

b) Espace de reconnaissance. Cette étape consiste à déterminer, à chaque instant, dans quel état fonctionnel se trouve le procédé lorsqu'une nouvelle observation est présentée. Il est basé sur le modèle de comportement construit hors ligne.

c) L'espace de diagnostic : La méthodologie de diagnostic proposée s'effectue en deux étapes. La première étape consiste à détecter à l'aide d'une classification floue, des fautes qui présentent les mêmes symptômes - à partir d'une connaissance préalable des états de défaillance obtenue par apprentissage - afin de réduire les chemins ou les scénarios à explorer lors de la seconde phase. Cette dernière sert à diagnostiquer la faute parmi celles détectées à l'étape précédente en levant l'ambiguïté. Pour ce faire, un raisonnement avant a été introduit, qui consiste à former pour chaque faute incriminée, un critère d'écart entre les mesures effectuées sur le système et celles émanant du modèle avec la faute simulée, sur une fenêtre temporelle et d'isoler ainsi la faute aboutissant au critère le plus faible.

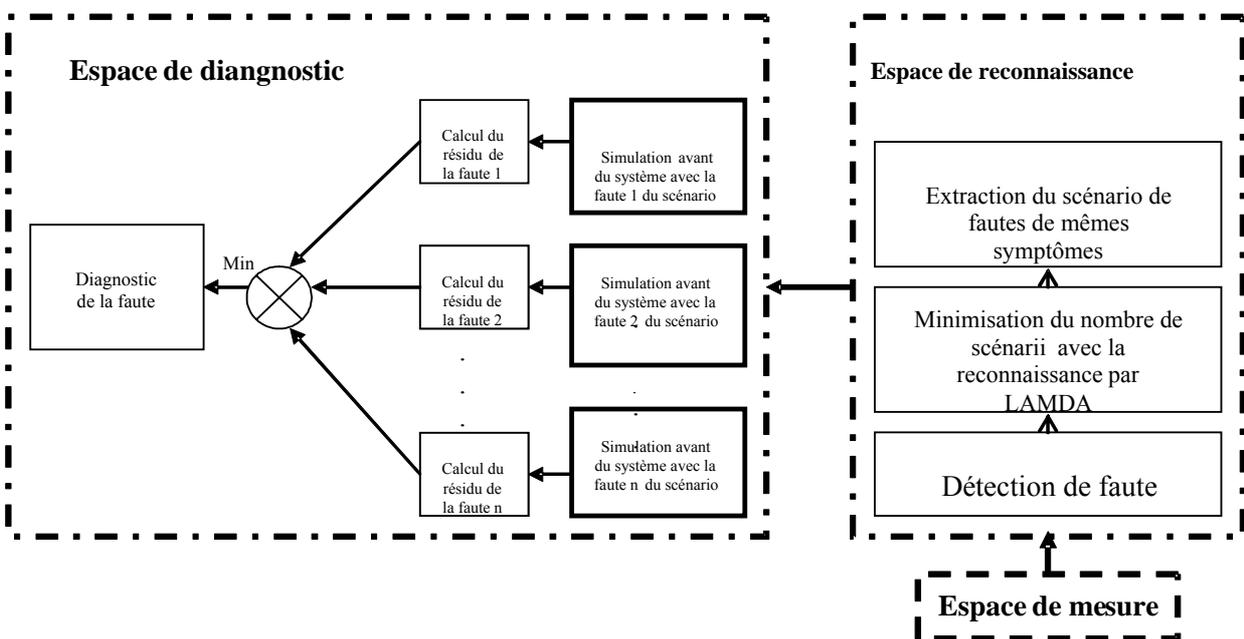


Figure 5.23 Méthodologie de Diagnostic en ligne

Nous avons obtenu trois classes dans notre exemple (figure 5.20)

- La première qui définit l'état normal du système.
- La seconde est un état de défaillance associé soit à P1BO ou P2BF ou bien V2BO.
- La troisième est un état de défaillance associé soit à P2BO ou V2BF

En appliquant le diagnostic en ligne et si nous détectons la classe 2, nous pouvons déduire que l'on est en présence soit de la défaillance P1BO soit P2BF soit V2BO. La détection de la classe 3 permet elle aussi de déduire que la défaillance est soit P2BO soit V2BF. A ce stade, nous pouvons dire que nous avons réduit les chemins des fautes à explorer (de 5 fautes à 3 pour la classe 2 et de 5 fautes à 2 pour la classe 3) et afin de lever cette ambiguïté, nous avons procédé à la simulation de chaque faute retenue dans une fenêtre temporaire en prenant comme conditions initiales l'état du système au moment de la détection de la défaillance. Enfin pour diagnostiquer la faute (c'est-à-dire discriminer parmi les possibilités de fautes restantes), un calcul d'écart entre les mesures des variables effectuées sur le système et celles

émanant du modèle avec la faute simulée est effectué, on choisira comme faute celle qui aboutit au critère le plus faible.

Nous présentons dans ce qui suit un exemple illustrant tout le processus de diagnostic pour les différentes fautes constituant la classe 2 (3 fautes).

3.3.1 Diagnostic des différentes fautes constituant la classe 2

Pour mieux se rapprocher de la réalité, des erreurs sur le modèle du procédé ainsi que sur les mesures de $\pm 5\%$ ont été introduites dans la simulation.

3.3.1.1 Simulation de la faute P1BO avec des erreurs sur le modèle du procédé

Une erreur de 5% dans la simulation a été introduite sur le débit de la pompe P1 permettant ainsi une erreur dans le modèle. La figure suivante montre les résultats de la simulation de la faute.

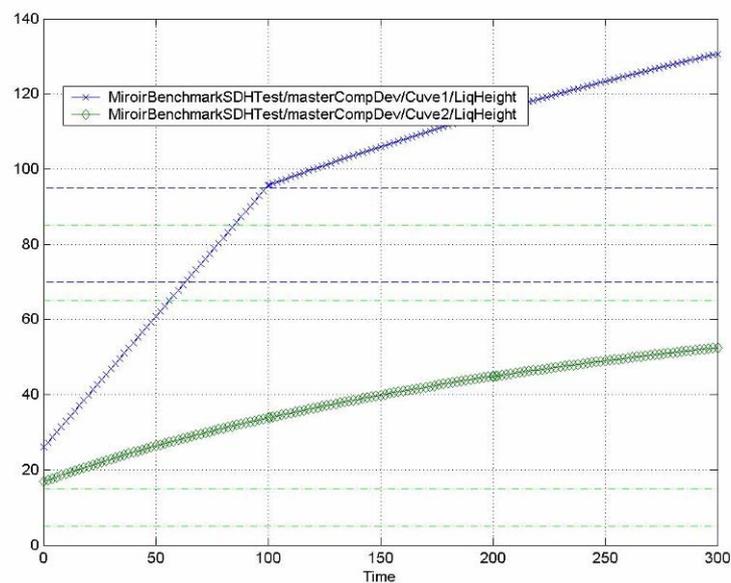


Figure 5.24 Simulation de la faute P1BO avec erreur sur le modèle

3.3.1.1.1 Phase de reconnaissance

En se basant sur le modèle de comportement obtenu précédemment (figure 5.20), nous allons établir une reconnaissance de la faute suivant les trois classes déterminées auparavant. La figure suivante nous montre la reconnaissance des classes lors de la simulation de la faute P1BO avec erreur sur le modèle du procédé.

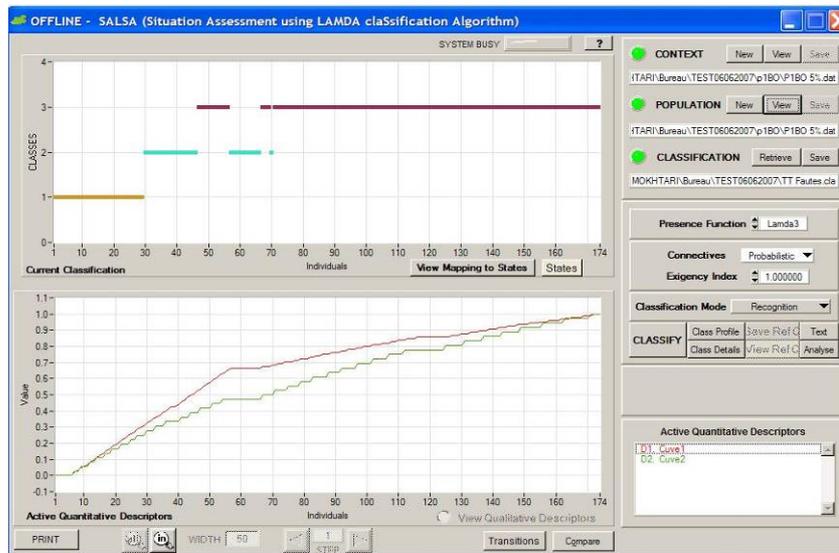


Figure 5.25 Reconnaissance de classe pour la faute P1BO avec erreur sur le modèle du procédé

Nous remarquons que dans la phase de reconnaissance, la première classe de défaillance reconnue est la classe 2 (car en diagnostic en ligne on s'arrête à la première faute détectée) ceci vérifie bien que la classification grâce au modèle de comportement préalablement établi lors de la phase d'apprentissage est apte à détecter un défaut (associé à la classe 2), défaut qui peut être imputable à plusieurs fautes P1BO, P2BF et V2BO.

Un autre résultat important est à relever : c'est le fait que la détection de la faute dans la phase de reconnaissance survient plus tôt que si on avait une détection par seuil. En effet, dans notre exemple la détection s'effectue à l'instant $t=30s$ alors que si on avait une détection par seuil, elle se ferait à l'instant $t=100s$ ceci démontre un autre intérêt de l'utilisation de la classification comme méthode de détection comparée à une méthode de dépassement de seuil.

La première étape de la méthodologie fournit donc à ce stade deux informations :

- la détection d'un fonctionnement anormal
- la possibilité d'une faute de type P1BO, P2BF ou V2BO

Pour pouvoir lever l'ambiguïté présente encore à ce stade, la deuxième phase de la méthodologie est appliquée. Elle consiste dans la simulation avant du fonctionnement du procédé dans les 3 cas de pannes possibles. L'identification de celle responsable du défaut est effectuée par un calcul d'écart entre modèle et mesure.

3.3.1.1.2 Discrimination des fautes

Afin de discriminer entre les fautes, un calcul d'écart est considéré. Nous avons choisi un critère simple qui est la somme de l'erreur quadratique sur les deux niveaux (h_1 et h_2) dans les cuves 1 et 2 (les seules mesures disponibles sur le système), entre le modèle et la mesure effectuée sur le procédé réel, il se traduit par la formule suivante :

$$(h_{1réel} - h_{1simulé})^2 + (h_{2réel} - h_{2simulé})^2$$

Les figures 5.26 donne le résultat de ces critères suivant les 3 cas considérés pour la faute supposée. .

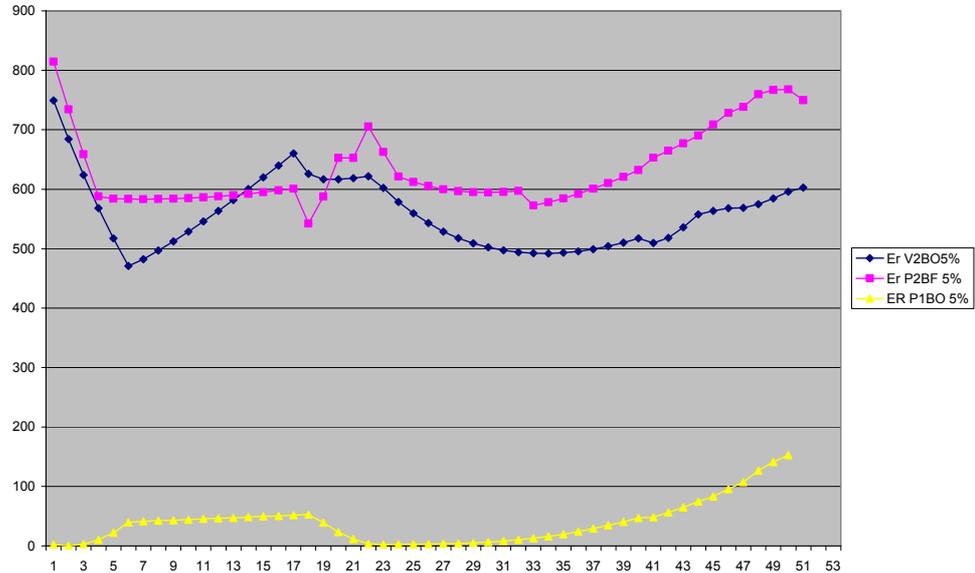


Figure 5.26 Calcul de l'écart avec 5% d'erreur sur le modèle du procédé

Au vu de ces résultats, nous pouvons constater que le plus petit écart est obtenu avec la simulation de la faute, pompe P1 bloquée ouverte (P1BO). Ainsi nous pouvons diagnostiquer que la faute survenue dans le système est bien P1BO.

3.3.1.2 Simulation de la faute P2BF avec des erreurs sur le modèle du procédé

Le même raisonnement est appliqué que précédemment mais cette fois sur la simulation de la faute P2BF dont les résultats sont décrits par la figure 5.27.

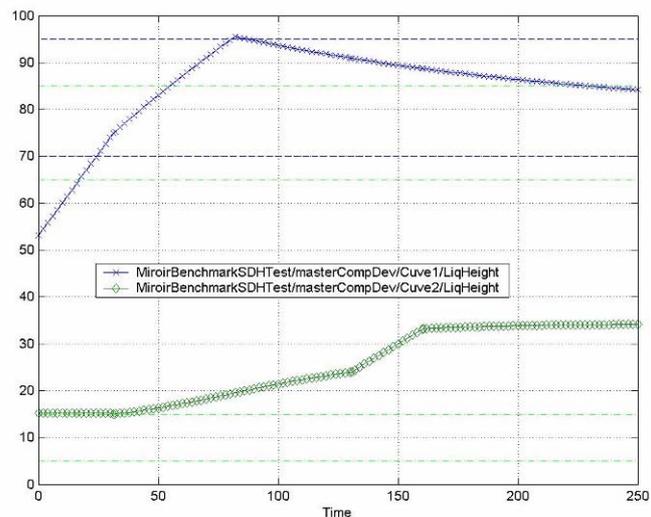


Figure 5.27 Simulation de la faute P2BF avec une erreur sur le modèle du procédé

3.3.1.2.1 Phase de reconnaissance

Nous allons procéder de la même façon que précédemment, c'est-à-dire associer les observations (mesures de h1 et de h2) à des états de comportement correspondant aux classes identifiées précédemment comme nous le montre la figure 5.28.

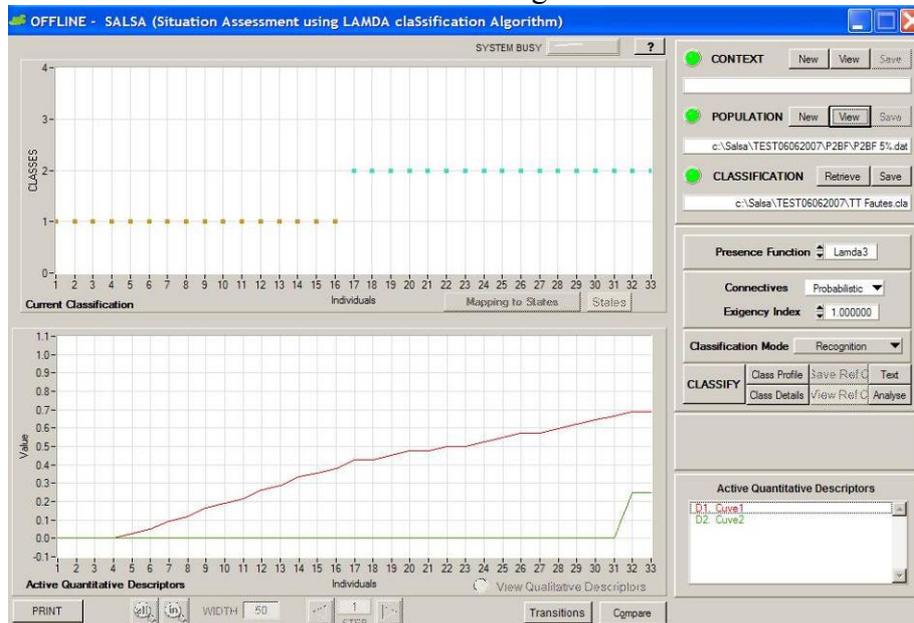


Figure 5.28 Reconnaissance de classe pour la faute P2BF

Comme précédemment, nous remarquons aussi que la détection se fait plus tôt qu'avec une méthode de détection par seuil puisque il se peut que la pompe soit bloquée fermée bien avant la détection de seuil. On retrouve bien que dans cette phase de la détection, la classe associée à la faute est la classe 2 et donc la même que dans le cas précédent..

3.3.1.2.2 Discrimination de la faute P2BF

Comme pour la faute précédente, le même critère d'écart est calculé et on obtient le résultat donné sur la figure 5.29.

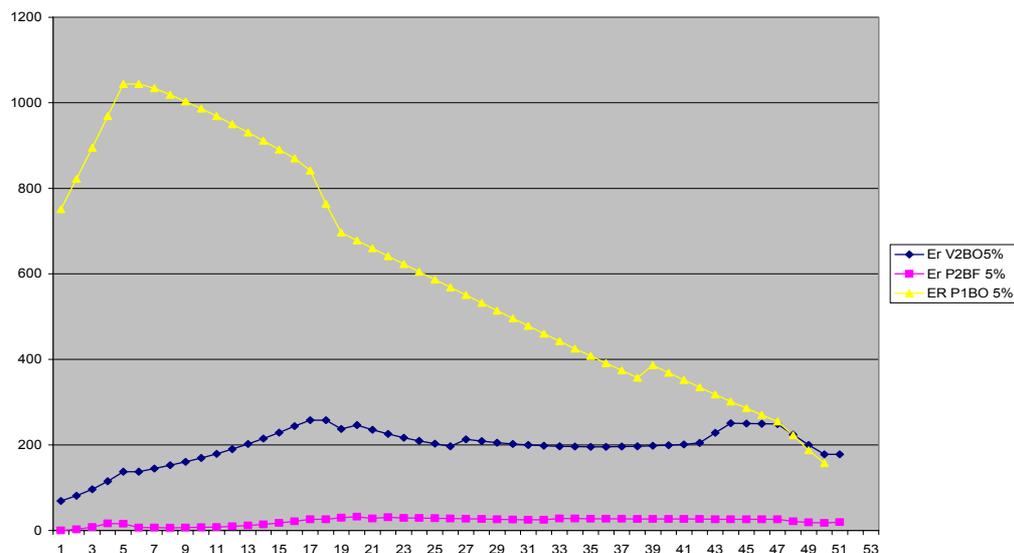


Figure 5.29 Calcul de l'écart avec 5% d'erreur sur le modèle du procédé

Dans ce cas, nous remarquons que le plus petit écart est celui associé à la faute P2BF qui est donc de cette manière, correctement diagnostiquée.

3.3.1.3 Simulation de la faute V2BO avec des erreurs sur le modèle du procédé

Une autre faute peut être assimilée à la classe 2 de notre modèle de comportement (cf. ; figure 5.20) c'est le cas où la vanne 2 est bloquée ouverte, cas que nous détaille la figure 5.30.

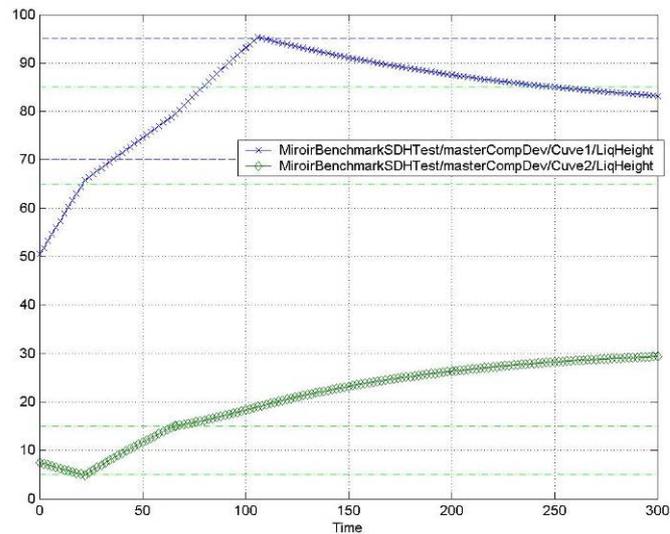


Figure 5.30 Simulation de la faute V2BO avec une erreur de 5% sur le modèle du procédé

3.3.1.3.1 Phase de reconnaissance

Les résultats de la reconnaissance des états fonctionnels à partir de ce fonctionnement observé sont donnés sur la figure 5.31. Elle montre clairement que la faute V2BO est bien détectée mais assimilée aussi à la classe 2.

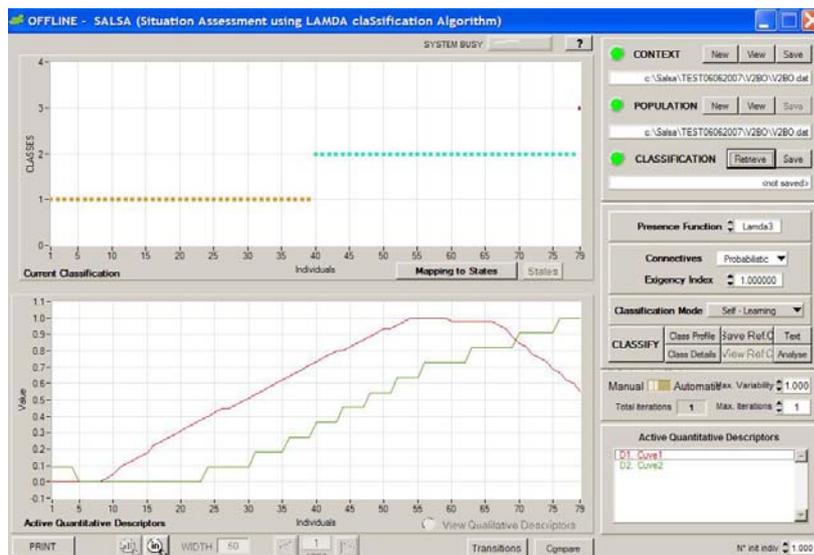


Figure 5.31 Reconnaissance de classe dans le cas de la faute V2B

3.3.1.3.2 Discrimination de la faute V2B0

La discrimination avec les autres fautes s'effectue de la même façon que précédemment grâce au calcul du critère d'écart qui permet de diagnostiquer correctement la faute V2B0 comme ceci est illustré par la figure 5.32.

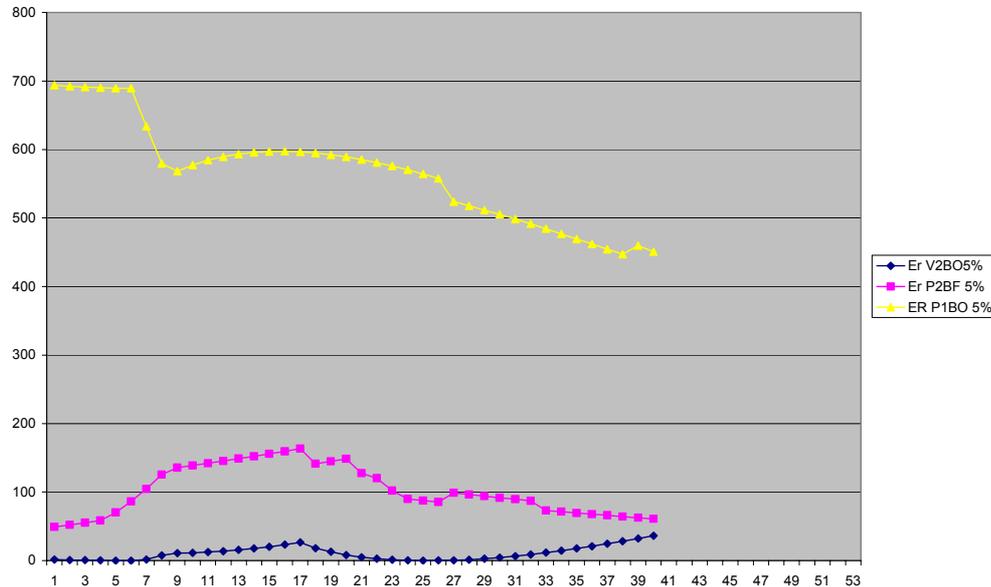


Figure 5.32 Calcul de l'écart avec 5% d'erreur sur le modèle du procédé

On voit bien que la faute incriminée est bien la faute V2B0.

3.3.1.4 Simulation des fautes avec des erreurs sur la mesure

Comme pour les erreurs qui peuvent affecter le modèle, nous avons voulu étudier l'influence des bruits de mesure sur la phase de détection et de discrimination. Dans la simulation suivante, l'erreur introduite n'est plus au niveau du modèle du procédé mais directement sur la mesure des hauteurs h_1 et h_2 (-5%) dans les conditions initiales de la simulation. Nous présentons dans ce qui suit les résultats des simulations des 3 fautes possibles avec ces erreurs de mesure (figures 5.33, 5.34, 5.35).

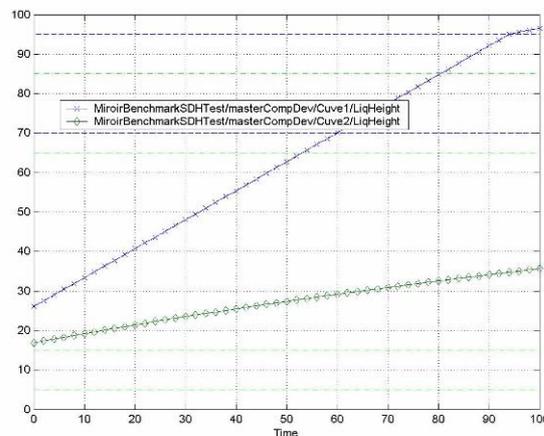


Figure 5.33 Simulation de la faute PIBO avec 5% d'erreur de mesure

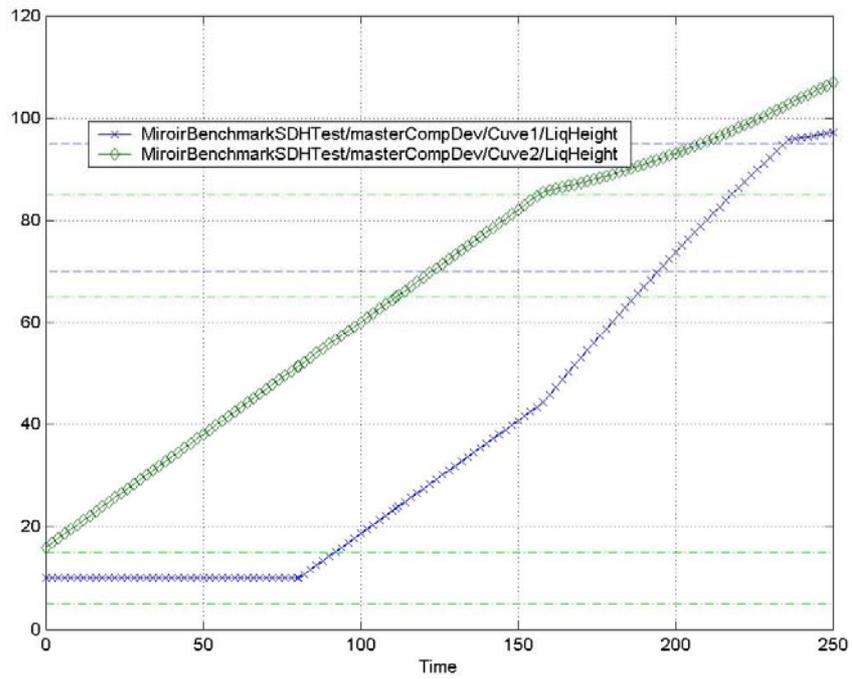


Figure 5.34 Simulation de la faute P2BF avec 5% d'erreur de mesure

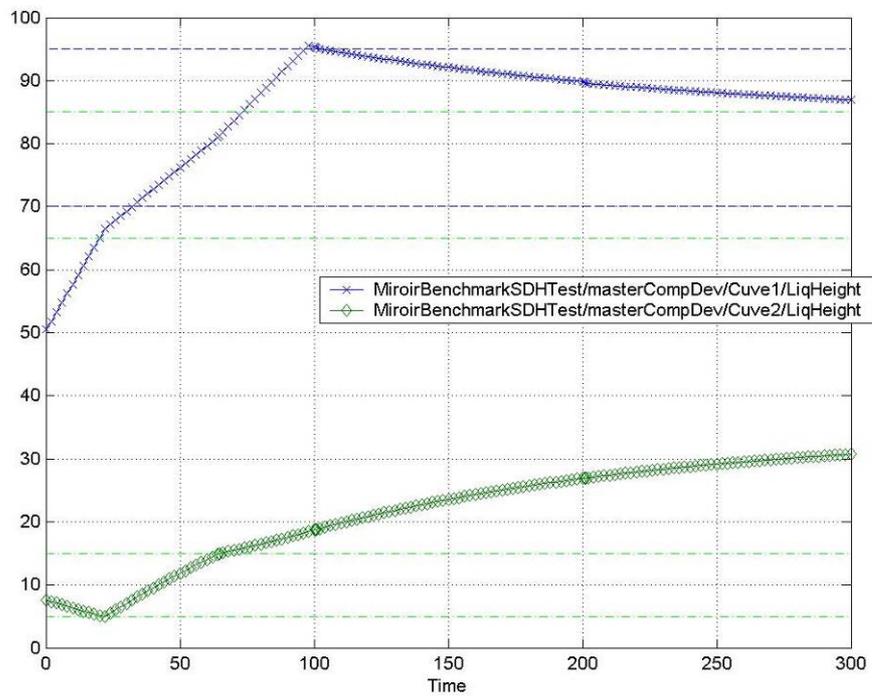


Figure 5.35 Simulation de la faute V2BO avec 5% d'erreur de mesure

3.3.1.4.1 Phase de reconnaissance

Nous allons procéder de la même façon que précédemment. Pour chaque cas de faute simulée, nous avons effectué une phase de reconnaissance qui consiste donc à associer les individus (ensemble constitué des mesures des hauteurs) aux classes du modèle de comportement. Ces phases de reconnaissance ont fourni les figures 5.36, 5.37, 5.38.

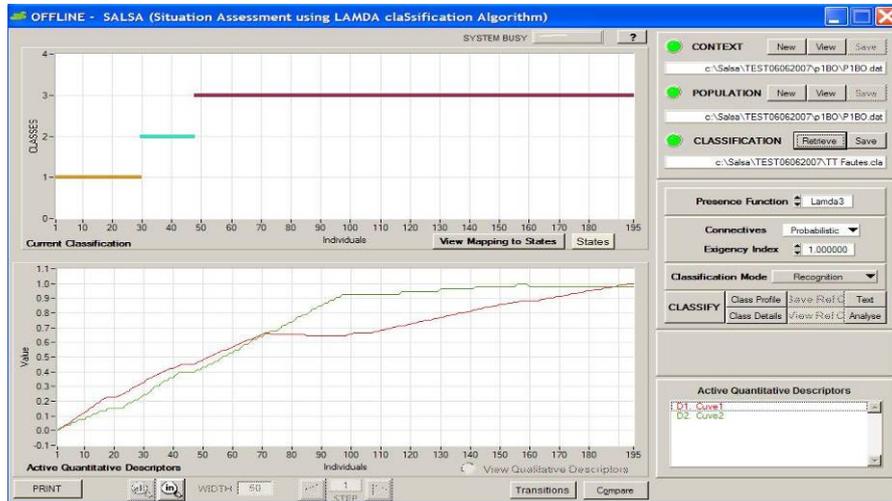


Figure 5.36 Reconnaissance de la faute PIBO avec 5% d'erreur de mesure

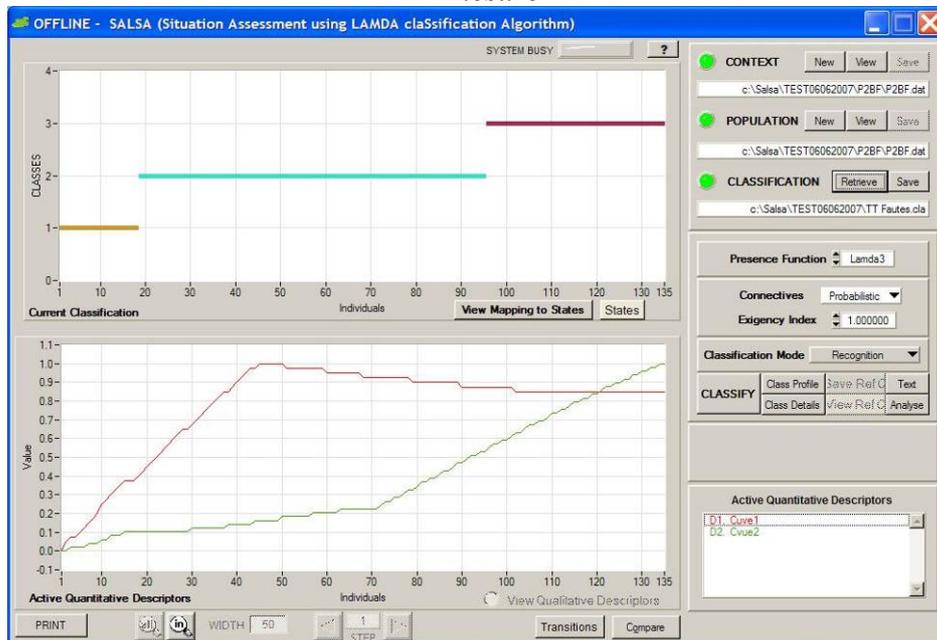


Figure 5.37 Reconnaissance de la faute P2BF avec 5% d'erreur de mesure

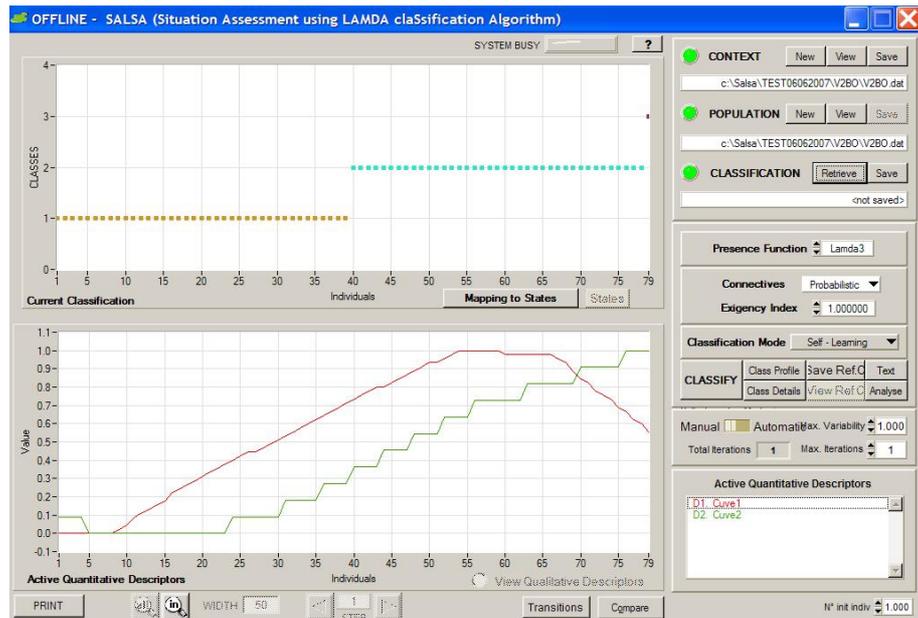


Figure 5.38 Reconnaissance de la faute V2BO avec 5% d'erreur de mesure

Même constat que précédemment : la classe identifiée est la classe 2 et donc la faute incriminée est soit P1BO, P2BF ou bien V2BO ce qui impose le déroulement de la deuxième phase méthodologie consistant à effectuer la simulation avant des fautes en question et le calcul du critère d'écart pour lever l'ambiguïté.

3.3.1.4.2 Calcul du critère d'écart

Nous donnons dans ce qui suit le résultat dans le cas où la faute simulée est P2BF (figure 5.39), mais des résultats similaires sont bien-sûr obtenus pour les autres fautes, comme il a été vu précédemment.

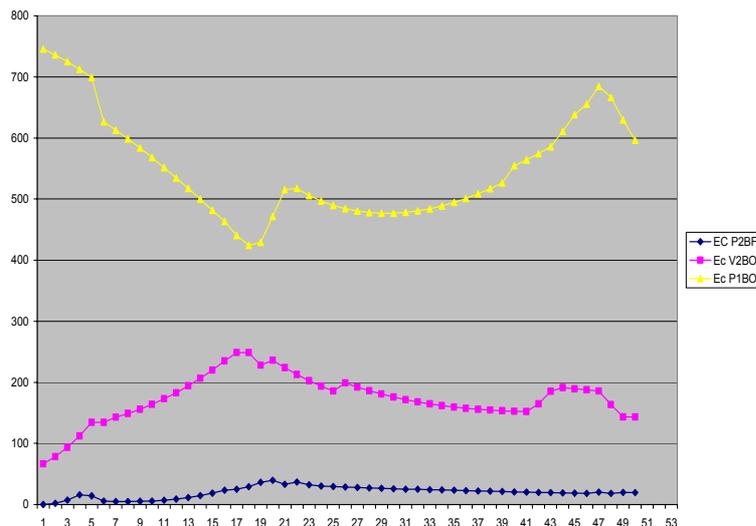
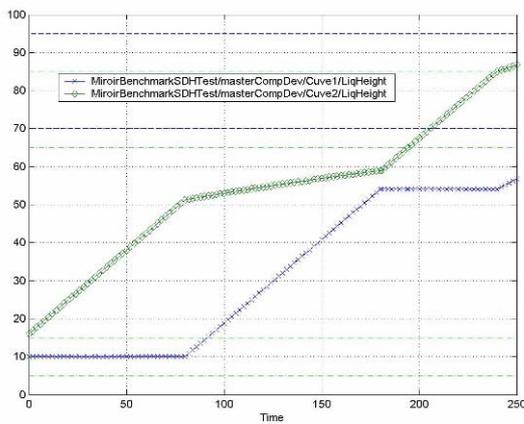


Figure 5.39 Les critères d'écart avec 5% d'erreur sur les mesures

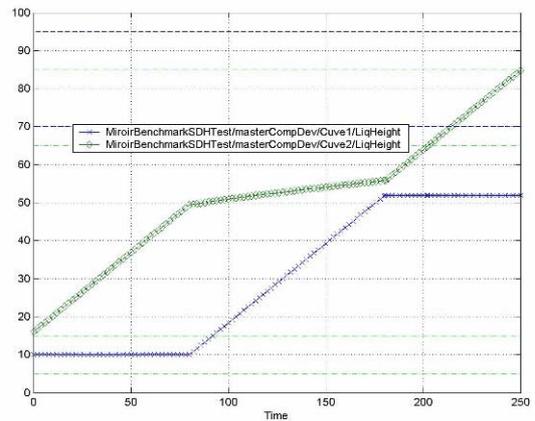
On remarque donc sur cette figure que le plus petit écart est celui qui correspond à la faute P2BF qui est bien celle qui a été simulée. Ainsi nous pouvons diagnostiquer que la faute survenue dans le système est bien P2BF.

3.3.2 Diagnostic des différentes fautes constituant la classe 3

L'autre indétermination concerne la détection de la classe 3. Dans ce cas de figure, les fautes incriminée (cf. figure 20) sont soit P2BO soit V2BF. Comme précédemment, nous avons suivi la même démarche afin d'étudier l'influence d'erreurs sur le modèle du procédé ou bien sur les mesures. Les figures suivantes nous montrent les simulations réalisées dans le cas où la faute à diagnostiquer est P2BO



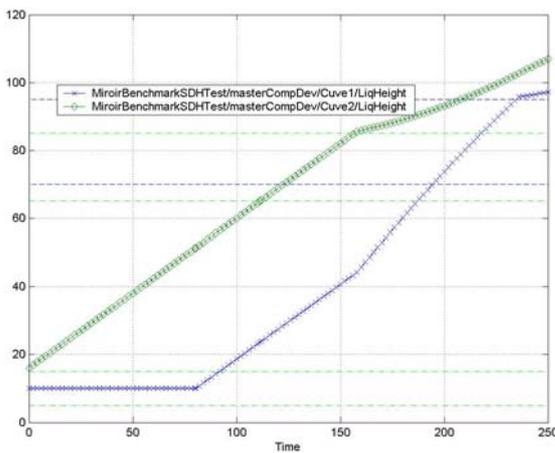
a) 5% erreur de mesure



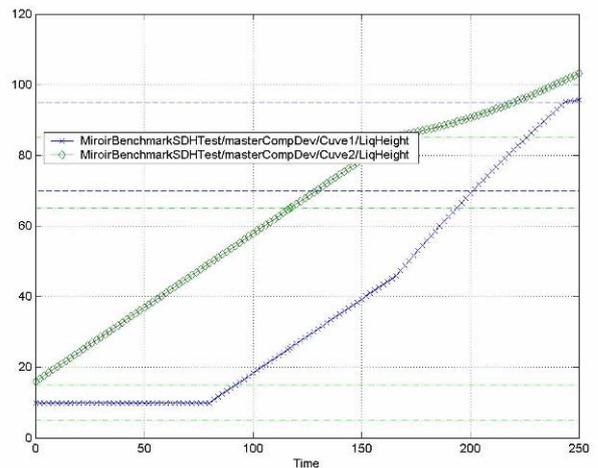
b) 5% erreur du procédé

Figure 5.40

Faute P2BO



a) 5% erreur de mesure

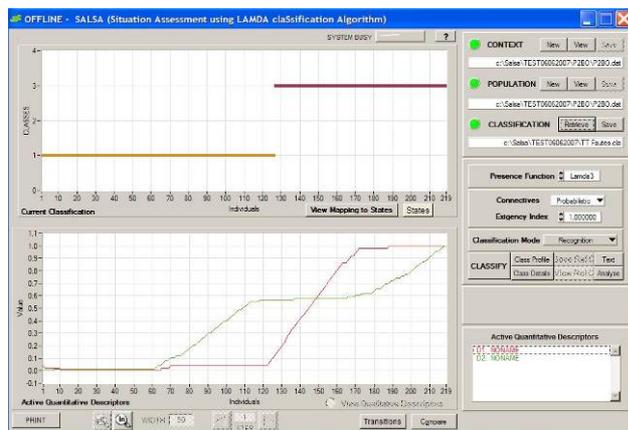


b) 5% erreur du procédé

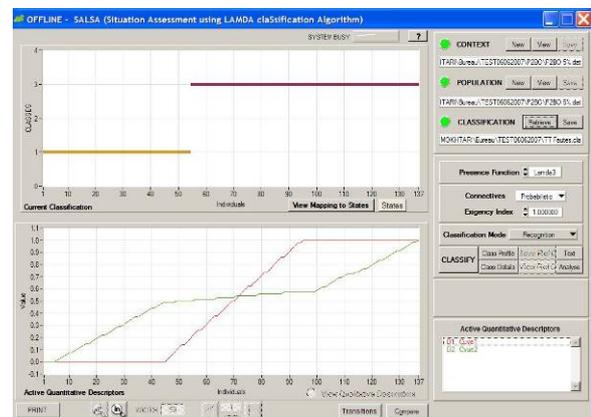
Figure 5.41

Faute V2BF Phase de reconnaissance

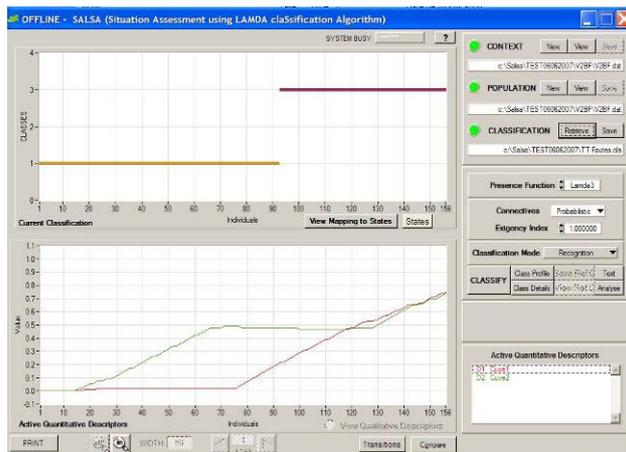
Nous allons vérifier que les fautes détectées sont bien assimilées à la classe 3 du modèle de comportement lors d'une phase de reconnaissance, comme nous le montrent les figures 5.42 et 5.43.



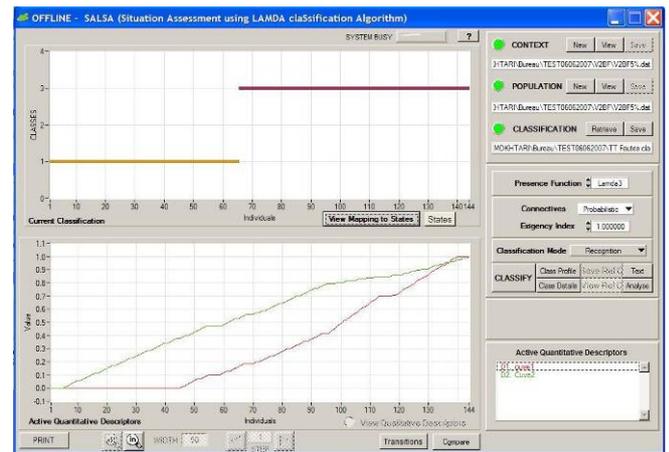
a) 5% erreur de mesure
Figure 5.42



b) 5% erreur du procédé
Reconnaissance de la faute P2BO



a) 5% erreur de mesure



b) 5% erreur du procédé

Figure 5.43 Reconnaissance de la faute V2BF

Nous remarquons dans les deux cas que la classe de défaillance reconnue est la classe 3 qui regroupe les deux fautes, le diagnostic se fera comme précédemment par le calcul du critère d'écart.

3.3.2.1 Calcul du critère d'écart

Le calcul du critère d'écart $(h_{1réel} - h_{1simulé})^2 + (h_{2réel} - h_{2simulé})^2$ dans le cas où la faute à identifier est P2BO est fourni sur la figure 5.44 et 5.45.

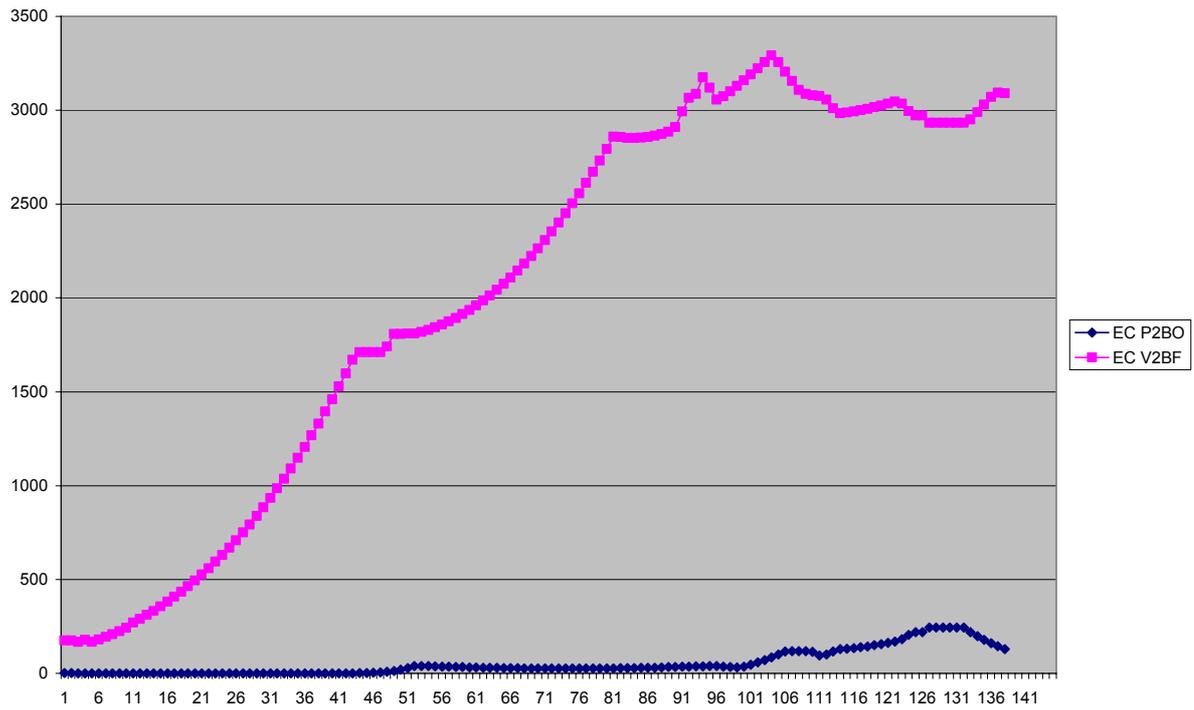


Figure 5.44 Critère d'écart avec 5% erreur de mesure

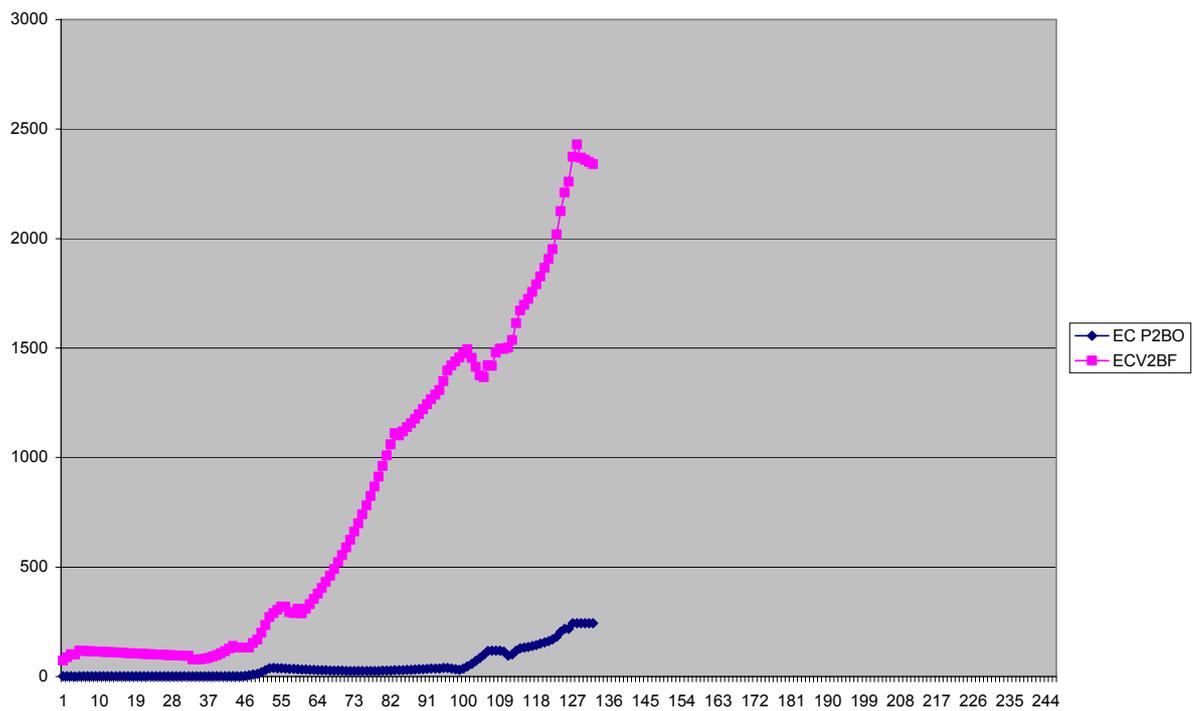


Figure 5.45 Critère d'écart avec 5% d'erreur sur le procédé

Nous pouvons constater dans les deux cas que la faute diagnostiquée est bien P2BO puisque que c'est celle qui présente le plus petit critère d'écart.

4 Conclusion

Ce chapitre a permis au travers d'une application, d'éclairer le lecteur sur la démarche à suivre pour diagnostiquer des fautes en utilisant la modélisation, la simulation et la classification. Il a permis entre autre de montrer que l'utilisation conjointe de ces diverses approches permettait de débloquer des situations que l'on pourrait observer lorsqu'une seule de ces approches était utilisée.

La première phase impliquant la classification a l'avantage certain de restreindre l'espace des possibilités des fautes à « étudier de plus près ». De plus, comme l'ont montré les résultats, cette approche permet de détecter la faute plus tôt que dans le cas d'une approche par détection de seuil. La présence d'erreurs de mesure ou d'erreurs sur le modèle du procédé (par rapport à celui utilisé lors de la phase d'apprentissage pour constituer le modèle de comportement) n'a pas détérioré la capacité de cette approche à identifier la « bonne classe ».

Mais il est clair que si on se contentait de cette phase, on serait incapable de faire un diagnostic de panne. La deuxième phase qui consiste à effectuer la simulation avant permet donc dans tous les cas de discriminer les différents cas constituant l'ensemble des fautes possibles. Si l'étape précédente n'était pas effectuée, il est clair qu'il faudrait examiner tous les cas de fautes y compris celles qui ne conduisent pas aux mêmes symptômes. C'est donc bien l'alliance de ces deux approches confère un des atouts à la méthodologie développée.

Conclusion générale

Le mécanisme de base utilisé pour la détection de fautes est de comparer les évolutions du système observé avec celles d'un modèle qui évolue de façon synchrone, c'est-à-dire en temps réel, avec le système. De même, la tendance actuelle est de fonder les phases de localisation et de diagnostic sur un modèle profond qui décrit la structure et/ou le comportement du système.

Les réseaux de Petri sont l'un des modèles les plus utilisés lorsqu'il s'agit de systèmes à événements discrets. Ils sont en effet, bien adaptés pour décrire la dynamique de tels systèmes (description des passages d'un état à un autre). Par contre, ils ne correspondent pas à un modèle structurel (l'architecture du système dans un état donné). Nous rappelons qu'un système à événement discret est un système (ou plutôt une vision d'un système) pour lequel le temps aussi bien que les composantes du vecteur d'état sont des variables discrètes.

Dans le cas des systèmes continus (les composantes du vecteur d'état tout comme le temps sont des variables continues), il est maintenant fréquent de fonder des approches de diagnostic sur des modèles qualitatifs qui sont des modèles dans lesquels le temps est continu alors que les états qualitatifs sont discrets. De plus, la tendance actuelle est d'affiner le diagnostic des systèmes en faisant intervenir explicitement le temps et leur dynamique sous la forme de changements d'état discret (qualitatif). On se rapproche alors notablement de la problématique des systèmes à événements discrets, d'autant plus que dans ce cadre, on assiste à une démarche en sens inverse. En effet, on cherche à introduire des modèles continus, fonctionnels et structurels pour compléter la vision offerte par les modèles à événements discrets jugés trop réducteurs. Il semble que les deux communautés soient appelées à se rejoindre sur un certain nombre de points, et c'est bien là l'un des buts de notre travail qui a été de développer une solution au diagnostic des systèmes dynamiques hybrides, solution associant les deux modèles continu et discret.

Ces quinze dernières années, les systèmes dynamiques hybrides ont reçu une attention particulière. En effet, l'émergence des procédés « batch », les systèmes de transport, les protocoles de communication n'ont fait qu'alimenter la nécessité de faire évoluer ce domaine

d'études. Les recherches menées se sont articulées autour de plusieurs axes principaux : la modélisation, l'analyse, la commande, la supervision et le diagnostic. Pour notre part, l'objectif principal était d'élaborer un modèle au travers d'un formalisme hybride inséré au sein d'un simulateur dynamique afin de l'analyser et de l'exploiter par simulation à des fins de diagnostic.

Cette thèse a développé une solution au diagnostic des systèmes dynamiques hybrides basé sur des modèles, L'étude bibliographique assez complète, décrite partiellement dans la première partie de ce mémoire, nous a permis de proposer une approche qui intègre différents concepts et méthodes issus de différentes communautés, dans une structure homogène de description événementielle à base de RdP. Nous pensons que l'apport principal de notre travail réside dans cet effort d'intégration qui a abouti à une structure de diagnostic avec un certain nombre de capacités, mises en évidence dans le mémoire.

Nous nous sommes intéressés aux modèles et outils de simulation couramment rencontrés dans le domaine des SDH. Dans ce cadre, il nous a paru important, dans un premier temps, de répertorier les propriétés permettant de les caractériser. Les modèles se définissent en termes de modularité, de causalité, de représentation explicite et rigoureuse des composantes continues et discrètes, de capacités à gérer la structure variable du système algébro-différentiel global. Les outils de simulation sont caractérisés par le caractère modulaire ou global de la simulation, leurs capacités à traiter les équations différentielles ordinaires / partielles / algébriques, la possession d'un algorithme de réduction d'index, la gestion des événements, la réinitialisation explicite et implicite des variables. Afin de mettre en œuvre le formalisme développé, nous avons tenté de dresser un état des lieux des modèles couramment rencontrés dans le domaine des SDH. Ces derniers sont classés selon trois catégories : les extensions des modèles discrets, les extensions des modèles continus et les formalismes mixtes. Pour notre part, le modèle développé repose sur la dernière approche; elle s'appuie sur le réseau de Petri pour la dynamique discrète, sur les équations différentielles algébriques pour la dynamique continue et sur le paradigme RdP/00 pour la structuration des données. Le modèle ainsi développé s'intitule *Réseau de Petri Différentiel à Objets*. C'est ce formalisme qui a été introduit dans la plate-forme de simulation PrODHyS, sur laquelle se sont appuyés nos travaux.

Nous nous sommes basés aussi sur la méthode de classification LAMDA, laquelle repose sur le concept de degré d'appartenance (adéquation) d'un objet aux classes existantes, qui remplace les critères classiques de distance. Cette méthode se différencie d'autres techniques par l'introduction d'une classe non informative qui permet de rendre compte de l'adéquation nulle et de traiter le cas d'éléments non reconnus. Il est clair que la première partie de la méthodologie proposée repose sur la capacité de cette méthode à classer correctement les observations effectuées, puisque sur cette information repose le choix des possibilités de fautes examinées dans l'étape 2. On voit donc que le fait de posséder la possibilité d'avoir comme résultat « le système ne fonctionne pas correctement mais on ne reconnaît pas le type de faute » (ce qui correspond à un état non reconnu), au lieu d'une fausse détection d'un état de faute, est très important pour la suite de la méthode. C'est une particularité de LAMDA qui a aussi motivé son choix. De plus, dans le groupe DISCO des travaux de recherche ont permis de développer des indices de mesure de la qualité des partitions obtenues permettant ainsi « d'optimiser » la partition de l'espace des données en termes de compacité et de séparation des classes (thèse de Claudia ISAZA), [C.Isaza Narvaez

et al., 06]. Un autre développement a permis de développer une validation en temps réel des transitions d'une classe à une autre [C.Isaza Narvaez *et al.*, 07] sur la base des degrés d'appartenance calculés, Ces avancées vont permettre de disposer d'un diagnostiqueur plus puissant et plus fiable et donc améliorer la première phase de la méthodologie proposée.

Notre démarche suivie lors de ces travaux pour diagnostiquer des fautes grâce à la modélisation, la simulation et la classification a permis entre autre de montrer que l'utilisation conjointe de ces diverses approches permettait de débloquent des situations que l'on pourrait observer lorsqu'une seule de ces approches était utilisée seule.

Des travaux sont été effectués en parallèle par Nelly Olivier-Maget au sein du Laboratoire de Génie Chimique de Toulouse, basés aussi sur l'utilisation de la simulation dynamique mais aussi sur l'analyse des signatures obtenues par filtrage de Kalman [N. Olivier-Maget *et al.*, 07]. On peut imaginer que l'on pourra à terme regrouper ces deux approches par, par exemple remplacer les mesures bruitées utilisés dans notre travail par les signaux filtrés, voir par les signatures ceci pourra nous permettre de réduire plus encore les scénarii de fautes à explorer dans la phase deux de la méthodologie proposée. Ces informations peuvent être aussi utilisées pour la construction d'épisodes sur une fenêtre temporelle (thèse de Tatiana) qui sont des descripteurs du fonctionnement dynamique du système de type qualitatif qui peuvent intervenir comme entrées de la classification.

De plus, l'application réalisée lors de ce travail a permis de soulever deux points essentiels :

- le premier repose sur la simplicité des modèles à établir par l'utilisateur pour représenter, finalement, des systèmes assez complexes. Un grand nombre de modèles d'opérations unitaires existent d'ores et déjà dans le simulateur dynamique. Les opérations à réaliser sont en effet peu nombreuses et suivent une procédure générale et systématique.
- La méthode que nous avons développée s'appuie sur le résultat d'une classification avec utilisation de données historiques du fonctionnement ou de la simulation du procédé. Ceci permet de quantifier l'apport d'un descripteur à la discrimination d'une classe de défaillance par rapport à l'état normal.

L'une des perspectives immédiates de notre travail est bien sûr d'approfondir la recherche dans le noyau de simulation afin de résoudre le problème rencontré lors de la résolution d'équations différentielles incluant des pressions, afin d'aboutir au développement de la méthodologie intégrant un raisonnement arrière, ce qui nous permettra en plus du diagnostic, de déterminer l'instant de la faute.

Une autre perspective aussi importante est, comme énoncée précédemment de faire coopérer cette approche avec celle développée lors de la thèse de Nelly Olivier-Marget effectuée au sein du LGC de Toulouse.

Enfin, une perspective à plus long terme serait d'utiliser la connaissance apportée par le modèle et la classification pour déterminer les capteurs les plus pertinents en termes de quantité d'information fournie comme ceci a été abordé dans les travaux effectués dans le groupe Disco [Orantes *et al.*, 06] sur le placement de capteurs.

Bibliographie

- [Abbou *et al.*, 03] Abbou R., Simeu-Abazi Z., Di Mascolo M. (2003). Les réseaux de Petri pour la modélisation et l'analyse des performances d'un atelier de maintenance, Modélisation et simulation (MOSIM 2003), 23-25 avril, Toulouse (France).
- [Alla *et al.*, 92] Alla H., Cavaille J.B., Le Bail J., Bel G. (1992). *Les systèmes de production par lot: une approche discretcontinue utilisant les réseaux de Petri hybrides*, Automation of Mixed Processes (ADPM 92) Janvier, Paris (France).
- [Alla, 94] Alla H. (1994). *Les réseaux de Petri: un outil particulièrement adapté à la modélisation des systèmes hybrides*, Automation of Mixed Processes (ADPM 94), p.17-25, Novembre, Bruxelles (Belgique).
- [Allan, 97] Allan B.A. (1997). *A More Reusable Modeling System*, PhD Thesis, Carnegie Mellon University, Pittsburg (USA).
- [Alur *et al.*, 95] Alur R., Courcoubetis c., Halbwachs N., Henzinger T.A., Ho P.H., Nicollin X, A. Olivero, Sifakis j., Yovine S. (1995). The algorithmic analysis of hybrid systems, *Theoretical Computer Science*, Vol.138, p.3-34.
- [Alur *et Dill*, 94] Alur R., Dill D.L. (1994). A Theory of Timed Automata, *Theoretical Computer Science*, Vol.126, N°2, p.183-225.

[Aguilar-Martin et al.,99]

Aguilar-martin j., Waissman-Vilanova j., Sarrate-Estruch r., et Dahou b. Knowledge based measurement fusion in bio-reactors, IEEE EMTECH, 1999.

[Aguilar-Martin et al.,80]

Aguilar-Martin, J., M. Balssa M., Lopez DE Mantaras R. Estimation réursive d'une partition. Exemples d'apprentissage et auto-apprentissage dans R^n et I^n . Rapport technique 880139, LAAS/CNRS, 1980.

[Aguilar-Martin et al.,82]

Aguilar-Martin, J., R. Lopez DE Mantaras. The process of classification and learning the meaning of linguistic descriptors of concepts".

[Andersson, 99]

Andersson M. (1995). *Omola and Omsim tutorial and user's manual*, Technical Report ISRN LUTFD2/TFRT-7504-SE, Lund Institute ofTechnology, Lund (Sweden).

[Andreu 96].

Andreu D. *Commande et Supervision des Procédés Discontinus: Une Approche Hybride*. Thèse dedoctorat, Novembre, Université Paul Sabatier, Toulouse (France).

[Atine, 05]

Atine J.C.«Méthodes d'apprentissage floue : Application à la segmentation d'images biologiques». Thèse de doctorat Institut National des Sciences Appliquées, Toulouse, France, 2005.

[Audry et al.,94]

Audry N., Ghabri M.-K., Dernongodin L, Prunet F., Ladet P. (1994). *Mode/ling and control of high throughput production lines*, IFAC Workshop on New Trends in Design of Control Systems,p.392-397, 7-10 Septembre, Smolenice (Slovakia).

- [Barton et Pantelides 94]. Barton P.L, Pantelides c.c. The Modelling of Combined Discrete/Continuous Processes, *AIChE Journal*, 40, p966-979.
- [Belaud et Pons, 02] Belaud J.P., Pons M. (2002). *Open Software Architecture For Process Simulation: The Current Status Of Cape-Open Standard*, European Symposium of Computer Aided Process Engineering (ESCAPE12), 26-29 mai, The Hague (The Netherlands).
- [Berthomieu et Menasche, 83] Berthomieu B.; Menasche M. (1983). An Enumerative Approach for Analyzing Time Petri Nets, *IFIP Congress Series*, Vol.9, p.41-46. Elsevier Science Publ. Comp. (North Holland).
- [Blanke et al., 03] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki (2003). *Diagnosis and Fault-Tolerant Control*, Springer.
- [Bousson et al., 98] K. Bousson, J. Steyer, L. Travé-Massuyès, B. Dahhou (1998). From a rule-based to a predictive qualitative model-based approach using automated model generation. Application to the monitoring and diagnosis of biological process, *Engineering Applications of Artificial Intelligence*, 11, pp. 477-493.
- [Branicky, 95] Branicky M.S. (1995). *Studies in hybrid systems: Modeling, Analysis and Control*, PhD thesis, MIT, Massachusetts (USA).
- [Branicky, 97] Branicky M.S., Matison S.E., « Simulation of Hybrid Systems », *Hybrid Systems IV*, Antsaklis P., Kohn W., Nerode A., Sastry S. (dir.), Springer-Verlag, LNCS 1273, p. 3456, 1997.
- [Brenan et al., 89] Brenan K., Campbell S. and Petzold L. (1989). Numerical Solutions of Initial-Value Problems in Differential-Algebraic Equations, Elsevier Science Publishers, North-Holland (Amsterdam).
- [Caradec et Prunet, 98] Caradec M., Prunet F. (1998). *Modélisation des systèmes hybrides flexibles et prise en compte des phénomènes d'accumulation*,

- Automation of Mixed Processes (ADPM 98), p.86-93, March 19-20, Reims (France).
- [Chan et al., 89] Chan, M., Aguilar-Martin J., J. Carrete, N., Celsis, P., et Vergnes, J. M., Classification techniques for feature extraction in low resolution tomographic evolutive images :application to cerebral blood flow estimation. In 12th Conf. GRESTI, 1989.
- [Champagnat,98] Champagnat R. (1998). Supervision des systèmes discontinus: définition d'un modèle hybride et pilotage en temps réel, Thèse de doctorat, LAAS,Toulouse (France).
- [Champagnat et al.,98a] Champagnat R., Andreu D., Pingaud H, Pascal J.c., Esteban P. (1998a). Modélisation des systèmes de production discontinue: proposition d'un modèle pour la supervision, Rapport LAAS N°97524, *Journal Européen des Systèmes Automatisés (APII-JESA)*, Vol.32, N°5-6, p.725-756.
- [Champagnat et al.,98b] Champagnat R., Pingaud H., Alla H., Valentin-Roubinet c., Flaus JM., Valette R. (1998b). A gas storage example as a benchmark for hybrid modelling, *European Journal of Automation*, Vol.32, N°9-10.
- [Chantler et al., 98] M. J. Chantler, G. M. Coghil, Q. Shen, R. R. Leitch (1998). Selecting tools and techniques for model-based diagnosis, *Artificial Intelligence in Engineering*, 12, pp. 81-98.
- [CHEM,06] Projet CHEM disponible sur : <http://www.chem-dss.org/>
- [Cordier et al., 00] M-O. Cordier, P. Dague, M. Dumas, F. Levy, J. Montmain, M. Staroswiecki, L. Través-Massuyès (2000). AI and Automatic Control Approaches of Model-Based Diagnosis: Links and Underlying Hypotheses, *4th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes (SAFEPROCESS'2000)*, Budapest (Hongrie), pp.274-279.
- [Dague, 01] P. Dague (2001). *Diagnostic, intelligence artificielle et reconnaissance des formes*, Traité IC2 Productique, Editions Hermes, Chapitre 1: Théorie logique du diagnostic à base de modèles, pp.17-105.

- [Dash et al., 03] S. Dash, R. Rengaswamy, V. Venkatasubramanian (2003). Fuzzy-logic based trend classification for fault diagnosis of chemical processes, *Computers and Chemical Engineering*, 27, pp.347-362.
- [Daubas et al., 94] Daubas B., Pagès A., Pingaud H. (1994). *Combined simulation of hybrid processes*, IEEE International Conference on Systems, Man and Cybernetics, p.320-325, San Antonio (Texas).
- [David et Alla, 87] David R., Alla H. (1987). *Continuous Petri Nets*, 8th European Workshop on Applications and Theory of Petri Nets, Saragosse (Espagne).
- [David et Alla, 89] David R., Alla H. (1989). *Du Grafcet aux réseaux de Petri*, Hermès Ed., Paris.
- [David et Alla, 94] David R., Alla H. (1994). Petri Nets for Modeling of Dynamic Systems - a Survey, *Automatica*, Vo1.30N°2.
- [David, 91] David R. (1991). *Modeling of Dynamic Systems by Petri Nets*, European Control Conference, 91, p.136147, Grenoble (France).
- [Desroches,87] Desroches, P., Syclare : Système de classification avec apprentissage et reconnaissance des formes. Manuel d'utilisation. Centre d'Etudis Avancés de Blanes, 1987.
- [Denoeux, 01] T. Denoeux (2001). *Diagnostic, intelligence artificielle et reconnaissance des formes*. Systèmes Automatisés (IC2), Editions Hermes, Chapitre 3: Diagnostic par reconnaissance de formes Approches non probabilistes, pp.215-244.
- [Demongodin et al.,93] Demongodin 1.,Audry N., Prunet F. (1993). *Batches Petri Nets*, in IEEE International Conference on Systems, Man and Cybernetics, Vol.1, p.607-617, Le Touquet (France).
- [Demongodin et al.,98] Demongodin 1., Caradec M., Prunet F. (1998). *Fundamental Concepts of Analysis of Batches Petri Nets*, IEEE Internationale Conference on Systems, Man, and Cybernetics, p.845-50, San Diego,(USA).

- [Demongodin.,01] Demongodin I. (2001). Generalised Batches Petri Net: Hybrid Model for High Speed Systems with Variable Delays, *Discrete Event Dynamic Systems: Theory and Applications*, Vol.11, N°1/2, p.137162.
- [Demongodin et Rouibia, 03] Demongodin I., Rouibia S. (2003). *Modélisation par réseaux de Petri lots et analyse de l'état stable par automates hybrides*, Modélisation et simulation (MOSIM 2003),23-25 avril, Toulouse (France).
- [Dubuisson, 01] B. Dubuisson (2001). *Diagnostic, intelligence artificielle et reconnaissance des formes*. Productique (IC2) Editions Hermes, Lavoisier, Paris.
- [Elmqvist et al, 99]. Elmqvist H., Boudaud F., Broenink J. Modelica - A Unified Object-Oriented Language for Physical Systems Modeling, version 1.3.
- [Engell, 97] Engell S. (1997). *Modelling and analysis of hybrid systems*, 2nd IMACS MATHMOD Conference, p.1731, Vienne (Autriche).
- [Fanni & Giua, 98] A. Fanni, A. Giua (1998). Discret Event Representation of Qualitative Models Using Petri Nets, *IEEE Transactions on Systems, Man and Cybernetics – Part B, Cybernetics*, vol. 28.
- [Florin et Natkin,85] Florin G., Natkin S. (1985). Les Réseaux de Petri Stochastiques, *Technique et Science Informatiques*, VolA, N°1, p143-160.
- [Florin et al, 91] Florin G., Fraize C., Natkin S. (1991) Stochastic Petri Nets: Properties, Applications and Tools, *Microelectronics and Reliability*, Vol.31, N°4, p.669-697.
- [Foche, 74] Foche H., Trannoy B., Faucher J., «Complete simulation of a static converter by digital and analogue methods », *1rst IFAC Symposium on Control of Power Electronics and Electrical Drives, Dusseldorf*, vol. 1, p. 363-373, 1974.

- [Frank & Köppen 97] M. P. Frank, B. Köppen-Seliger (1997). New Developments Using AI in Fault Diagnosis, *Engng Applic. Artif. Intell.*, vol. 10, n.1, pp. 3-14.
- [Galindo,02] Galindo M., Aguilar-Martin J., « Interpretación secuencial de encuestas con aprendizaje LAMDA. Aplicación al diagnóstico en psicopatología” IBERAMIA’02-Iberoamerican Conference on Artificial Intelligence, Seville, Espagne, 2002.
- [Gani et Braunschweig, 02] Gani R., Braunschweig B.L. (2002). Software architectures and tools for computer aided process Engineering, Elsevier, ISBN :0-444-50827-9.
- [Gear 71]. Gear C.W. The Simultaneous Numerical Solution of Differential-Algebraic Equations, *IEEE Transaction on Corcuit Theory*, CT 18 (1), Ed. Academie Press.
- [Gentil, 03] S. Gentil (2003). *Commande et supervision – les procédés agroalimentaires 2*, Traité IC2 Systèmes Automatisés, Editions Hermes, Chapitre 4: Supervision et diagnostic, pp. 101-135.
- [Gehin et al., 00] A. Gehin, M. Assas, M. Staroswiecki (2000). Structural Analysis of System Reconfigurability, *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS2000)* Budapest, Hungary, 292-297.
- [Gentil et al., 04] S. Gentil, J. Montmain, C. Combastel (2004). Combining FDI and AI Approches Within Causal-Model-Based Diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics – part B*, vol. 34, n.5, pp. 2207-2221.
- [Guéguen ;Lefebvre,01] Guéguen, H. and Lefebvre, M. A (2001). A comparison of mixed specification formalisms, *JournalEuropéen des Systèmes Automatisés (APII JESA)*, Vol.35, N°4, p.381-394.

- [Ibrahim, 93] Ibrahim, F., Vers un outil de simulation d'un processus dynamique hybride et de sa commande, Thèse de doctorat de l'INPL, Nancy, 1993.
- [Iung, 76] Iun C., Louis J.P., «A digital simulation for some processes fed by AC.DC converters », *Simulation of Systems*, Deckker L. (dir.), North Holland, p. 805-813, 1976.
- [C.Isaza Narvaez et al., 06] C.Isaza narvaez, J.Aguilar Martin, M.V.Le lann, J.Aguilar, A.Rios Bolivar An optimization method for the data space partition obtained by classification techniques for the monitoring of dynamic processes *9èmes Congrès International de l'Association Catalane d'Intelligence Artificielle (CCIA'06)*, Perpignan (France), 26-27 Octobre 2006 Artificial Intelligence Research and Development, IOS Press, 2006, ISBN 1-58603-663-7, pp.80-87, (2006)
- [C.Isaza Narvaez et al., 07] C.I saza Narvaez, E.Diez lledo, H.Hernandez de leon, J.Aguilar Martin, M.V.Le lann New decision method for functional states validation in a drinking water plant monitoring *10th Computer Applications in Biothechnology, DYCOPS/CAB*, Cancun, (Mexique) 6–8 juin 2007, (2007)
- [Jarvis et Pantelides, 92] Jarvis, R. B. and Pantelides, C. C. (1992). DASOLV a diferencial algebraic equation solver. Technical report, Centre for Process Systems Engineering, Imperial College, London.
- [Jourda 96]. Jourda L. *Composants Logiciels Orientés Objets pour la Modélisation et la Simulation des Procédés Chimiques*, Thèse de doctorat, INP, Toulouse(France).
- [Jourda et al ,96] Jourda L., Joulia X., Koehret B. (1996). Introducing ATOM, the Applied Thermodynamic Object Model, *Computers and Chemical Engineering*, Suppl. A, 20, p.S157-S164.

- [Kesten et Pnueli,92] Kesten Y., Pnueli A (1992). Timed and hybrid statecharts and their textual representation, *Lecture Notes in Computer Science (LNCS)*, Vol.N°571.
- [Hélias et al., 04] A. Hélias, F. Guerrin, J-P Steyer (2004). Ababstracting Continuous System Behaviours into Timed Automata: Application to Diagnosis of an Anaerobic Digestion Process, *International Workshop on Principles and Diagnosis (DX 2004)*, Carcassone, France.
- [Heim et al., 03] B. Heim, S. Gentil, B. Celse, S. Cauvin, L. Través-Massuyès (2003). FCC Diagnosis Using Several Causal and Knowledge based Models, *5th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS-2003)*, Washington, USA.
- [Hernandez, 06] Hernandez H., « Supervision et diagnostic des procédés de production d'eau potable » , Thèse de Doctorat, Institut National des Sciences Appliquées, Toulouse, France, octobre 2006.
- [Isermann, 84] Isermann R. Process fault diagnosis based on modelling and estimation methods- A survey. *Automatica*, vol 20, pp. 387-40.1984.
- [Isermann, 97] R. Isermann (1997). Supervision, Fault-Detection and Fault-Diagnosis Methods – An Introduction, *Control Eng. Practice*, vol.5, pp. 639-652.
- [Iwasaki & Simon, 86] Y. Iwasaki, H.A. Simon (1986). Causality in device behavior, *Artificial Intelligence*, vol. 29, n.1, pp. 3-33.
- [Kempowsky,04a] Kempowsky T. « Surveillance de procédés à base de méthodes de classification : Conception d'un outil d'aide pour la détection et le diagnostic des défaillances ». Thèse de doctorat, Institut National des Sciences Appliquées, Toulouse, France, décembre 2004.
- [Kempowsky, 04b] Kempowsky T., SALSA user's manual. Rapport LAAS-CNRS no. 04160, 2004.
- [Kleer, 86] J. De Kleer (1986). Theories of causal ordering, *Artificial Intelligence*, vol. 29, n.1, pp. 33-62.

- [Kleer et al., 92] J. De Kleer, A. Mackworth, R. Reiter (1992). Characterizing diagnoses and systems, *Artificial Intelligence*, vol. 56, pp. 197-222.
- [Kleer, 03] J. Kleer (2003). Fundamentals of Model-based Diagnosis, *International Workshop on Principles of Diagnosis (DX'03)* June 11-14, Washington, DC, USA.
- [Le Bail et al.,91] Le Bail J., Alla H., David R. (1991). *Hybrid Petri Nets*, European Control Conference, p.1472-1477,Grenoble (France).
- [Le Bail et al.,93] Le Bail J., Alla H, David R (1993). Asymptotic Continuous Petri Nets, *Journal of Discrete Event Dynamic Systems: Theory and Applications*, Vol.2, p.235-263.
- [LE LANN, 99] Le Lann J.M .Habilitation à diriger des recherche , INP, Toulouse (france)
- [Lunze, 98] J. Lunze (1998). Qualitative modelling of dynamical systems Motivation, methods and prospective applications, *Mathematics and Computers in Simulation*, 46, pp. 465-483.
- [Lunze, 00] J. Lunze (2000). Diagnosis of Quantised Systems, *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS-2000)* Budapest, Hungary, vol. 1, pp. 28-39.
- [MacGregor, 95] MacGregor J. F., Kourti T., “Statistical process control & multivariate processes.” *Control Engineering Practice*, Vol. 3, N°3, pp. 403-414, 1995.
- [Mardon,89] Mardon D. (1989). *Modélisation Numérique et Simulation Orientée Objet d'une Usine de Retraitement Nucléaire*, Thèse de doctorat, Université de Bordeaux I, France.
- [Marsan,89] Marsan M.A. (1989). Stochastic Petri Nets: An Elementary Introduction, G. Rozenberg, ed., *Advances in Petri Nets, Lecture Notes in Computer Science*, Vol.424, p.1-29, Springer Verlag.
- [Medjoudj 06] Medjoudj.M *Contribution à l'analyse des systèmes pilotés par calculateurs : Extraction de scénarios redoutés et vérification de contraintes temporelles* Thèse de doctorat UPS Toulouse (France)

- [Meléndez & Colomer, 01] J. Meléndez, J. Colomer (2001). Episodes representation for supervision. Application to diagnosis of a level control system, *Workshop on Principles of Diagnosis (DX'01)*, Sansicario, Italia.
- [Merlin et Segall,76] Merlin P. M., Segall A. (1976). Recoverability of Communication Protocols-Implications of a Theoretical Study, *IEEE Transactions on Communications*.
- [Mokhtari et al.,06 a] A.MOKHTARI, M.V.LE LANN, G.HETREUX, J.M.LE LANN
A fault diagnosis approach for hybrid systems,32 nd Annual Conference of the IEEE Industriel Electronics Society(IECON006) Paris 7-10 NOV 2006.
- [Mokhtari et al.,06 b] A.MOKHTARI, M.V.LE LANN, G.HETREUX, J.M.LE LANN A fault diagnosis approach using ODPN simulation for hydraulic systems EMSS European Modeling and Simulation Symposium 4-6 octobre 2006 Barcelone (Espagne)
- [Mokhtari et al.,06 c] A.MOKHTARI, M.V.LE LANN, G.HETREUX, J.M.LE LANN Diagnostic à base de modèle des systèmes dynamiques hybrides, SIMO 06 Systèmes d'Information, Modélisation, Optimisation et Commande en génie des procédés 11-12 octobre 2006 Toulouse.
- [Mokhtari et al.,06 d] A.MOKHTARI, M.V.LE LANN, G.HETREUX, J.M.LE LANN Approche de diagnostic par la simulation des réseaux de Petri à objets 'International Conference on Control, Modelling and Diagnostics (ICCMD'06) 22-24 Mai 2006 Annaba (Algerie)
- [Moler, 98] Moler C. Are we there yet? Zero crossing and event handling for differential equations,*Simulink 2 special edition, Matlab, News andNotes, 16-17*.
- [Mosterman, 01] J. Mosterman (2001). Diagnosis of Physical Systems With Hybrid Models Using Parameterised Causality, *Proceedings of Hybrid Systems: Computation and Control, 4th International Workshop (HSCC01)*, Rome, Italia, pp.447-458.
- [Mosterman & Biswas, 99] J. Mosterman, G. Biswas (1999). Diagnosis of Continuous Valued Systems in Transient Operation Regions, *IEEE Transactions on Systems, Man, And Cybernetics*, vol. 29, n.6, pp. 554-565

- [Moyse, 00] Moyse A (2000). *Modélisation Orientée Objet des Opérations Unitaires*, Thèse de doctorat INP, Toulouse, France.
- [Nakhle, 83] Nakhle.M, Roux P., Exposé de présentation du prologiciel NEPTUNIX, Document CISI, RTI/AST/SIMU/83-004, 1983.
- [Nomikos, 94] Nomikos P., MacGregor J., “MoniiDring batch processes using multiway principal component analysis.” *American Institute of Chemical Engineers Journal*, Vol. 40, N° 8, pp.1361-1375, 1994.
- [Olivier-Maget et al., 07] Fault detection using a hybrid dynamic simulator: Application to a hydraulic system *International Modeling and Simulation Multiconference*, , *IMSM07/ CMS07*, 8-10 Février 2007, Buenos Aires, (Argentine), 8p., (2007)
- [Orantes, 05] Orantes A. «Méthodologie pour le placement des capteurs à base de méthodes de classification en vue de son diagnostic ». Thèse de doctorat, Institut National des Sciences Appliquées, Toulouse, France, octobre 2005.
- [Orantes et al., 06] A. Orantes, T. Kempowsky and M.-V. Le lann « Classification as an aid tool for the selection of sensors used for fault detection and isolation » *Transactions of the Institute of Measurement and Control*, 12 2006; vol. 28: pp. 457 - 479.
- [Orantes et al., 07] A. Orantes, T. Kempowsky, M.-V. Le lann and J. Aguilar-Martin « A new support methodology for the placement of sensors used for fault detection and diagnosis, Chemical Engineering and Processing, In Press, Corrected Proof, Available online 19 January 2007
- [Pagès et Pingaud, 95] Pagès A, Pingaud H. (1995). *An hybridprocess model based on Petri Nets applied to short term scheduling of batch-semi continuous plants*, Proc, of the Workshop on Analysis and Design of Event-Driven Operations in Process Systems (ADEDOPS), Imperial College, London, UK.
- [Palomino 00] Palomino B. Conception d'un simulateur hybride pour l'aide à la décision en ordonnancement d'ateliers batch de l'agroalimentaire, Thèse de doctorat, INP, Toulouse (France).

- [Pascal 99]. Pascal J.c.Habilitation à Diriger les Recherches, Université Paul Sabatier, Toulouse (France).
- [Perret .03] Perret J. Intégration des Réseaux de Petri Différentiels à Objets dans une plateforme de simulation dynamique hybride : application aux procédés industriels, Thèse de Doctorat, INP, Toulouse (France).(2003).
- [Pingaud . 98] Pingaud H. Habilitation à Diriger les Recherches, INP, Toulouse (France).
- [Piera, et al.,91] Piera N., Aguilar-Martin J. Controlling selectivity in non-standard pattern recognition algorithms ». IEEE transactions on systems, man and cybernetics, Vol 21, No. 1, 1991.
- [Pons & Través-Massuyès, 97] R. Pons, L. Través-Massuyès (1997). Causal Ordering for multiple mode system, *11th International Workshop on Qualitative Reasoning*, Cortona (Italie), 11p.
- [Ramchandani, 74] Ramchandani C. (1974). *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, PhD thesis, Massachusetts Institute of Technology, Cambridge (Etats-Unis).
- [Recht, 66] Recht J. L., Failure Mode and Effect. National Safety Council, 1966.
- [Reiter, 87] R. Reiter (1987). A theory of diagnosis from first principles, *Artificial Intelligence*, vol. 32, pp. 57-95.
- [Sargousse, 99]. Sargousse A, *Noyau numérique Orienté-Objet dédié à la Simulation des systèmes Dynamiques Hybrides*, Thèse de doctorat, INP, Toulouse, France.
- [Scenna, 00] N. J. Scenna (2000). Some aspects of fault diagnosis in batch processes, *Reliability Engineering and Systems Safety* 70, pp. 95-110.

- [Sifakis, 77] Sifakis, J. (1977). Use of Petri Nets for Performance Evaluation, *Measuring, Modelling and Evaluating Computer Systems*, p 75-93. Amsterdam (North Holland).
- [Toscano, 05] Toscano R., *Commande et diagnostic des systèmes dynamiques: Modélisation, analyse, commande par PID et par retour d'état, diagnostic*. Ellipses édition Marketing, S. A., 2005.
- [Travé-Massuyès et al., 97] L. Travé-Massuyès, P. Dague, F. Guerrin (1997). *Le Raisonnement qualitatif pour les sciences de l'ingénieur*. Editions Hermes, Paris.
- [C. Valentin-Roubinet 98] Valentin-Roubinet C. (1998), Modelling of hybrid systems: DAE supervised by Petri nets, the example of a gaz storage, *Automation of Mixed Processes (ADPM 98)*, p. 142-149, Match 19-20, Reims (France).
- [Valentin-Roubinet, 99] Valentin-Roubinet C. (1999), *Hybrid Systems modelling : Mixed Petri Nets*, 3rd IMACS/IEEE Conference CSCC 99, 4-8 Juillet, p.223-228, Athènes (Grèce).
- [Valette, 02] Valette R. (2002). Notes de cours: Les réseaux de Petri, INSA, Toulouse (France).
- [Venkatasubramanian et al., 03a] V. Venkatasubramanian, R. Rengaswamy, N. Kavuri, K. Yin (2003). A review of process fault detection and diagnosis Part I: Quantitative model-based methods, *Computers and Chemical Engineering*, 27, pp. 293-311.
- [Venkatasubramanian et al., 03b] Venkatasubramanian V., Rengaswamy R., Kavuri. (2003). A review of process fault detection and diagnosis Part II: Qualitative Models and Search Strategies, *Computers and Chemical Engineering*, 27, pp. 313-326.

[Venkatasubramanian et al., 03c]

Venkatasubramanian V., Rengaswamy R., Kavuri, N., Yin K. (2003). A review of process fault detection and diagnosis Part III: Process history based methods, *Computers and Chemical Engineering*, 27, pp. 327-346.

[Villemeur, 88]

Villemeur A., Sûreté de fonctionnement des systèmes industriels. Fiabilité Facteurs humains Informatisation. Eyrolles, 1988 .

[Waissman-Vilanova, 2000]

Waissman-Vilanova J., Construction d'un modèle comportemental pour la supervision de procédés : application à une station de traitement des eaux. Thèse de doctorat, Institut National Polytechnique de Toulouse, France, 2000.

[Zaytoon, 01]

Systèmes dynamiques hybrides. Collection Hermès. Paris France.

[Zerhouni et Alla, 92]

Zerhouni N., Alla H. (1992). Sur l'analyse des lignes de fabrication par réseaux de Petri continus, *Journal Européen des Systèmes Automatisés (RAIRO-APII-JESA)*, Vol.26, N°3, p.253-76.

[Zwingelstein, 95]

Zwingelstein G., Diagnostic des défaillances-Théorie et pratique pour les systèmes industriels. Traité des Nouvelles Technologies, série Diagnostic et Maintenance. Hermès, Paris, 1995.

Abstract

Hybrid systems involve both continuous and discrete variables. The continuous dynamics is generally given by differential-algebraic equations while the discrete dynamics is modelled by automata or input-output transition systems. For any industrial system, the early detection and diagnosis of faults is important, since a lot of damage and loss can result before a fault present in the system is detected. In addition, it becomes harder to distinguish the root cause of the fault as it propagates through the system. This is therefore more crucial in hybrid processes mixing both continuous and discrete aspects.

This work presents the development of a methodology associating the fault detection performed by a data-driven technique with the dynamic hybrid simulation for the diagnosis step. The detection is generally performed by comparing process measurement and simulation result of the system in normal conditions. This phase identifies symptoms. The problem of diagnosis is then to link them to a precise dysfunction. A possibility is therefore to explore all possible scenarios of faults and compare with actual measurements. Nevertheless the number of possibilities increases in an exponential way. The aim of the developed methodology is to restrict the detected fault to a category of failures. Only these failures are then explored.

The data-driven technique used in the proposed methodology is a fuzzy-classification method (LAMDA) enables to partition the data space in clusters related to identify symptoms. This method has the capacity to treat simultaneously quantitative and qualitative information and to propose automatic learning. It has been already used for detection of dysfunctions in complex chemical plants.

The second step of the procedure involves the dynamic hybrid simulation performed only for the restricted faults. In the framework of this study, the simulation aspects are ensured by the general object-oriented environment *ProODHyS(Process Object Dynamic Hybrid Simulator)*, designed and developed within the LGC. Its major characteristic is its ability to simulate systems described with *Object Differential Petri Nets (ODPN)* formalism. Each fault of this set is simulated. Then, the simulated scenarios are compared to the observed behaviour through a criterion composed of residues (the squared difference between the variables measured and the variable simulated with the fault). Finally, the diagnosis of the fault is performed by choosing the fault with the smallest residue.

Keywords: Fault diagnosis, Petri nets, Hybrid Dynamic Simulation. Classification.

