

THÈSE

Préparée aux

Laboratoire de Génie Chimique du CNRS
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

Nelly OLIVIER-MAGET

Ingénieur ENSIACET Toulouse

SURVEILLANCE DES SYSTÈMES DYNAMIQUES HYBRIDES :

APPLICATION AUX PROCÉDÉS

Soutenue le 12 décembre 2007, devant le jury composé de :

Rapporteurs : Jean-Pierre CORRIOU (Président)
 Sandro MACCHIETTO

Directeurs de thèses : Jean-Marc LE LANN
 Marie-Véronique LE LANN

Co-directeur de thèse : Gilles HÉTREUX

Examineurs : Jean-Christophe BUVAT
 Didier CAUDRON

Invité : Philippe BAUDET

À la mémoire de mon grand-père,

À mes parents et ma sœur,

À mon mari,

Les devises Shadok



EN ESSAYANT CONTINUUELLEMENT
ON FINIT PAR RÉUSSIR. DONC:
PLUS ÇA RATE, PLUS ON A
DE CHANCES QUE ÇA MARCHE.

REMERCIEMENTS

Enfin finie ! Je réalise aujourd'hui à quel point cette thèse a été un véritable challenge pour moi, mais aussi une incroyable aventure. Tant de sentiments contradictoires s'y sont entremêlés : l'enthousiasme, le découragement, la sérénité, les doutes. Je tiens donc à remercier ici un certain nombre de personnes pour m'avoir aidée à traverser cette épreuve.

Mes remerciements vont tout d'abord aux rapporteurs de cette thèse, Monsieur Sandro MACCHIETTO, professeur à l'Imperial College of London, et Monsieur Jean-Pierre CORRIOU, professeur à l'ENSIC de Nancy, pour l'intérêt qu'ils ont porté à mon travail. Leurs remarques, lors de la lecture de ce mémoire, ont permis d'apporter des précisions à ce dernier.

J'adresse ensuite mes sincères remerciements à Monsieur Jean-Claude BUVAT, professeur ENSAM à l'INSA de Rouen, et Monsieur Didier Caudron, Directeur Process Control à Sanofi Pasteur, d'avoir accepté de participer au jury de cette thèse.

Je remercie également la société ProSim pour le suivi et le soutien de ces travaux, et plus particulièrement Monsieur Philippe BAUDET, ingénieur dans cette société, pour sa participation à cette commission d'examen.

J'exprime toute mon amitié à mes deux directeurs de thèse, Jean-Marc et Marie-Véronique LE LANN. Malgré les difficultés rencontrées et les doutes que j'ai pu avoir, vous n'avez cessé d'être optimistes sur l'issue et je vous remercie pour toute la confiance que vous m'avez accordée.

Merci Jean-Marc de toujours t'être soucieux de mon bien-être, merci pour toutes ces franches rigolades que l'on a pu partager. Sache que tes venues en fin de journée ont toujours été très appréciées par tous les thésards du bureau, et c'est avec un grand plaisir que je repense à ces bons moments de détente ! Merci de m'avoir accompagnée et soutenue jusqu'au dernier moment.

Merci Marie-Véronique pour ton aide sur le plan scientifique qui a apporté de la rigueur au travail, mais aussi et surtout, merci pour tes qualités humaines... Ton soutien final a vraiment été déterminant pour mener à bien cette thèse. Encore merci !

Ensuite, j'ai une pensée émue pour Gilles, mon encadrant durant ces trois années de thèse. Nous n'avons jamais eu un rapport « professeur élève ». Bien au contraire, tu m'as considérée dès le début comme ta collègue. Je t'en suis profondément reconnaissante. Merci aussi pour toutes ces discussions, ces débats qui nous ont fait avancer dans ces recherches : toujours constructifs, toujours au plus haut. Cette thèse est le fruit de notre collaboration. Le perfectionnisme était au rendez-vous. Ce fut vraiment très enrichissant de travailler avec toi. Je te remercie du fond du cœur pour toute la confiance que tu as eue en moi. Maintenant, c'est à ton tour... Bonne rédaction !

J'exprime toute ma reconnaissance au département PSI pour leur accueil chaleureux et leur gentillesse. Un merci tout particulier à Catherine, qui a toujours suivi mes travaux. Et qui sait, peut-être qu'un jour, nous finirons par de nouveau travailler ensemble.

Je tiens plus particulièrement à remercier les équipes AFP et GI, mes anciens professeurs devenus aujourd'hui mes collègues. Merci à Xuan, Xavier, Pascal, Denis, Vincent, Raphaële, Steph, Alain, Philippe, Pascale, Sophie, Jean-Pierre, Christian, Chantal, Karim et Michel. Merci à vous pour votre soutien, pour cette ambiance si particulière, pour ces bons moments passés avec vous, pour ces crises de fous rires... Et un dernier merci pour avoir tout bonnement aidé à ce que la couleur de notre algéco paraisse un peu moins grise !

Cette thèse a également été une véritable aventure humaine faite de rencontres et de partages simples enrichissants et inoubliables. Merci pour toutes ces discussions qui n'ont absolument pas refait le monde, mais qui nous ont ouverts à d'autres cultures. Merci en particulier à Mohamed et Hilaire pour le continent africain, à Anton, Barnabas et Csaba pour le continent européen, et enfin pour l'Amérique du sud, merci à Galo, Ivonne, Guillermo et Alejandro. Merci pour tous ces souvenirs joyeux.

Enfin, cette thèse n'aurait pas pu aboutir sans le soutien inconditionnel de mes amis. Je tiens donc à remercier en premier lieu mes amis thésards, amis du bureau, pour cette aide précieuse :

- Françoise pour son écoute quotidienne, son réconfort, sa répartie et ses longues discussions, pour ces moments de détente remplies de gourmandises et de fous rires, et merci surtout pour son empathie,
- Flop pour m'avoir accompagnée dans ces grands « moments de solitude » qui me sont propres, pour toutes nos soirées interminables de détente (le squatteur attitré et toujours bienvenu de la maison !), pour nos débats et pour son rire inoubliable,
- Flof pour son humour caustique, pour nos grandes discussions sur notre avenir (et c'est promis un jour je t'apprendrais à faire de la purée de carottes),
- Mourad, le patriarche du bureau, comme on aimait l'appeler, qui, de sa lointaine Algérie, n'a cessé de m'envoyer des mails d'encouragements et de sympathie, que j'espère revoir un jour ou l'autre au détour d'un voyage,
- Et Bouchra, pour sa gentillesse, sa volonté, ses conseils et son soutien final.

Au-delà des amis rencontrés au travail, je tiens à remercier mes amis toulousains :

- Floc, Claire, Bob et Camille, pour les soirées entre amis en toute simplicité,
- Renaud pour ses lunettes et chapeaux, et parce qu'il le vaut bien,
- Ma petite Élé, pour tous ses conseils, ses confidences, nos virées du midi naturelles et plaisantes, pour toutes ces choses que nous avons partagées et qui nous restent encore à partager,
- Et ma grande amie Laurie, avec qui j'ai partagé tant de souvenirs mémorables (repas à la cyril, week-end à Seignosse, périple en avion, soirée dans les « étoiles »...), mon amie

qui a toujours été là pour moi, que j'adore pour ces grandes qualités humaines et sa droiture.

Et puis il y a les amis de longues dates, les amis d'Orléans, qui ont toujours été là dans les bons et mauvais moments, merci à vous tous d'être des amis exemplaires. La liste est longue mais je souhaite témoigner de leur amitié ici : Charly, Dadou, Lapin, Cyril, Guigui, Lolo, Juju, Marion, Oliv, la famille Delaporte, Guilhem, Sophie, la belle famille Lepain expatriée dans une contrée lointaine. Merci les amis. J'ai une pensée toute particulière pour mon p'tit Tom, mon pote « people ». Merci pour tous ces mails d'encouragements et de confiance, pour ces discussions philosophiques que tu as pu avoir sur nos répondeurs, merci de ce soutien sans faille qui réchauffe le cœur.

J'exprime ensuite toute mon affection à mes beaux-parents. Merci à tous les deux de m'avoir soutenue et intégrée au clan Maget.

Je tiens également à témoigner ici de tout mon amour à mes grands parents paternels et à ma p'tite mémé. Ils ont toujours veillé à mon bien-être, m'ont toujours encouragée et félicitée. Je vous en remercie tellement.

Pour finir, je tiens à remercier ma famille, présente à tous les instants, bons et mauvais, pour avoir vécu avec moi ces années, pour m'avoir toujours soutenue, écoutée et conseillée. Merci « Môman » pour avoir toujours été là, pour toutes nos longues conversations, merci de tes relectures en règle, merci « Papounet » pour m'avoir encouragée dans cette voie, merci pour toutes nos discussions scientifiques qui m'ont permis de voir le problème sous un autre angle. En somme merci à tous les deux de tout cet amour !

Toute ma gratitude va ensuite à ma p'tite sœur, ma meilleure amie, ma confidente, mon modèle. Merci d'avoir tout simplement été toujours là pour moi ! La prochaine c'est toi, et je serais là et fière.

Je ne peux finir ces remerciements sans écrire un dernier paragraphe sur mon mari. Nous avons traversé ensemble cette même épreuve. Les mots seront trop faibles pour dire à quel point je te remercie de partager ma vie.

TABLE DES MATIÈRES

| | |
|------------------------------|------------|
| REMERCIEMENTS | VII |
| TABLE DES MATIÈRES | XI |
| INTRODUCTION GÉNÉRALE | 1 |

PARTIE 1 : LE CONTEXTE DE L'ÉTUDE

| | |
|---|-----------|
| CHAPITRE 1 LE CADRE DE L'ÉTUDE | 9 |
| 1.1 LES SYSTÈMES INDUSTRIELS CONSIDÉRÉS : LES PROCÉDÉS | 11 |
| 1.1.1 La notion d'opérations unitaires | 11 |
| 1.1.2 Les différents flux | 13 |
| 1.1.3 Le mode de production par lots (ou discontinu) | 13 |
| 1.1.4 La notion de recette | 14 |
| 1.2 LES SYSTÈMES DYNAMIQUES HYBRIDES | 17 |
| 1.2.1 La définition générique | 17 |
| 1.2.2 La modélisation des systèmes dynamiques hybrides | 18 |
| 1.3 LA CONDUITE DE PROCÉDÉ AU SEIN DU SYSTÈME DE PILOTAGE | 19 |
| 1.3.1 La place de la conduite de procédé dans le modèle CIM | 19 |
| 1.3.2 La supervision | 21 |
| 1.3.2.1 Le pilotage | 21 |
| 1.3.2.2 La surveillance | 22 |
| 1.4 VERS UN SYSTÈME DE SURVEILLANCE AVANCÉ | 22 |
| 1.4.1 Les enjeux de la surveillance | 22 |
| 1.4.1.1 Le rôle de l'opérateur au sein de la supervision | 22 |
| 1.4.1.2 L'impact financier | 23 |
| 1.4.1.3 L'impact sécuritaire et environnemental des accidents | 24 |
| 1.4.2 Les limites des méthodes classiques de surveillance | 24 |
| 1.4.2.1 L'utilisation de la redondance | 24 |
| 1.4.2.2 Les méthodes basées sur le dépassement de seuil | 25 |
| 1.4.3 L'intérêt d'un modèle du procédé pour la surveillance | 28 |
| 1.4.4 Le cadre des travaux | 28 |
| 1.4.4.1 Les bases | 28 |
| 1.4.4.2 Les caractéristiques | 29 |
| 1.4.4.3 Les objectifs de nos travaux | 30 |
| 1.5 LE BILAN | 31 |
| CHAPITRE 2 LA PLATE-FORME DE SIMULATION <i>PRODHYS</i> | 33 |
| 2.1 LA SIMULATION DYNAMIQUE HYBRIDE | 35 |
| 2.1.1 Les concepts | 35 |
| 2.1.2 Les spécificités de la simulation des procédés | 36 |
| 2.1.2.1 Un modèle | 36 |
| 2.1.2.2 La résolution | 37 |

| | | |
|-----------|--|----|
| 2.1.2.2.1 | <i>La simulation modulaire</i> | 37 |
| 2.1.2.2.2 | <i>La simulation globale</i> | 37 |
| 2.1.2.3 | Le cas particulier des procédés discontinus | 37 |
| 2.2 | LES CONCEPTS FONDAMENTAUX DE <i>PRODHYS</i> | 38 |
| 2.3 | L'ARCHITECTURE LOGICIELLE DE <i>PRODHYS</i> | 39 |
| 2.4 | LE FORMALISME RDPDO | 41 |
| 2.4.1 | La spécification du comportement dynamique | 41 |
| 2.4.1.1 | La partie continue | 41 |
| 2.4.1.2 | La partie discrète | 42 |
| 2.4.1.3 | La structure de données | 42 |
| 2.4.2 | Les éléments du formalisme RdPDO | 42 |
| 2.4.2.1 | L'introduction des objets dans les réseaux de Petri | 43 |
| 2.4.2.2 | L'introduction des réseaux de Petri dans les objets | 44 |
| 2.4.2.3 | L'évolution du modèle RdPDO | 45 |
| 2.5 | LA MODÉLISATION D'UN PROCÉDÉ | 46 |
| 2.5.1 | Le modèle de simulation | 46 |
| 2.5.2 | La modélisation de la partie opérative | 47 |
| 2.5.2.1 | La modélisation selon l'axe topologique | 48 |
| 2.5.2.1.1 | <i>La modélisation topologique selon l'axe connexion</i> | 48 |
| 2.5.2.1.2 | <i>La modélisation topologique selon l'axe décomposition/composition</i> | 48 |
| 2.5.2.2 | La modélisation d'un procédé selon l'axe phénoménologique | 48 |
| 2.5.3 | La spécification de la recette | 48 |
| 2.5.3.1 | La connexion des RdP « appareils » et « recette » | 49 |
| 2.5.3.2 | La mise en œuvre de macro-place | 51 |
| 2.6 | L'APPLICATION | 52 |
| 2.6.1 | Le procédé étudié | 52 |
| 2.6.1.1 | La description | 52 |
| 2.6.1.2 | Le modèle mathématique | 53 |
| 2.6.2 | La modélisation du procédé | 54 |
| 2.6.2.1 | La décomposition | 54 |
| 2.6.2.2 | La modélisation des appareils | 56 |
| 2.6.2.2.1 | <i>L'alimentation matière</i> | 56 |
| 2.6.2.2.2 | <i>Le détecteur de niveau</i> | 57 |
| 2.6.2.2.3 | <i>La cuve de stockage</i> | 58 |
| 2.6.2.2.4 | <i>La vanne</i> | 60 |
| 2.6.3 | La modélisation de la recette | 64 |
| 2.6.4 | La simulation | 65 |
| 2.7 | LA CONCLUSION | 67 |

PARTIE 2 : L'ÉTAT DE L'ART

| | |
|---|-----------|
| CHAPITRE 3 LES TECHNIQUES DE SURVEILLANCE | 71 |
| 3.1 LA TERMINOLOGIE | 73 |
| 3.1.1 Le système physique | 75 |
| 3.1.2 Les observations | 77 |
| 3.1.3 Les anomalies | 77 |
| 3.1.4 La surveillance | 79 |
| 3.1.4.1 La détection | 79 |
| 3.1.4.2 Le diagnostic | 80 |
| 3.1.4.3 La surveillance prédictive | 80 |
| 3.1.5 Les corrections | 80 |
| 3.2 LA CLASSIFICATION DES MÉTHODES DE DIAGNOSTIC | 81 |
| 3.3 LES MÉTHODES SANS MODÈLE OU À BASE DE MODÈLES COMPORTEMENTAUX | 83 |
| 3.3.1 Les méthodes à base de connaissance | 83 |
| 3.3.1.1 Les méthodes d'analyse des défaillances | 83 |
| 3.3.1.1.1 <i>L'AMDE et ses dérivées</i> | 83 |
| 3.3.1.1.2 <i>Les arbres de défaillances ou arbres de causes</i> | 85 |
| 3.3.1.2 Les systèmes experts | 86 |
| 3.3.2 Les méthodes basées sur le traitement de données | 88 |
| 3.3.2.1 Les techniques d'extraction statistique de données | 88 |
| 3.3.2.1.1 <i>L'analyse en composantes principales</i> | 88 |
| 3.3.2.1.2 <i>La classification de données et reconnaissance de formes</i> | 89 |
| 3.3.2.2 Les méthodes d'extraction non statistique | 91 |
| 3.3.2.2.1 <i>L'approche fréquentielle</i> | 91 |
| 3.3.2.2.2 <i>Les réseaux de neurones</i> | 92 |
| 3.3.2.2.3 <i>L'analyse qualitative des tendances</i> | 93 |
| 3.4 LES MÉTHODES À BASE DE MODÈLES PHÉNOMÉNOLOGIQUES | 94 |
| 3.4.1 Les modèles quantitatifs | 94 |
| 3.4.1.1 Les relations de parité | 94 |
| 3.4.1.2 L'estimation de paramètres | 95 |
| 3.4.1.3 L'estimation d'état | 97 |
| 3.4.2 Les modèles qualitatifs ou semi-qualitatifs | 98 |
| 3.4.2.1 Les équations différentielles qualitatives | 99 |
| 3.4.2.2 Les graphes d'influence/causaux | 99 |
| 3.4.3 Les modèles flous | 100 |
| 3.4.3.1 La logique floue | 100 |
| 3.4.3.2 Les modèles linguistiques flous | 101 |
| 3.4.3.3 Les modèles flous type Takagi-Sugeno-Kang | 102 |
| 3.5 LA SYNTHÈSE DES MÉTHODES | 102 |
| 3.6 LA FORMULATION DU PROBLÈME | 108 |
| 3.6.1 La surveillance à base de modèles | 108 |
| 3.6.1.1 La structure | 108 |

| | | |
|---|--|------------|
| 3.6.1.2 | Les tests de détection | 109 |
| 3.6.1.3 | Le diagnostic | 110 |
| 3.6.1.3.1 | <i>Les résidus directionnels</i> | 110 |
| 3.6.1.3.2 | <i>Les résidus structurés</i> | 111 |
| 3.7 | LE BILAN | 112 |
| CHAPITRE 4 LA GÉNÉRATION DE RÉSIDUS À L'AIDE DE RECONSTRUCTEURS D'ÉTAT | | 113 |
| 4.1 | LES NOTIONS D'OBSERVABILITÉ ET D'OBSERVATEUR | 115 |
| 4.1.1 | L'observabilité | 116 |
| 4.1.2 | Les observateurs | 117 |
| 4.1.2.1 | Les observateurs des systèmes linéaires | 117 |
| 4.1.2.2 | Les observateurs des systèmes non – linéaires | 118 |
| 4.1.3 | La synthèse des observateurs de systèmes dynamiques hybrides | 119 |
| 4.2 | LE FILTRE DE KALMAN | 119 |
| 4.2.1 | Le principe | 120 |
| 4.2.2 | La linéarisation des problèmes non linéaires | 122 |
| 4.2.2.1 | Le filtre de Kalman linéarisé | 123 |
| 4.2.2.2 | Le filtre de Kalman étendu | 125 |
| 4.3 | L'EXPLOITATION DES RÉSIDUS POUR DES ÉTUDES DE DIAGNOSTIC | 127 |
| 4.3.1 | Le principe de l'approche | 127 |
| 4.3.2 | La génération de résidus | 127 |
| 4.3.3 | La détection | 128 |
| 4.3.3.1 | Un choix difficile | 128 |
| 4.3.3.2 | Les seuils fixes et adaptatifs | 129 |
| 4.3.4 | Le diagnostic | 130 |
| 4.3.4.1 | La table d'incidence et la signature de défaut | 130 |
| 4.3.4.2 | Le processus | 131 |
| 4.4 | LA SYNTHÈSE | 132 |
| <u>PARTIE 3 : LA SURVEILLANCE : CONCEPTS ET MISE EN OEUVRE</u> | | |
| CHAPITRE 5 LA FORMALISATION DE LA DÉTECTION ET LE DIAGNOSTIC DE FAUTES | | 135 |
| 5.1 | L'ARCHITECTURE GÉNÉRALE DU SYSTÈME DE SURVEILLANCE <i>FDIPRO</i> | 137 |
| 5.2 | LA GÉNÉRATION DE RÉSIDUS AVEC LE FILTRE DE KALMAN ÉTENDU | 138 |
| 5.2.1 | Le modèle utilisé | 139 |
| 5.2.2 | L'observateur : le filtre de Kalman étendu | 140 |
| 5.2.3 | La génération de résidus | 142 |
| 5.3 | L'ÉVALUATION DE RÉSIDUS | 143 |
| 5.3.1 | Un travail sur fenêtre glissante | 143 |
| 5.3.2 | La génération des signatures | 144 |
| 5.4 | LA DÉTERMINATION DE LA MATRICE D'INCIDENCE | 146 |
| 5.5 | LA LOCALISATION DE DÉFAUTS | 148 |

| | | |
|--|---|------------|
| 5.5.1 | Le principe | 148 |
| 5.5.2 | Les distances | 150 |
| 5.5.3 | La prise de décision | 154 |
| 5.6 | LA SYNTHÈSE | 155 |
| CHAPITRE 6 L'INTÉGRATION DE LA SURVEILLANCE SOUS <i>PRODHYS</i> | | 157 |
| 6.1 | LES ÉLÉMENTS CONSTITUTIFS DU MODULE <i>PRODHYS</i> <i>AEM</i> | 160 |
| 6.1.1 | La reconstruction d'état | 160 |
| 6.1.1.1 | Le filtre de Kalman étendu | 161 |
| 6.1.1.2 | Le modèle dynamique | 162 |
| 6.1.2 | La détection | 163 |
| 6.1.2.1 | La génération de résidus et de signatures | 163 |
| 6.1.2.2 | La structure de données | 163 |
| 6.1.2.2.1 | Les résidus | 163 |
| 6.1.2.2.2 | Les tables | 164 |
| 6.1.3 | Le diagnostic | 165 |
| 6.1.3.1 | La signature de défaut théorique | 165 |
| 6.1.3.2 | La localisation et l'identification de défauts | 166 |
| 6.1.3.3 | La distance | 166 |
| 6.1.4 | La surveillance | 166 |
| 6.1.4.1 | La création du module de surveillance | 167 |
| 6.1.4.2 | L'initialisation du module de surveillance | 167 |
| 6.1.4.3 | L'exécution du module de surveillance | 168 |
| 6.2 | LE MODÈLE DE SIMULATION DU PROCÉDÉ DÉFECTUEUX | 169 |
| 6.2.1 | Les anomalies prises en compte | 169 |
| 6.2.2 | Le principe | 170 |
| 6.2.2.1 | La modélisation d'un défaut | 170 |
| 6.2.2.2 | La modélisation des appareils | 170 |
| 6.2.3 | La simulation du procédé défectueux | 171 |
| 6.2.3.1 | La génération de défaut | 171 |
| 6.2.3.2 | La simulation en parallèle | 171 |
| 6.3 | LA MODÉLISATION DES APPAREILS | 173 |
| 6.3.1 | Les capteurs | 174 |
| 6.3.1.1 | Le capteur sans défaut | 175 |
| 6.3.1.2 | Le capteur avec défaut | 177 |
| 6.3.2 | Les détecteurs | 178 |
| 6.3.2.1 | Le détecteur sans défaut | 179 |
| 6.3.2.2 | Le détecteur avec défaut | 179 |
| 6.3.3 | Les alimentations | 181 |
| 6.3.3.1 | L'alimentation matière sans défaut | 182 |
| 6.3.3.2 | L'alimentation matière avec défaut | 182 |
| 6.3.4 | Les vannes | 185 |
| 6.3.4.1 | La vanne sans défaut | 186 |
| 6.3.4.1.1 | La vanne simple | 186 |

| | | |
|---|---|------------|
| 6.3.4.1.2 | <i>La vanne à inversion de flux</i> | 186 |
| 6.3.4.2 | La vanne avec défaut | 186 |
| 6.4 | LA CONCLUSION | 189 |
| <u>PARTIE 4 : LA MISE EN OEUVRE DE LA MÉTHODOLOGIE DE SURVEILLANCE</u> | | |
| <u>CHAPITRE 7 L'IMPLEMENTATION DU SYSTÈME DE SURVEILLANCE DE PROCÉDÉ</u> | | 193 |
| 7.1 | LE PROCESSUS GÉNÉRAL DE MODÉLISATION | 195 |
| 7.2 | LA MODÉLISATION DU PROCÉDÉ | 196 |
| 7.2.1 | Le procédé de référence | 196 |
| 7.2.2 | Le procédé réel | 197 |
| 7.3 | LA MODÉLISATION DE LA RECETTE PRINCIPALE | 198 |
| 7.3.1 | La recette pilotant le procédé | 198 |
| 7.3.1.1 | Le procédé de référence | 198 |
| 7.3.1.2 | Le procédé réel émulé | 199 |
| 7.3.1.3 | Le procédé reconstruit par le filtre de Kalman étendu | 200 |
| 7.3.2 | La recette de génération de défauts | 200 |
| 7.3.3 | La recette générale | 200 |
| 7.4 | LA MISE EN OEUVRE DU FILTRE DE KALMAN | 202 |
| 7.4.1 | Les équations des observations | 202 |
| 7.4.2 | Le réglage du filtre | 203 |
| 7.4.2.1 | Les bruits de mesure | 203 |
| 7.4.2.2 | Les bruits de modèle | 203 |
| 7.4.2.3 | L'initialisation au changement de modèle | 206 |
| 7.5 | LA MATRICE D'INCIDENCE | 207 |
| 7.5.1 | Les défauts pris en compte | 207 |
| 7.5.2 | La construction initiale | 207 |
| 7.6 | LES RÉSULTATS | 210 |
| 7.6.1 | Le scénario | 210 |
| 7.6.2 | La génération de signatures | 210 |
| 7.6.3 | La localisation et l'identification de défauts | 212 |
| 7.7 | LE BILAN | 213 |
| <u>CHAPITRE 8 L'EXPLOITATION DU SYSTÈME DE SURVEILLANCE</u> | | 215 |
| 8.1 | L'OPÉRATION DE CHANGEMENT DE SOLVANT : UN DÉFAUT DE TEMPÉRATURE | 217 |
| 8.1.1 | Le procédé mis en œuvre | 217 |
| 8.1.2 | La modélisation | 219 |
| 8.1.2.1 | La modélisation du procédé | 219 |
| 8.1.2.1.1 | <i>Le procédé de référence</i> | 219 |
| 8.1.2.1.2 | <i>Le procédé réel</i> | 220 |
| 8.1.2.2 | La modélisation de la recette | 221 |
| 8.1.3 | Les résultats | 222 |

| | | |
|------------|---|------------|
| 8.1.3.1 | La simulation du modèle de référence | 223 |
| 8.1.3.2 | La détection | 224 |
| 8.1.3.3 | Le diagnostic | 225 |
| 8.2 | LE BENCHMARK SDH : APPARITION DE DÉFAUTS MULTIPLES | 227 |
| 8.2.1 | Le procédé mis en œuvre | 228 |
| 8.2.2 | La modélisation | 228 |
| 8.2.2.1 | La modélisation du procédé | 228 |
| 8.2.2.1.1 | <i>Le procédé de référence</i> | 228 |
| 8.2.2.1.2 | <i>Le procédé réel</i> | 229 |
| 8.2.2.2 | La modélisation de la recette | 231 |
| 8.2.3 | Les résultats | 231 |
| 8.2.3.1 | La simulation du modèle de référence | 231 |
| 8.2.3.2 | La détection | 235 |
| 8.2.3.3 | Le diagnostic | 235 |
| 8.3 | LA SURVEILLANCE D'UN PROCÉDÉ COMPLEXE | 237 |
| 8.3.1 | Le procédé mis en œuvre | 237 |
| 8.3.2 | La procédure | 239 |
| 8.3.3 | La modélisation | 240 |
| 8.3.3.1 | La modélisation du procédé | 240 |
| 8.3.3.1.1 | <i>Le procédé de référence</i> | 240 |
| 8.3.3.1.2 | <i>Le procédé réel</i> | 240 |
| 8.3.3.2 | La modélisation de la recette | 243 |
| 8.3.3.2.1 | <i>Le niveau procédure</i> | 243 |
| 8.3.3.2.2 | <i>Le niveau opération</i> | 244 |
| 8.3.3.2.3 | <i>Le niveau phase</i> | 244 |
| 8.3.4 | Les résultats | 245 |
| 8.3.4.1 | La simulation du modèle de référence | 245 |
| 8.3.4.2 | La détection | 246 |
| 8.3.4.3 | Le diagnostic | 248 |
| 8.4 | LA CONCLUSION | 250 |
| | CONCLUSION GÉNÉRALE | 251 |
| | ANNEXES | 257 |
| | Annexe A : Les concepts orientés objet et le langage UML | 259 |
| | Annexe B : La définition formelle d'un RdPDO | 273 |
| | Annexe C : Le noyau numérique : <i>DISCo</i> | 279 |
| | Annexe D : Les distances de Manhattan relative et améliorée | 287 |
| | Annexe E : Les structures DAEMatrix et DAEVector | 291 |
| | NOMENCLATURE | 295 |
| | BIBLIOGRAPHIE | 301 |
| | LISTE DES PUBLICATIONS ET COMMUNICATIONS | 315 |

LISTE DES ILLUSTRATIONS

| | |
|--|----|
| • Liste des figures | |
| Figure 1.1 Les étapes de fabrication d'un produit chimique | 11 |
| Figure 1.2 Une variante d'un procédé discontinu décrit dans [Joglekar et al., 1985] | 13 |
| Figure 1.3 L'utilisation des ressources en production discontinue (alternance de régimes permanents et transitoires) | 14 |
| Figure 1.4 Les types de recette et leurs contenus | 15 |
| Figure 1.5 La hiérarchisation de la procédure | 16 |
| Figure 1.6 La modélisation d'un système dynamique hybride [Tittus, 1995] | 17 |
| Figure 1.7 La pyramide du modèle CIM | 19 |
| Figure 1.8 La structure logique du niveau conduite | 20 |
| Figure 1.9 Des synoptiques de systèmes de supervision commerciaux | 23 |
| Figure 1.10 La redondance matérielle | 24 |
| Figure 1.11 Des exemples de redondance matérielle | 25 |
| Figure 1.12 La notion de référentiel | 25 |
| Figure 1.13 Les seuils des alarmes | 26 |
| Figure 1.14 Les limites de l'observation | 27 |
| Figure 1.15 La redondance analytique | 28 |
| Figure 1.16 La procédure de diagnostic | 30 |
| Figure 2.1 L'architecture générale d'un simulateur dynamique hybride | 36 |
| Figure 2.2 L'architecture logicielle de PrODHyS | 40 |
| Figure 2.3 La sémantique du formalisme RdPDO | 43 |
| Figure 2.4 L'approche « objet dans les réseaux de Petri » | 44 |
| Figure 2.5 L'approche « réseau de Petri dans les objets » | 45 |
| Figure 2.6 Le principe d'évolution des RdPDO dans le modèle simplifié de la cuve | 46 |
| Figure 2.7 La modélisation topologique | 47 |
| Figure 2.8 La place de signal d'un dispositif actif | 49 |
| Figure 2.9 Les interactions entre la partie commande et la partie procédé | 49 |
| Figure 2.10 La vision du réseau de Petri de commande utilisée par convention | 50 |
| Figure 2.11 La notion de Macro-Place sous PrODHyS | 51 |
| Figure 2.12 La modélisation UML de la procédure | 51 |
| Figure 2.13 Les vases communicants | 52 |
| Figure 2.14 Le schéma bloc associé au système des vases communicants | 55 |
| Figure 2.15 Le processus de spécialisation d'appareils élémentaires | 55 |
| Figure 2.16 Le réseau de Petri de l'alimentation instantanée | 56 |
| Figure 2.17 Le réseau de Petri d'un détecteur | 57 |
| Figure 2.18 Le réseau de Petri de la cuve de stockage | 58 |
| Figure 2.19 Le réseau de Petri de la vanne | 60 |
| Figure 2.20 Le réseau de Petri de la vanne à inversion de flux | 62 |
| Figure 2.21 La recette associée au système des vases communicants | 64 |
| Figure 2.22 Les évolutions du flux et des niveaux de liquide dans les bacs C1 et C2 | 66 |
| Figure 2.23 Les évolutions des fractions molaires liquide en eau et méthanol dans les bacs C1 et C2 | 66 |

| | |
|---|-----|
| Figure 3.1 La boucle de supervision | 74 |
| Figure 3.2 Le modèle phénoménologique | 76 |
| Figure 3.3 Le modèle comportemental | 76 |
| Figure 3.4 Un exemple de détection grâce à un émulateur | 79 |
| Figure 3.5 Les méthodes selon leur degré de connaissance et de nécessité de données expérimentales | 81 |
| Figure 3.6 La typologie des méthodes de diagnostic | 82 |
| Figure 3.7 L'arbre de défaillances | 85 |
| Figure 3.8 La structure d'un système expert [Kempowski, 2004] | 86 |
| Figure 3.9 Un exemple de base de connaissance d'un système expert | 87 |
| Figure 3.10 Une illustration de l'analyse en composantes principales | 89 |
| Figure 3.11 La reconnaissance de formes | 90 |
| Figure 3.12 Un neurone formel | 92 |
| Figure 3.13 Les relations de parité [Touaf, 2005] | 94 |
| Figure 3.14 L'estimation de paramètres [Frank, 1996] | 96 |
| Figure 3.15 L'estimation d'état | 97 |
| Figure 3.16 La fonction d'appartenance | 101 |
| Figure 3.17 La structure de la surveillance | 108 |
| Figure 3.18 Les résidus directionnels | 111 |
| Figure 4.1 Les notations du système | 115 |
| Figure 4.2 L'organigramme de principe du filtre de Kalman | 121 |
| Figure 4.3 La trajectoire réelle et nominale du filtre de Kalman linéarisé | 123 |
| Figure 4.4 Le schéma bloc du filtre de Kalman étendu | 126 |
| Figure 4.5 Le diagnostic FDI | 127 |
| Figure 4.6 Les incertitudes sur la détection | 128 |
| Figure 4.7 Les différents seuils | 129 |
| Figure 4.8 Une table d'incidence non localisante | 131 |
| Figure 4.9 Une table d'incidence déterministiquement localisante | 131 |
| Figure 4.10 Une table d'incidence statistiquement localisante | 131 |
| Figure 5.1 L'architecture générale du système de surveillance FDIPro | 137 |
| Figure 5.2 La génération de résidus | 138 |
| Figure 5.3 Le schéma bloc du système | 140 |
| Figure 5.4 Les résidus absolus | 143 |
| Figure 5.5 La génération de résidus sur fenêtre glissante | 144 |
| Figure 5.6 L'évaluation des résidus | 144 |
| Figure 5.7 La génération de la matrice d'incidence | 146 |
| Figure 5.8 La matrice d'incidence reconstruite | 147 |
| Figure 5.9 Le système de localisation | 148 |
| Figure 5.10 Une signature de défaut et une signature de résidu | 149 |
| Figure 5.11 Un exemple de calcul de distance de Hamming | 150 |
| Figure 5.12 La signature de défauts multiples | 152 |
| Figure 5.13 Les tables des distances (Manhattan relative et améliorée) et d'indicateurs de défauts correspondants | 153 |
| Figure 5.14 La loi normale centrée réduite $\mathfrak{N}(moy, \sigma)$ | 154 |

| | |
|---|-----|
| Figure 6.1 La structure de la plate-forme PrODHyS | 159 |
| Figure 6.2 Les classes associées aux éléments constitutifs de l'approche SimAEM | 160 |
| Figure 6.3 Le diagramme de classe du filtre de Kalman | 161 |
| Figure 6.4 Le corps de la méthode <code>perform</code> du filtre de Kalman | 161 |
| Figure 6.5 Le diagramme de classe du modèle | 162 |
| Figure 6.6 Le diagramme de classe de la détection | 163 |
| Figure 6.7 Le diagramme de classe des résidus | 164 |
| Figure 6.8 Le diagramme de classe de la table | 164 |
| Figure 6.9 La structure de données | 165 |
| Figure 6.10 Le principe de la méthode <code>tableUpdate</code> | 165 |
| Figure 6.11 Le diagramme de classe de la signature de défaut | 165 |
| Figure 6.12 Le diagramme de classe du diagnostic | 166 |
| Figure 6.13 Le diagramme de classe de la distance | 166 |
| Figure 6.14 Le diagramme de classe de la surveillance | 167 |
| Figure 6.15 Le diagramme de séquence de la construction de <code>monitoring</code> | 167 |
| Figure 6.16 Le diagramme de séquence de l'initialisation de <code>monitoring</code> | 168 |
| Figure 6.17 Le diagramme de séquence de l'exécution de <code>monitoring</code> | 168 |
| Figure 6.18 Les sources de défaillances | 169 |
| Figure 6.19 Le diagramme de classe d'un défaut | 170 |
| Figure 6.20 Le réseau de Petri associé au générateur de défaut | 171 |
| Figure 6.21 Le réseau de Petri de la recette générale | 172 |
| Figure 6.22 Les capteurs spécialisés | 174 |
| Figure 6.23 Le réseau de Petri du capteur non défaillant | 175 |
| Figure 6.24 Le réseau de Petri du capteur défaillant | 176 |
| Figure 6.25 Les notions de tolérance et valeur bruitée | 177 |
| Figure 6.26 Le principe d'un détecteur | 178 |
| Figure 6.27 Les détecteurs spécialisés | 179 |
| Figure 6.28 Le réseau de Petri du détecteur défaillant | 180 |
| Figure 6.29 Les alimentations spécialisées | 182 |
| Figure 6.30 Le réseau de l'alimentation matière défaillante | 183 |
| Figure 6.31 Les vannes spécialisées | 185 |
| Figure 6.32 Le réseau de Petri de la vanne à inversion de flux défaillante | 187 |
| Figure 7.1 L'exemple didactique | 196 |
| Figure 7.2 Le schéma bloc associé au système réel des vases communicants | 197 |
| Figure 7.3 La recette de contrôle associée au système des vases communicants | 198 |
| Figure 7.4 La recette de génération de défaut associée au système des vases communicants | 200 |
| Figure 7.5 La recette générale associée au système des vases communicants | 201 |
| Figure 7.6 L'utilisation d'une méthode d'ensemble pour la spécification des statistiques d'erreur du modèle | 203 |
| Figure 7.7 L'utilisation d'une méthode d'ensemble pour la spécification des statistiques d'erreur du modèle | 204 |
| Figure 7.8 L'initialisation du filtre de Kalman étendu | 206 |
| Figure 7.9 Le chronogramme d'état continu et discret du système | 208 |

| | |
|--|-----|
| Figure 7.10 Les résidus caractérisant l'occurrence du défaut | 208 |
| Figure 7.11 La reconstruction d'état obtenue par le filtre de Kalman étendu | 211 |
| Figure 7.12 La détection | 211 |
| Figure 8.1 La modélisation du procédé de référence d'ajouts-évaporations | 219 |
| Figure 8.2 Le schéma bloc associé au système réel d'ajouts-évaporations | 220 |
| Figure 8.3 La recette décrivant le procédé d'ajouts-évaporations | 221 |
| Figure 8.4 L'évolution de la composition liquide dans le réacteur | 223 |
| Figure 8.5 L'évolution de la rétention molaire liquide dans le réacteur | 224 |
| Figure 8.6 La détection du défaut « chauffe » lors du changement de solvant | 224 |
| Figure 8.7 Le flowsheet du benchmark SDH | 227 |
| Figure 8.8 La modélisation du benchmark SDH | 228 |
| Figure 8.9 Le schéma bloc associé au système réel du benchmark SDH | 229 |
| Figure 8.10 Le réseau de Petri de la recette du benchmark SDH | 230 |
| Figure 8.11 Le scénario étudié | 232 |
| Figure 8.12 Le chronogramme d'état | 232 |
| Figure 8.13 L'évolution des débits traversant les vannes V1 et V4 au cours du temps | 233 |
| Figure 8.14 L'évolution des hauteurs de liquide des cuves C1 et C2 au cours du temps | 233 |
| Figure 8.15 La détection des défauts sur le benchmark SDH | 234 |
| Figure 8.16 Le schéma du procédé [Joglekar et al., 1985] | 237 |
| Figure 8.17 La vitesse de la réaction en fonction de la température | 238 |
| Figure 8.18 La recette de fabrication du produit P | 239 |
| Figure 8.19 La modélisation du procédé [Joglekar et al., 1985] | 241 |
| Figure 8.20 Le schéma bloc associé au système réel du procédé [Joglekar et al., 1985] | 242 |
| Figure 8.21 Le réseau de Petri du niveau Procédure | 243 |
| Figure 8.22 Le réseau de Petri du niveau Opération : opération séparation | 244 |
| Figure 8.23 Le réseau de Petri du niveau Phase : phase distillation | 245 |
| Figure 8.24 L'évolution de la composition dans BR1 | 246 |
| Figure 8.25 La réaction dans BR1 | 246 |
| Figure 8.26 La détection d'une dégradation | 248 |
| • Liste des tableaux | |
| Tableau 2.1 Les données techniques des vases communicants | 53 |
| Tableau 2.2 Les conditions et actions associées aux transitions du réseau de Petri de l'alimentation instantanée | 56 |
| Tableau 2.3 Les conditions et actions associées aux transitions du réseau de Petri du détecteur | 57 |
| Tableau 2.4 Les conditions et actions associées aux transitions du réseau de Petri de la cuve de stockage | 59 |
| Tableau 2.5 Les conditions et actions associées aux transitions du réseau de Petri de la vanne | 61 |
| Tableau 2.6 Les conditions et actions associées aux transitions du réseau de Petri de la vanne à inversion de flux | 63 |
| Tableau 2.7 Les conditions initiales de la simulation des vases communicants | 65 |
| Tableau 3.1 Les anomalies | 79 |
| Tableau 3.2 L'AMDEC | 84 |

| | |
|--|-----|
| Tableau 3.3 L'analyse comparative des méthodes dédiées à la détection et au diagnostic de fautes : tableau récapitulatif | 103 |
| Tableau 3.4 Les couples formalisme/technique de détection | 109 |
| Tableau 6.1 Les conditions et actions associées aux transitions du réseau de Petri du capteur sans défaut | 175 |
| Tableau 6.2 Les conditions et actions associées aux transitions du réseau de Petri du capteur avec défaut | 176 |
| Tableau 6.3 Les conditions et actions associées aux transitions du réseau de Petri du détecteur avec défaut | 181 |
| Tableau 6.4 Les conditions et actions associées aux transitions du réseau de Petri de l'alimentation matière avec défauts | 184 |
| Tableau 6.5 Les conditions et actions associées aux transitions du réseau de Petri de la vanne à inversion de flux avec défaut | 188 |
| Tableau 7.1 Les conditions opératoires de l'exemple didactique | 196 |
| Tableau 7.2 Les incertitudes appliquées sur les processus, paramètres et conditions initiales du modèle | 204 |
| Tableau 7.3 Les incertitudes du modèle | 206 |
| Tableau 7.4 La signature théorique du défaut 1 | 209 |
| Tableau 7.5 La matrice d'incidence | 209 |
| Tableau 7.6 La signature de défauts instantanée | 212 |
| Tableau 7.7 Les indicateurs de défauts de l'exemple didactique | 212 |
| Tableau 8.1 Les conditions opératoires du procédé d'ajouts-évaporations | 223 |
| Tableau 8.2 La signature de défauts instantanée | 225 |
| Tableau 8.3 Les indicateurs de défauts de l'exemple didactique | 226 |
| Tableau 8.4 Les conditions opératoires du benchmark SDH | 232 |
| Tableau 8.5 La signature de défauts instantanée | 235 |
| Tableau 8.6 Les indicateurs de défauts de l'exemple didactique | 236 |
| Tableau 8.7 La signature de défauts instantanée | 248 |
| Tableau 8.8 Les indicateurs de défauts de l'exemple didactique | 249 |

INTRODUCTION GÉNÉRALE

Ces trente dernières années, l'évolution de la conjoncture a conduit à l'apparition d'une production diversifiée de petites et moyennes séries, afin de satisfaire la demande en termes de quantité, de gamme de produits. Ces systèmes sont des systèmes à traitement par lots, appelés couramment « procédés batch ». Ils s'adaptent rapidement à des campagnes de produits. L'atelier est ainsi flexible et polyvalent.

Parallèlement, ce type de production a amené une exigence en termes de qualité. Cette qualité constante se traduit par un gain de productivité. Les écarts de conduite ont alors des conséquences irréversibles pour le produit [Vanbergen, 1994]. L'automatisation et l'informatisation apportent une réponse, adéquate à ce besoin de reproductibilité, nécessaire à l'augmentation des performances du système de production, à la garantie de la qualité du produit et à la diminution des coûts de fabrication. Un autre objectif des systèmes de contrôle – commande est d'améliorer la sécurité de l'installation industrielle, des hommes y travaillant et de son environnement. S'ajoutent à ces objectifs, pour des industries spécifiques, telles que l'agroalimentaire, la traçabilité du produit et la satisfaction des normes de qualité.

Ainsi, l'intensification des procédés s'accompagne d'un effort de recherche considérable sur les performances des systèmes de supervision. Vu la taille et la complexité des installations industrielles, ainsi que les enjeux financiers impliqués, la détection et le diagnostic de pannes représentent l'un des défis les plus importants de ce domaine.

Traditionnellement, cet aspect de la supervision des procédés était sous la responsabilité des opérateurs humains, éventuellement aidés par un ensemble de capteurs et de détecteurs. Néanmoins cet appareillage a été mis en place pour le contrôle du procédé et non pour la détection et le diagnostic de défauts. De plus en phase transitoire, le nombre de variables à considérer est important, et parmi elles, beaucoup sont non observables, ce qui rend l'interprétation du comportement du procédé non triviale [Sundarraman et Srinivasan, 2003]. En conséquence, le développement de nouvelles approches plus élaborées est indispensable pour une surveillance robuste.

Ces méthodes se différencient par le type de connaissances qu'elles utilisent. On peut ainsi les classer de la façon suivante :

- *les méthodes à base de modèles* : elles reposent sur un modèle structurel du comportement du système dynamique hybride, basé sur les lois fondamentales régissant les phénomènes physicochimiques présents. Le modèle peut alors être *quantitatif* (par exemple un système d'équations différentielles algébriques), *qualitatif* ou *semi qualitatif* (par exemple un ensemble de relations logiques), ou encore *flou* (représentation des caractéristiques d'un système à l'aide de règles floues qui décrivent son comportement) ;
- *les méthodes à base de connaissances* : les connaissances, les compétences et le raisonnement des experts humains sont valorisés dans ces méthodes. Ils sont en effet traduits en règles afin de résoudre le problème ;
- et *les méthodes à base de données historiques* : de grandes quantités de données historiques du processus sont alors nécessaires. Deux types de méthodes sont distingués : les méthodes issues de la statistique et les méthodes dites d'apprentissage, de classification et de reconnaissance de formes.

L'apport de ces nouvelles approches est incontestable. Il est de plus indéniable, que, pour mettre en place de tels systèmes de surveillance, cela nécessite des connaissances approfondies de l'installation : connaissance de son comportement normal mais aussi et surtout de son comportement anormal. Ainsi, ces systèmes requièrent, pour leur bon déroulement, des informations sur des grandeurs non mesurées. L'intérêt de disposer d'un algorithme estimant ces grandeurs à partir de mesures du système physique est alors décisif. Cet algorithme est aussi appelé *capteur logiciel*.

Ici, nous distinguons deux types de grandeurs inconnues :

- les grandeurs qui sont constantes ou qui évoluent peu dans le temps ; nous les appelons *paramètres*. Notons que l'estimation de ces grandeurs relève du domaine de *l'identification paramétrique* [Ljung, 1987 ; Söderström et Stoica, 1989].
- les grandeurs qui évoluent de façon significative dans le temps ; nous les nommons *variables*. Les algorithmes d'estimation d'état sont dans le cas déterministe des *observateurs* [Gauthier et Kupka, 2001] et dans le cas stochastique des filtres [Jazwinski, 1970].

Les travaux développés au sein de l'équipe Génie Industriel du Laboratoire Génie Chimique portent sur la conception et la mise au point de méthodologies et d'outils de la décision permettant dans un environnement intégré, la maîtrise et la gestion de la chaîne logistique *produit-procédé-processus* par la prise en compte d'environnements liés à la notion de l'entreprise étendue dans son contexte technologique et socio-économique. L'équipe de recherche associe ses compétences dans des thématiques disciplinaires précises et ciblées selon le triptyque gestion de l'innovation et des projets innovants, gestion de la chaîne logistique de production et gestion intégrée des systèmes d'information et d'aide à la décision afin d'aborder des problématiques du génie industriel. Le but est donc de proposer un environnement de recherche, des thématiques et des outils propres au génie industriel dans un contexte métier génie des procédés. Par définition, il s'agit d'une recherche transdisciplinaire.

Notre travail se situe dans un contexte voisin puisqu'il porte sur la supervision avancée des systèmes dynamiques hybrides, en particulier des procédés. Il s'effectue en collaboration avec le Laboratoire d'Analyse et d'Architecture des Systèmes (Groupe DISCO (*Diagnostic Supervision CONduite*), Professeur Marie-Véronique LE LANN). L'objectif est de suivre en ligne les évolutions du procédé surveillé, d'identifier des situations anormales et d'aider l'opérateur dans les prises de décisions, en particulier lorsque les situations anormales ont pour origine un dysfonctionnement de la partie opérative. Ce travail apporte une contribution aux méthodes de diagnostic basées sur l'utilisation de modèles de bon fonctionnement. Nous y développons une approche à base de modèle, qui s'effectue en ligne.

Par ailleurs, 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. 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 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 ; Sargousse, 1999 ; Moyses, 2000 ; Perret, 2003].

Ce travail s'inscrit dans le domaine du diagnostic à base de modèle. Il contribue à enrichir tant les méthodes pour l'isolation de défauts que les méthodes de génération de résidus. L'objectif de cette thèse est la prise en compte commune des aspects dynamiques continus et discrets des systèmes étudiés, par la simulation en parallèle du modèle de référence. La mise en place d'un filtre de Kalman étendu dans le cas non linéaire est proposée afin d'estimer en temps réel l'état continu et discret du système et permettre ainsi la surveillance de ce dernier. La méthodologie proposée peut alors tenir compte de la validité du modèle de référence ainsi que des incertitudes de mesures. Ceci amène au développement d'une méthode de génération de résidus pour détecter les changements de mode de fonctionnement non souhaités, au cours d'une opération ou lors des modes transitoires. Le second intérêt de ces travaux se situe sur la résolution du problème de diagnostic par l'utilisation de distances particulières.

Ce manuscrit s'articule autour de quatre parties :

- La première partie a pour objectif de préciser la problématique de la surveillance des systèmes dynamiques hybrides, plus particulièrement la surveillance des procédés batch. Dans ce cadre, deux chapitres sont développés.
 - Le premier fixe le cadre de l'étude et définit le domaine d'application, le génie des procédés. Nous exposerons la problématique de la surveillance et soulignerons les contraintes et principes auxquels doit répondre le module de surveillance.
 - Le deuxième chapitre est consacré à la présentation de la plate-forme de *simulation dynamique hybride*, nommée *PrODHyS*, développée au sein du *Laboratoire de Génie Chimique*.
- Le deuxième volet dresse un état des lieux des méthodes existantes dans le domaine de la surveillance. Celui-ci se décompose en deux chapitres :
 - Dans le troisième chapitre, la terminologie employée dans ce mémoire est présentée afin de bien positionner le cadre de l'étude. Ensuite l'accent est mis sur les différentes approches existantes pour la détection et le diagnostic de fautes. L'analyse de ces méthodes est résumée dans un tableau qui offre une vision globale et synthétique de l'existant. Enfin, la théorie *Fault Detection and Isolation* sur laquelle repose nos travaux est abordée.
 - Le quatrième chapitre résume les résultats connus sur les observateurs, utilisés particulièrement dans nos travaux, et leurs exploitations pour des études de détection et diagnostic de fautes.
- La troisième partie est consacrée au développement de l'approche de surveillance proposée dans ce travail. Deux chapitres sont dédiés à cette définition :
 - Le cinquième chapitre est consacré au développement d'une stratégie originale et générale de détection et diagnostic de fautes qui nous permettra de répondre correctement au cahier des charges précédemment établi. Il s'agit non seulement de détecter des défaillances mais aussi d'en trouver leur origine.

- Le sixième chapitre est consacré à l'intégration dans la plate-forme *PrODHyS* de la méthodologie de surveillance proposée. Dans ce cadre, les éléments constitutifs et la sémantique de l'approche développée sont décrits et modélisés.
- Enfin, la quatrième et dernière partie du manuscrit s'intéresse à l'exploitation de cette approche pour la surveillance de procédés.
 - Le premier volet de cette partie est consacré à la mise en place du système de surveillance d'un procédé. La démarche à suivre est soulignée ;
 - Le second a pour objet de montrer comment un utilisateur peut surveiller un procédé complet. Dans ce cadre, trois exemples sont présentés : le premier consiste en la surveillance d'une opération de changement de solvant (*détection d'un défaut thermique*) ; le second concerne un benchmark de la littérature (*détection de défauts multiples*) et le dernier aborde la surveillance d'un procédé complexe (*caractérisation du degré de criticité du défaut*).

PARTIE 1 :

LE CONTEXTE DE L'ÉTUDE

CHAPITRE 1 LE CADRE DE L'ÉTUDE

Ce chapitre introductif décrit le contexte dans lequel s'inscrit l'approche de diagnostic proposée. Après une brève définition des systèmes dynamiques hybrides, nous présentons les systèmes industriels plus particulièrement visés : les procédés discontinus. Ensuite, nous donnons quelques aspects fondamentaux d'un système de supervision. Enfin, la liste des principaux verrous est établie. Elle met alors en évidence la nécessité d'une supervision avancée.

De nos jours, le monde industriel est soumis à la concurrence du marché. Afin d'assurer la pérennité de leurs activités, les entreprises doivent faire face à de nombreux enjeux socio-économiques. Leurs systèmes de production sont constamment complexifiés pour atteindre les objectifs fixés. Leur capacité à réagir, à respecter les délais et à garantir une certaine productivité est devenue des facteurs concurrentiels interentreprises.

Par ailleurs, dans des domaines tels que les procédés chimiques, il est quasi impossible de suspendre une opération en raison de la présence d'une matière que nous pouvons qualifier de vivante. Celle-ci est en constante évolution et peut donc échapper à tout contrôle (par exemple, une réaction chimique peut s'emballer). Une simple défaillance est alors jugée néfaste. La conduite optimale du procédé doit donc être assurée en permanence. On comprend alors l'intérêt d'une bonne surveillance pour les entreprises.

Ce chapitre introductif vise à positionner la problématique de la supervision des systèmes dynamiques hybrides. Pour cela, nous présentons, dans un premier temps, les procédés discontinus qui sont plus particulièrement étudiés dans ces travaux. La modélisation de ce type de système passe par la notion de système dynamique hybride. Enfin, après avoir présenté les objectifs de la supervision et des fonctions qui la composent, l'intérêt d'une supervision « avancée » est souligné.

1.1 LES SYSTÈMES INDUSTRIELS CONSIDÉRÉS : LES PROCÉDÉS

Le Génie des procédés consiste à étudier, concevoir et mettre en œuvre de façon optimale les procédés, systèmes industriels complexes de transformation de la matière et de l'énergie. Cette discipline a ainsi apporté des concepts novateurs dans des domaines d'application très variés :

- la chimie lourde (le pétrole, le charbon, l'électrochimie industrielle, la métallurgie, la pétrochimie, le nucléaire, le ciment, le papier, le verre, le plastique, le textile, etc.),
- la chimie fine (la pharmacie, la cosmétologie, la photographie, etc.),
- la biochimie (l'agroalimentaire, la bio-industrie, etc.).

1.1.1 La notion d'opérations unitaires

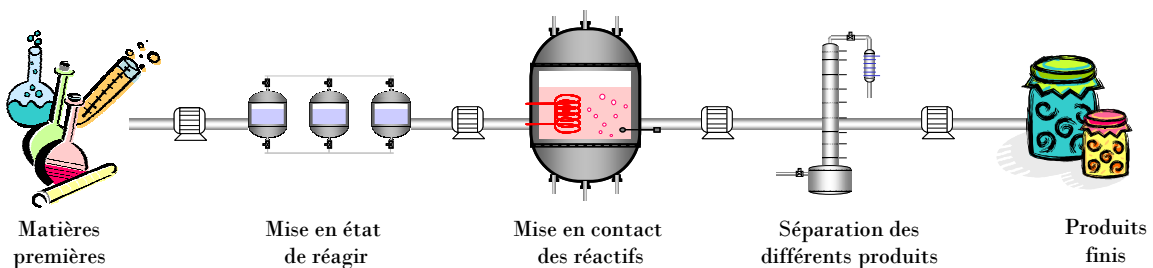


Figure 1.1 Les étapes de fabrication d'un produit chimique

Les appareillages nécessaires pour l'implémentation industrielle des procédés sont très variés. Néanmoins, ils font appel à un nombre restreint de techniques appelées *opérations unitaires*. Ainsi, ces appareillages se regroupent selon des principes communs.

Un procédé complet apparaît alors comme une succession d'opérations unitaires mises en œuvre pour réaliser la synthèse d'un produit chimique. De manière générale, la fabrication d'un produit ou d'un intermédiaire se déroule en quatre étapes de base (cf. figure 1.1).

Le stockage et le transfert des matières premières

Les matières premières sont stockées dans des containers appropriés puis acheminées vers le réacteur par diverses technologies, telles que la différence de pression, la gravité, des pompes ou encore des moyens mécaniques. De plus, il faut noter que les politiques de stockage des produits ont souvent un impact important dans la gestion de la production, notamment dans le cas des procédés batch.

La préparation des matières premières

Les opérations de broyage, de mélange, de chauffe, de dilution, de dissolution, etc. amènent les matières premières dans un état leur permettant de réagir. Des matières solides doivent par exemple être soumises à des opérations de broyage afin d'atteindre le degré de finesse souhaité ; elles peuvent également être mises en suspension dans un solvant ou un diluant.

La mise en contact des réactifs dans des conditions opératoires appropriées

Les opérations de réaction, d'agitation, de chauffe, de refroidissement, de fermentation, de pasteurisation, etc. sont nécessaires pour la mise en contact des réactifs. Cette mise en contact a lieu généralement au sein d'un réacteur. Ce dernier doit par ailleurs être maintenu manuellement ou automatiquement dans des conditions opératoires adaptées (température, pression, pH, etc.). Il est possible d'ajouter et de soutirer des réactifs au cours de la réaction, par exemple, par distillation.

La séparation des produits résultant de la réaction

Après la réaction, la masse réactionnelle ne contient pas seulement le produit recherché, mais aussi les produits secondaires qui ont pu se former, les matières premières non transformées, des impuretés, les solvants ou diluants utilisés, etc. Une étape de séparation de ces produits est alors nécessaire. Cette étape est souvent résolue par combinaison de plusieurs techniques séparatives telles que les opérations de décantation, de filtration, de sédimentation, de centrifugation, de cristallisation, de distillation, de rectification, d'absorption, d'adsorption, d'extraction par solvant, de séchage, de tamisage, etc.

Lorsque le produit final nécessite la synthèse de produits intermédiaires, ces étapes de base (« mise en état de réagir », « mise en contact des réactifs » et « séparation des différents produits ») sont répétées autant de fois que nécessaire.

1.1.2 Les différents flux

Un procédé est généralement traversé par un ensemble de flux. Parmi ceux-ci, on distingue : les flux physiques qui permettent un transport de matière ou d'énergie et les flux d'information.

Les flux physiques

Les *flux d'énergie* peuvent être apportés au procédé sous forme thermique, électrique ou mécanique ; ils peuvent également résulter des transformations mises en jeu au sein du procédé.

Au cours des différentes étapes de fabrication du produit chimique, les *flux de matière* transitent au travers des appareils et subissent des transformations physiques et chimiques. Il faut noter que la nature de ces flux de matière caractérise les deux principales catégories de procédé :

- Lorsqu'un flux continu transite entre les appareils, le procédé est dit *continu* ;
- Si au contraire, la matière transite par lots, on parle alors de procédés *discontinus*. Cette dernière catégorie de procédé fait l'objet d'une étude plus spécifique dans ces travaux.

Les flux d'information

Les *flux d'information* permettent un échange d'information entre les différents appareils du procédé. Ce type de flux est largement utilisé au niveau du contrôle et de la régulation du procédé (par exemple : capteurs, détecteurs, systèmes numériques de contrôle/commande, etc.).

1.1.3 Le mode de production par lots (ou discontinu)

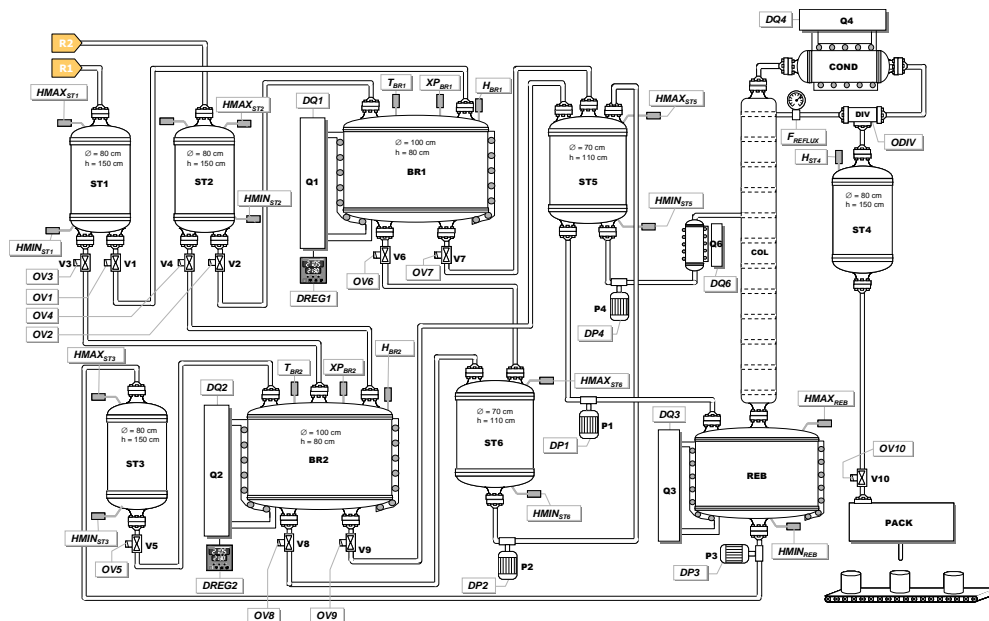


Figure 1.2 Une variante d'un procédé discontinu décrit dans [Joglekar et al., 1985]

Ces dernières années, le contexte économique, dans lequel s'inscrit le domaine du génie des procédés, se montre de plus en plus concurrentiel. En effet, le marché mondial doit répondre à une clientèle

de plus en plus exigeante en termes de qualité et de spécificité des produits. Les unités de production doivent fabriquer en petites quantités des produits à haute valeur ajoutée. Les modes de production qui en résultent sont alors souvent de type batch ou semi-batch (cf. figure 1.2).

De plus, les délais de production doivent être minimales afin de satisfaire le client dans les plus brefs délais. Dans ce contexte, la production doit être flexible, dynamique et réactive. Ce mode de production par lot concerne, par exemple, les entreprises du secteur agroalimentaire, pharmaceutique ou cosmétologique.

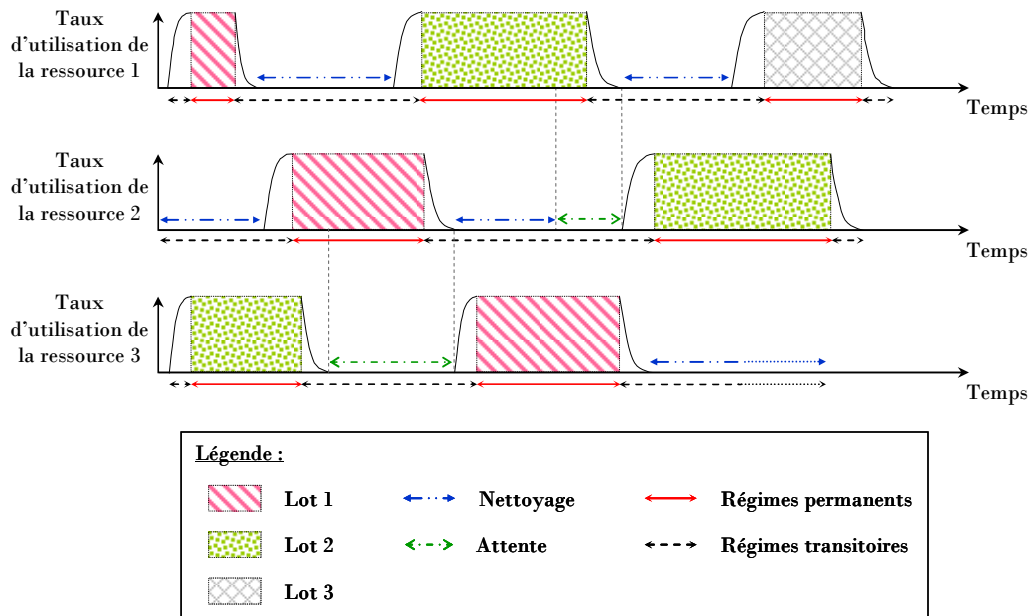


Figure 1.3 L'utilisation des ressources en production discontinue
(alternance de régimes permanents et transitoires)

Un procédé fonctionnant selon un mode de production discontinu se caractérise donc par :

- une production organisée en lots de produits diversifiés se partageant les mêmes ressources ;
- des opérations unitaires réalisées successivement sur un lot de produit ;
- des flux de produit variables et discontinus ;
- de nombreuses phases de régime transitoire dont les durées sont non négligeables (cf. figure 1.) ;
- la nécessité d'introduire de nombreuses cuves tampons.

1.1.4 La notion de recette

L'ensemble des informations nécessaires à l'élaboration d'un produit particulier est regroupé au sein d'une structure de données appelée *recette*.

Le contenu de la recette est exploité à différents niveaux décisionnels avec une granularité adaptée (recette agrégée à un niveau tactique et recette très détaillée à un niveau supervision). Dans le cas de procédés constitués d'un grand nombre d'appareils ou de procédures de fabrication élaborées, la recette peut rapidement devenir très complexe. Afin d'établir une approche standard pour traiter la complexité du

contrôle/commande des procédés batch, la norme *ISA/SP88* (www.isa.org) propose une modélisation hiérarchisée de la recette.

Elle est constituée de cinq éléments d'information :

- l'*en-tête* qui regroupe les informations administratives (identifiant, auteur, etc.),
- la *formule* qui indique la liste des matières premières et des produits intermédiaires nécessaires ainsi que leur proportion (en pourcentage) et les conditions opératoires à appliquer,
- les *besoins en équipement*, ou type d'équipement,
- la *procédure* définissant la suite d'opérations unitaires ordonnées dans le temps qui décrit la fabrication du produit,
- des informations diverses liées aux contraintes de qualité et de sécurité.

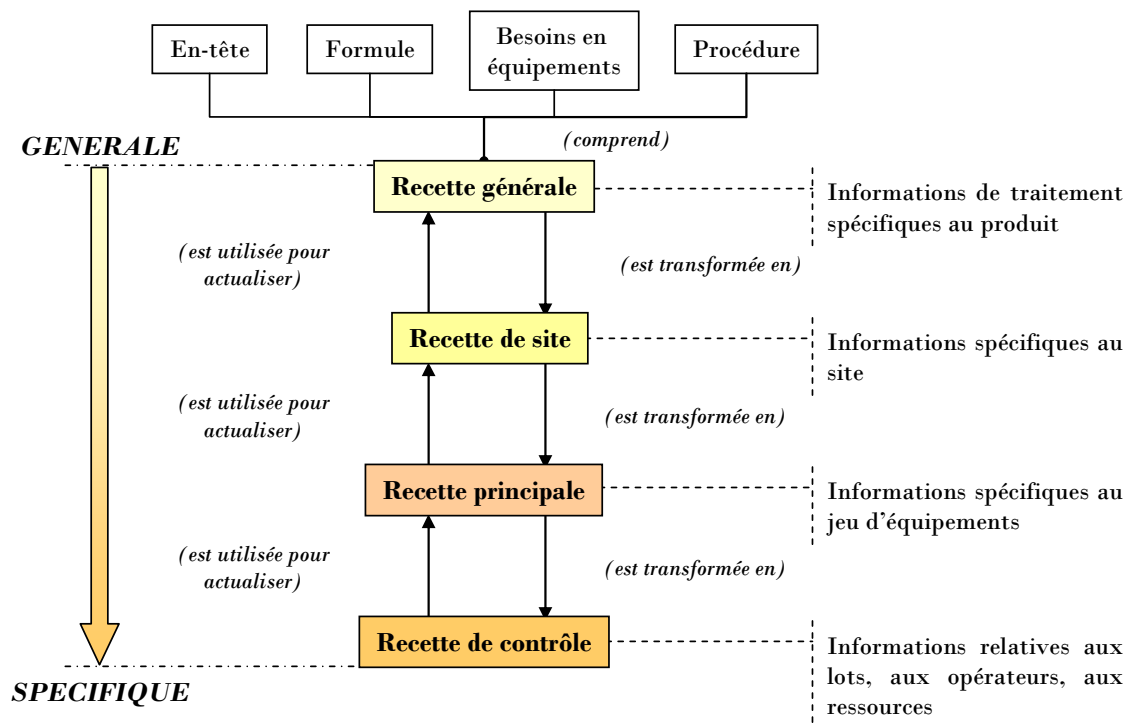


Figure 1.4 Les types de recette et leurs contenus

Comme l'illustre la figure 1.4, selon le niveau de décision, quatre types de recette sont définis :

- la *recette générique* (ou générale) spécifie la méthode d'obtention du produit indépendamment des équipements utilisés ou d'une campagne particulière. Il s'agit généralement de la recette fournie directement par le service « industrialisation » ;
- la *recette de site* complète la recette générale en précisant les caractéristiques des équipements et des ingrédients requis compte tenu des ressources disponibles sur le site de production. Cette recette est utilisée généralement pour la planification ;

- la *recette principale* est une recette de site adaptée à une campagne particulière (taille des lots définis), l'équipement est décrit par classe (caractéristiques telle que la capacité) mais aucun appareil particulier n'est encore désigné. Ce niveau de recette est mis en œuvre pour la fonction d'ordonnancement.
- enfin, la *recette lot* (ou de contrôle) est une recette appliquée à un lot particulier auquel elle est associée pendant tout son processus de fabrication. Elle correspond à l'exécution en temps - réel de la recette principale avec une affectation effective des équipements et est utilisée au niveau de la supervision.

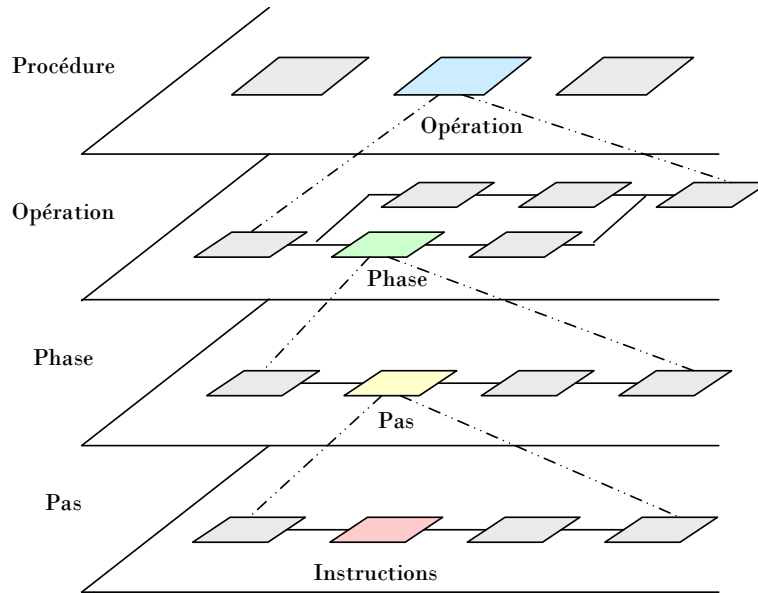


Figure 1.5 La hiérarchisation de la procédure

La partie procédurale est l'élément clé de la recette. À ce niveau, la norme *ISA SP88* préconise aussi une hiérarchisation des informations selon cinq niveaux (cf. figure 1.5) :

- le *niveau procédure* définit un ensemble d'opérations à exécuter dans une ligne de production pour fabriquer un produit particulier.
- Le *niveau opération* définit l'ensemble des phases exécutées en séquence ou en parallèle dans une unité de production. Ce niveau correspond à la notion d'opération unitaire induisant une modification des propriétés physico-chimiques d'un produit (réaction, séparation, décantation, chauffe etc.).
- Le *niveau phase* définit une succession de pas permettant de réaliser une fonction élémentaire. Une phase est limitée par des frontières définies par les instants où le déroulement des pas peut être interrompu en toute sécurité pour l'installation et sans dommage pour le produit.
- Le *niveau pas* désigne une séquence temporelle ou événementielle d'instructions à exécuter dans un module d'équipement en vue d'accomplir une tâche spécifique
- Enfin le *niveau instruction* définit une commande élémentaire orientée équipement adressée à un élément ou à un composant afin qu'il effectue un traitement donné.

1.2 LES SYSTÈMES DYNAMIQUES HYBRIDES

Les méthodes de l'automatique n'ont longtemps reposé que sur deux types de technologies, fondées sur des méthodes qui leurs étaient propres et appliquées à des domaines différents : les systèmes continus et les systèmes évènementiels. Les systèmes continus font appel pour leur description à des outils mathématiques tels que les équations différentielles, tandis que les systèmes à évènements discrets sont classiquement représentés par l'algèbre de Boole combinée à des formalismes états – transitions ayant une représentation graphique (Grafcet, réseau de Petri, automates à états finis, etc.).

Néanmoins, 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, démontrant ainsi que la frontière entre systèmes continus et systèmes à évènements discrets n'était pas si étanche. Il apparaît alors évident que ces systèmes ne peuvent se contenter d'une représentation homogène à dynamique purement continue ou purement évènementielle. Depuis les années quatre-vingt-dix, une attention particulière de la part de la communauté scientifique s'est portée sur l'étude de ces systèmes dits « hybrides » [Alur *et al.*, 1995 ; Branicky, 1995 ; Engell, 1997 ; Guéguen et Lefebvre, 2001] et de nombreuses approches de modélisation traitant à la fois les aspects continus et discrets ont été proposées.

1.2.1 La définition générique

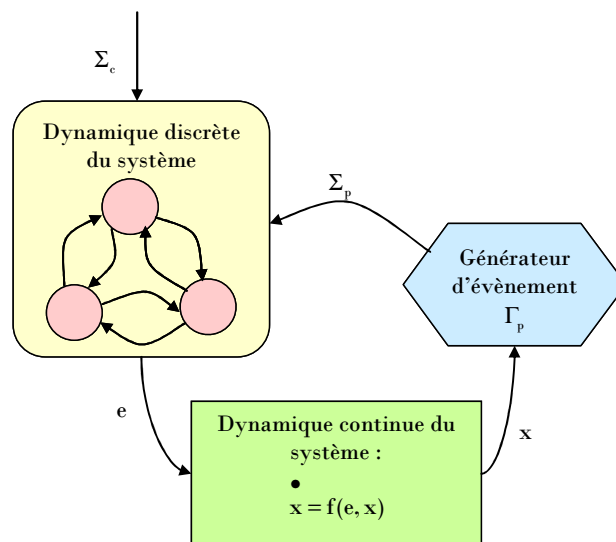


Figure 1.6 La modélisation d'un système dynamique hybride [Tittus, 1995]

Les systèmes dynamiques hybrides couvrent de nombreux domaines allant des technologies de l'information et de la communication aux procédés chimiques. Plus particulièrement, les procédés batch sont intrinsèquement hybrides, puisqu'ils rassemblent de nombreuses unités de traitement continu interconnectées et partagées. Par ailleurs, avec un mode de production par lot, la procédure associée à la recette comporte une forte composante évènementielle mais dépend aussi de paramètres à valeurs continues (conditions opératoires, température, pression, etc.).

De manière plus générale, les systèmes dynamiques hybrides caractérisent les systèmes faisant intervenir explicitement et simultanément des phénomènes de types continu et événementiel. Pour leur description, ils nécessitent l'utilisation de fonctions du temps continues par morceaux et de fonctions à valeurs discrètes. Ces différentes fonctions représentent aussi bien l'état interne que les entrées, les sorties et les perturbations.

Les systèmes dynamiques hybrides sont caractérisés par des dynamiques continues propres à chaque état discret composant le système, de telle sorte que l'occurrence d'un événement discret provoque un changement d'état discret e mais également une modification de la dynamique continue du système modélisé par le système différentiel $\dot{x} = f(e, x)$. La figure 1.6 illustre une modélisation d'un système dynamique hybride d'après [Tittus, 1995]. La détermination de l'état discret e du processus est réalisée en tenant compte des événements de contrôle Σ_c du processus et des événements physiques Σ_p du processus issus du générateur d'événement Γ_p .

1.2.2 La modélisation des systèmes dynamiques hybrides

Le modèle (cf. le point 2.1.2.1) est la traduction du comportement dynamique du système physique par un ensemble de propriétés qui en facilitent la description et la compréhension. C'est une étape primordiale à toute étude dont l'approche expérimentale apparaît difficile à mettre en place. Cette étape doit répondre à des contraintes en termes de qualité (proche de la réalité), de lisibilité et de réutilisabilité.

Le comportement des systèmes dynamiques est traditionnellement décrit par un modèle, dont la représentation, continue ou discrète, est directement liée à la nature des variables d'état et temporelles qui le caractérisent [Zaytoon, 2001].

Les variables d'état d'un système sont :

- soit *continues* : la variable prend alors ses valeurs dans l'ensemble des réels (température, concentration),
- soit *discrètes* : la variable prend ses valeurs dans un ensemble dénombrable, sous-ensemble des entiers naturels (nombre de pièces),
- soit *symboliques* : la variable prend ses valeurs dans un ensemble fini non structuré (par exemple la vanne TOR est caractérisée par ses états *ouvert* et *fermé*).

La modélisation des systèmes dynamiques hybrides doit donc 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. Il est évidemment impossible de passer en revue la totalité des approches proposées. Par conséquent, nous invitons le lecteur à lire les articles référencés pour plus d'informations. Selon l'approche de modélisation adoptée, une classification a été mise en place [Chombart *et al.*, 1996 ; Flaus, 1998 ; Zaytoon, 2001] :

- les approches qui peuvent être vues comme une extension des modèles des systèmes à dynamique continue pour intégrer le comportement discret (cas des bond-graphs [Buisson et Cormerais, 1998] à

commutation par exemple). Ces approches consistent à introduire au sein du modèle continu des variables booléennes ou entières (par exemple VRAI/FAUX, 0/1).

- les approches qui peuvent être vues comme une extension des approches discrètes classiques pour inclure le comportement continu, telles que les réseaux de Petri hybrides [Alla *et al.*, 1992], les réseaux de Petri temporisés, temporels et stochastiques [Sifakis, 1977 ; Merlin et Segall, 1976 ; Florin *et al.*, 1991], les automates temporisés [Alur et Dill, 1994]. Ces formalismes tiennent compte des aspects quantitatifs du temps en ajoutant par exemple des contraintes temporelles.
- les approches mixtes qui combinent les modèles du discret et du continu dans une même représentation (par exemple, les automates hybrides [Alur *et al.*, 1995], les statecharts hybrides [Kesten et Pnueli, 1992], les réseaux de Petri prédicats-transitions différentiels [Champagnat *et al.*, 1998]). Chacune des deux parties (continue et discrète) est représentée de façon rigoureuse et explicite et leur collaboration est traitée au sein de l'interface qui les relie. Notons que la structuration se fait généralement à partir du discret.

1.3 LA CONDUITE DE PROCÉDÉ AU SEIN DU SYSTÈME DE PILOTAGE

Quelle que soit la branche d'activité concernée, les procédés sont, de nos jours, pilotés grâce à de nombreux systèmes informatiques. La tâche de ces systèmes va de la gestion prévisionnelle des Ordres de Fabrication jusqu'à la conduite en temps – réel du procédé. L'objectif de ce chapitre est de montrer la place de la conduite au sein du système de pilotage et l'importance de la supervision pour la bonne marche de la production.

1.3.1 La place de la conduite de procédé dans le modèle CIM

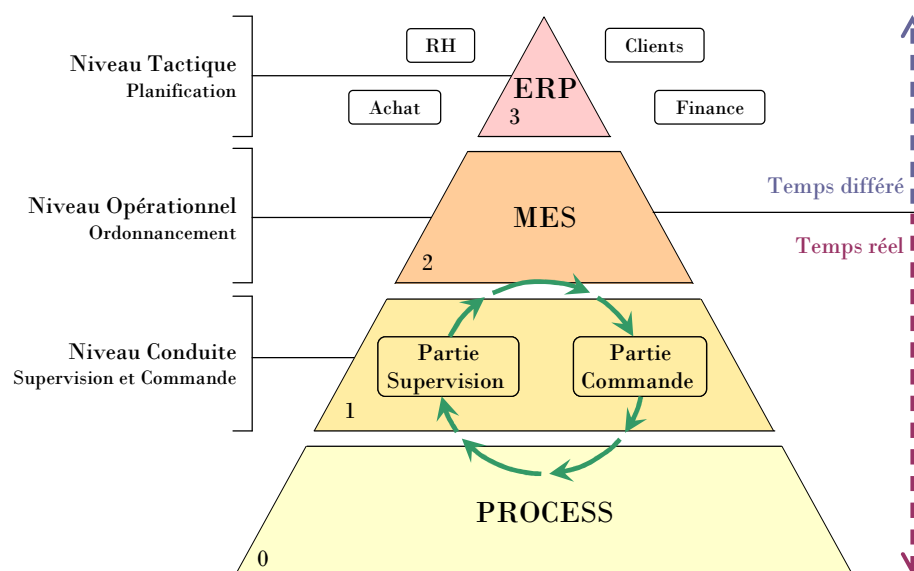


Figure 1.7 La pyramide du modèle CIM

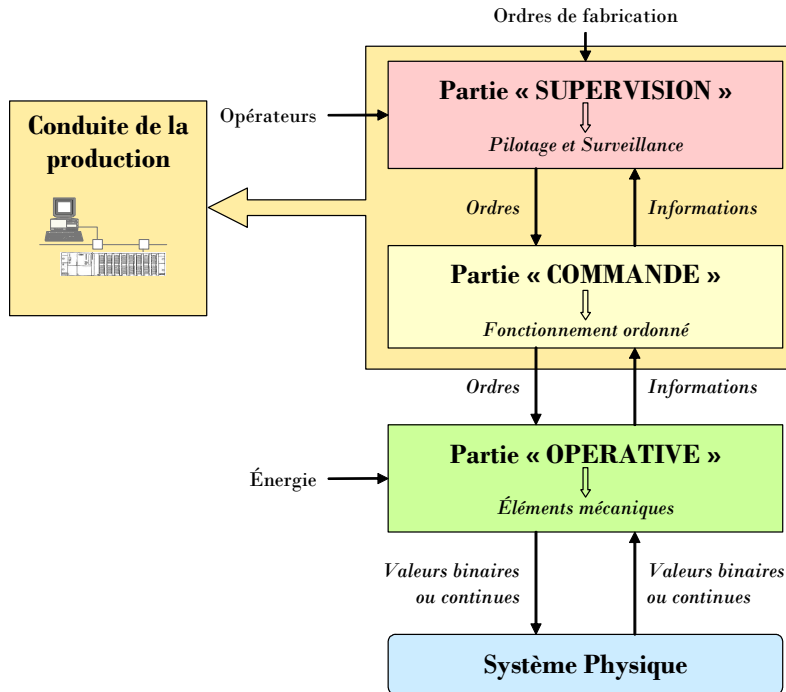


Figure 1.8 La structure logique du niveau conduite

Durant les années 80, le manque de cohérence dans la chaîne d'information de l'entreprise a amené une prise de conscience du besoin important d'intégration des systèmes informatiques tout au long de la chaîne de production. Le modèle *Computer Integrated Manufacturing* (CIM), défini par un consortium d'industriels, décrit la mise en œuvre des moyens informatiques et automatiques concourant à assurer simultanément la réactivité, la qualité et la rentabilité de tout ou partie d'un ensemble industriel.

Il intègre les fonctions de CAO (Conception Assistée par Ordinateur), d'ateliers flexibles, de centres d'usinage à commande numérique (CNC), d'ERP (Enterprise Resources Planning) de stockage, de MES (Manufacturing Execution System), de manutention automatisée ainsi que les méthodologies conceptuelles d'intégration de ces composants à l'intérieur d'un système global d'information de l'entreprise.

La représentation la plus courante fait apparaître un découpage structurel de l'organisation de l'entreprise, et une communication spécifique entre les niveaux du modèle (cf. figure 1.7). Il comporte quatre niveaux correspondant à des niveaux de décision.

Plus on s'élève dans cette pyramide, plus le niveau de décision/d'abstraction est important, plus la visibilité est globale et plus les cycles standards s'allongent. Un niveau supérieur décide ce qu'un niveau inférieur exécute.

- Le niveau 3 correspond à la gestion prévisionnelle de la production ; c'est-à-dire à la gestion des produits et des stocks, à la gestion des approvisionnements, et à la gestion des clients, des commandes et de la facturation.
- Le niveau 2 est à l'interface entre le temps différé et le temps réel. Il réalise la localisation des produits en stocks, la détermination des mouvements physiques et la gestion des lots et assure la fonction de traçabilité.

- Le niveau 1 correspond aux équipements assurant la commande (automates programmables, régulateurs numériques,...) et la surveillance du procédé (console de contrôle).
- Le niveau 0 est constitué des capteurs et actionneurs du process.

Le couplage de ces différents moyens assure le pilotage du système industriel.

Le niveau conduite (cf. figure 1.8) se décompose lui-même en deux sous – niveaux : le contrôle / commande et la supervision.

La partie « commande » rassemble les moyens informatiques permettant la gestion autonome d'un cycle de travail préétabli, alors que la partie « supervision » doit mettre en œuvre les ordres provenant du niveau ordonnancement et assurer la surveillance du procédé.

1.3.2 La supervision

La supervision d'un procédé est à l'interface entre la gestion hors ligne et les systèmes de commande. Elle consiste en la conduite d'installations industrielles tant en fonctionnement normal qu'en présence de défaillances. Elle permet de suivre en temps réel des informations disponibles et ainsi de réagir face aux variations observées pour maintenir le procédé dans un mode de fonctionnement optimal.

La supervision englobe donc les tâches de *pilotage* et de *surveillance* en temps réel d'un procédé complexe, selon un certain degré d'abstraction [Dubois et Gentil, 1990].

1.3.2.1 Le pilotage

Le *pilotage en temps réel* consiste à exécuter la séquence opératoire de chaque recette et assure en temps réel la gestion des ressources nécessaires à cette exécution. Pour cela, il met en œuvre en temps réel les ordres de fabrication (consigne) provenant de la fonction *ordonnancement* en traduisant cette information sous forme de tâche de commande et en la distribuant aux automates et contrôleurs locaux. Cette fonction est donc située au-dessus des systèmes de commande, et est chargée :

- de gérer (lancer, arrêter, suspendre, etc.) des lois de commande prédéfinies (programmes automate, paramètres de régulation, etc.),
- de gérer le transitoire correspondant aux changements de régime du système (démarrage, changement de points de fonctionnement, arrêt).

De plus, la commande doit s'adapter à une déviation observée. Cela va de la modification du plan de fabrication par interaction avec le module d'ordonnancement, à la modification de conditions opératoires par interaction avec les niveaux de commande inférieurs.

Cette fonction induit souvent une volonté d'autonomie vis à vis des opérateurs (source de reproductibilité) et une réactivité relative au mode de fonctionnement. C'est une fonction très automatisée. Les objectifs du pilotage résident essentiellement dans le respect des critères de qualité de productivité et de la tenue des délais.

1.3.2.2 La surveillance

La fonction de *surveillance* complète alors parfaitement la tâche *pilotage* [Combacau et Courvoisier, 1990]. Adapter la commande suppose en effet :

- de constater tout d'abord les déviations (ce qui est assuré par la fonction de *détection*),
- d'en trouver la cause (ce qui est géré par la fonction *diagnostic*),
- et de déterminer la réponse adéquate (ce qui est établi par la fonction *reprise*).

La surveillance veille au bon déroulement de la fabrication. Dans ce but, elle vérifie de manière récurrente la conformité de l'évolution dynamique des opérations effectuées sur le système de production en se référant à une situation de fonctionnement normal que l'on cherche à maintenir pour assurer les conditions nominales de fabrication.

1.4 VERS UN SYSTÈME DE SURVEILLANCE AVANCÉ

Les procédés industriels sont devenus de plus en plus complexes. Ceci est dû non seulement, à la complexité des phénomènes physico-chimiques qui les gouvernent, mais aussi au nombre continuellement croissant de variables nécessaires pour garantir leur bon fonctionnement. De nos jours, les procédés sont de plus en plus instrumentés (capteurs, régulateurs, actionneurs) et sont couplés à un ou plusieurs calculateurs numériques, destinés à l'acquisition de données et à la mise en œuvre de l'automatisation. Les objectifs premiers de cette automatisation accrue ont d'abord été :

- l'augmentation des performances du système de production,
- la garantie de la qualité du produit fabriqué qui doit satisfaire un certain nombre de normes,
- la diminution des coûts de fabrication,

Par ailleurs, il est fréquent aujourd'hui que la supervision soit aussi chargée :

- de l'amélioration de la sécurité de l'installation industrielle, des hommes qui y travaillent et de son environnement,
- de la traçabilité du produit.

Face à ces exigences et au flux d'informations dynamiques et variables, l'opérateur humain a besoin de nouveaux outils d'aide à la décision afin d'assurer de manière réactive et sûre la conduite du procédé. C'est dans ce contexte que sont nés les M.E.S. dans lesquels la fonction de surveillance tient une place importante.

1.4.1 Les enjeux de la surveillance

1.4.1.1 Le rôle de l'opérateur au sein de la supervision

Malgré une automatisation accrue, l'opérateur se positionne encore comme le maillon « intelligent » de la boucle de surveillance : il a en charge d'analyser la situation et de prendre la décision

adéquate. De plus, suite à une défaillance ou une dérive, les moyens de réaction sont souvent manuels ou semi-automatiques.

Sa tâche d'analyse est facilitée par la mise en place d'interfaces ergonomiques donnant différentes vues du procédé sous forme de synoptiques. Ceux-ci reproduisent l'installation et affichent en temps réel les grandeurs mesurées. La figure 1.9 montre quelques exemples de synoptiques proposés sur des systèmes de supervision commerciaux :

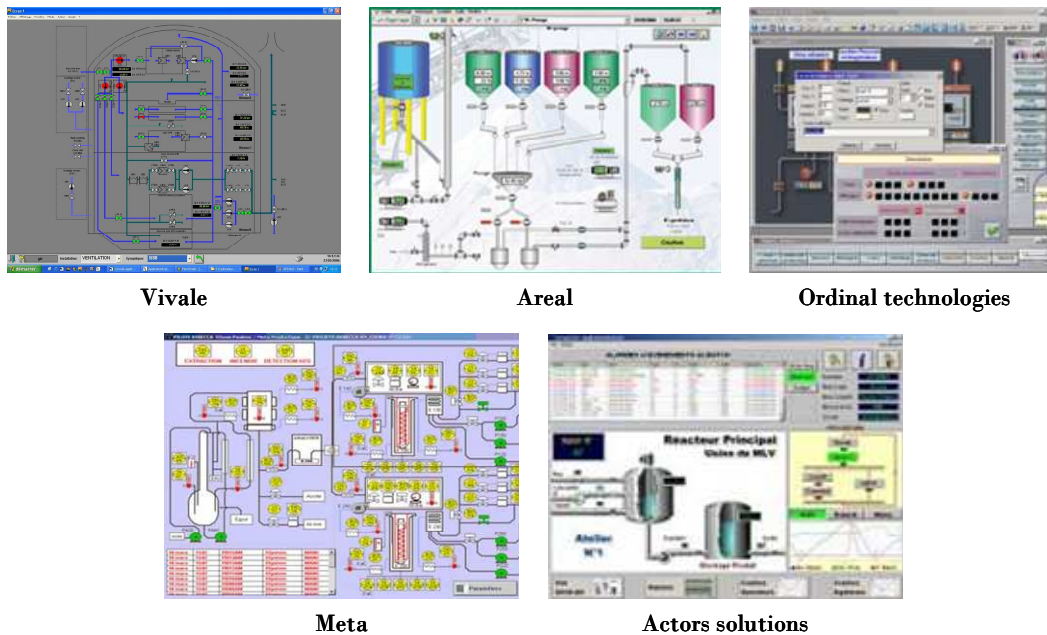


Figure 1.9 Des synoptiques de systèmes de supervision commerciaux

Généralement, ces systèmes permettent aussi de stocker des historiques de certaines variables, afin d'assurer la traçabilité des processus. Naturellement, l'opérateur a aussi la possibilité de commander directement certains actionneurs, ainsi que de modifier certains paramètres de régulateurs s'il constate une dérive ou un effet de pompage (dû(e) à une régulation en limite de stabilité par exemple).

Toutefois, cette tâche est de plus en plus difficile pour l'opérateur du fait de la taille et la complexité des installations modernes. Dans ces conditions, l'opérateur peut prendre de mauvaises décisions et commettre alors une erreur irréparable. Selon [Venkatasubramanian *et al.*, 2003], 70% des accidents industriels sont dus à une erreur humaine. Ces accidents ont un impact financier, sécuritaire et environnemental.

1.4.1.2 L'impact financier

Des études ont montré que des incidents mineurs sont très fréquents, quotidiens même, et occasionnent des pertes économiques importantes. Ainsi, l'industrie pétrochimique aux États-Unis a des pertes annuelles de 20 milliards de dollars [Nimmo, 1995]. Un autre exemple concerne l'économie anglaise : les pertes pour l'industrie pharmaceutique et la chimie fine sont de 27 milliards de dollars chaque année [Laser, 2000].

1.4.1.3 L'impact sécuritaire et environnemental des accidents

À ces incidents quotidiens, s'ajoutent des accidents chimiques, catastrophes économiques, humaines et environnementales. Le plus notable reste la fuite de quarante tonnes de gaz toxiques de l'usine de pesticides d'Union Carbide à Bhopal en Inde faisant plus de 8 000 morts dans les trois premiers jours et plus de 20 000 en près de vingt ans.

Deux ans plus tard, la pollution chimique de Schweizerhalle touche gravement les écosystèmes et entraîne l'interdiction de pêche pendant six mois. Nous pouvons rajouter à cette liste l'accident de la plateforme pétrolière Piper-Alpha en 1988, qui cause la mort de 167 personnes et près de 3,5 milliards de dollars de dommages, celui de l'usine de pesticides Anaversa de Córdoba, répandant 38 000 litres de pesticides dans l'atmosphère, ou encore, en 2005, celui de l'usine pétrochimique de Jilin déversant 100 tonnes de benzène, produit hautement toxique, dans la rivière Songhua.

En 2001, l'explosion de l'usine chimique AZF à Toulouse, avec 30 morts et plus de 2500 blessés graves tout en causant des destructions importantes, souligne la nécessité d'une surveillance avancée dans les systèmes complexes. De ce fait, automatiser cette fonction peut améliorer les performances du système physique et la sécurité du dispositif industriel et de son environnement.

1.4.2 Les limites des méthodes classiques de surveillance

1.4.2.1 L'utilisation de la redondance

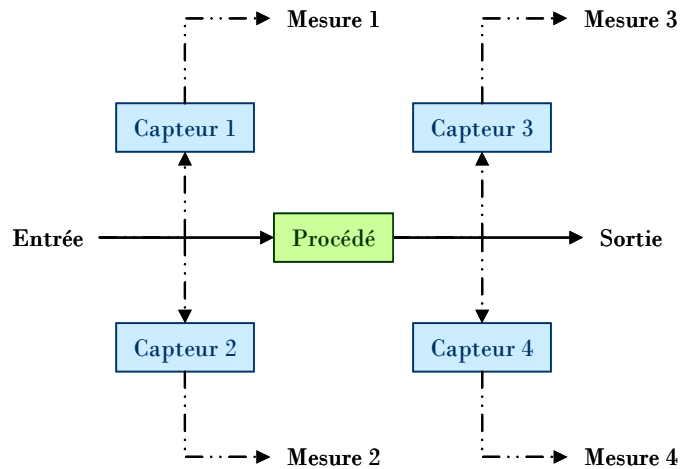


Figure 1.10 La redondance matérielle

Afin d'améliorer la fiabilité d'une installation, une première solution est d'utiliser la technique de redondance (cf. figure 1.10). Ici, l'objectif n'est ni de détecter ni de localiser un défaut, mais plutôt de le masquer. Le principe de redondance est mis en œuvre en dupliquant certains composants matériels du système tels que les actionneurs, telles les pompes et les vannes, ou l'instrumentation, tels les capteurs. On parle alors de *redondance matérielle*. Dans le cas des actionneurs, des paires sont constituées de façon à ce que le second puisse pallier une défaillance du premier (cf. figure 1.11.a). Dans le cas des capteurs, on peut aussi les doubler. Dans ce cas, si les deux capteurs émettent des signaux différents, on conclut qu'une

défaillance d'au moins un des capteurs existe [Zhang, 1999]. Par contre un nombre impair, généralement trois, est requis si l'on veut rendre le système décidable (arbitrage à la majorité) (cf. figure 1.11.b).

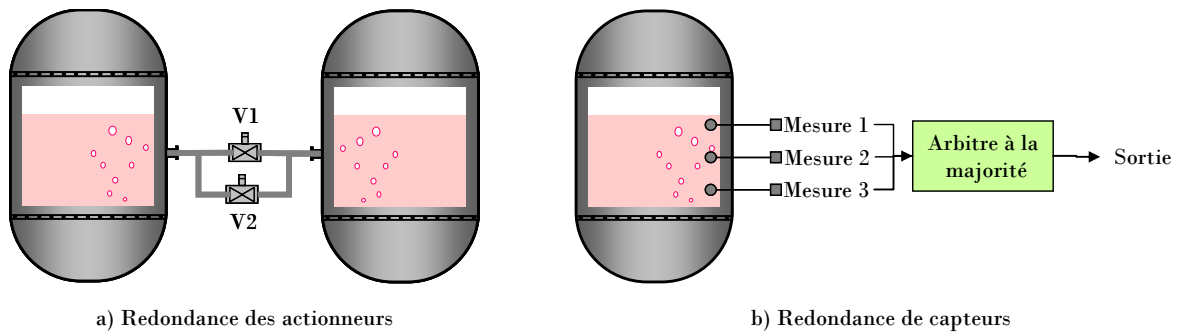


Figure 1.11 Des exemples de redondance matérielle

Ce même principe est aussi utilisé pour les systèmes de commande, autant pour la partie matérielle (calculateur) que pour la partie logicielle (programme). Dans ce cas, afin de palier les défauts de programme (bug), un code développé par trois éditeurs différents sont implantés sur chaque calculateur. On parle alors de *redondance logicielle*.

Cette solution a l'avantage d'être conceptuellement simple mais est bien évidemment pénalisante en termes de poids, puissance consommée, volume et coût. Un autre inconvénient réside dans le fait que les instruments localisés à un même point du système peuvent se dégrader en même temps. Ce type de méthode est réservé aux cas où la continuité de service est obligatoire (dans l'industrie nucléaire par exemple), ou pour des sous-systèmes critiques dont la défaillance conduirait à une catastrophe (centrale nucléaire, aéronautique, etc.). Notons que la redondance des actionneurs peut permettre une reprise instantanée.

1.4.2.2 Les méthodes basées sur le dépassement de seuil

Pour des systèmes moins critiques, on se contente souvent de comparer les mesures effectuées sur le procédé à un référentiel correspondant à un état normal de fonctionnement. La nature de ce référentiel peut varier selon la connaissance que l'on possède du système physique : cela va du domaine de valeurs caractéristiques du comportement normal (évoqué dans ce point) pour les systèmes simples à la définition d'un modèle du procédé pour les systèmes plus élaborés (évoquée au point 1.4.3). La notion de référentiel est illustrée par la figure 1.12 :

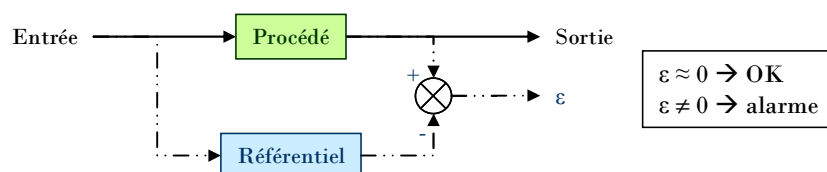


Figure 1.12 La notion de référentiel

Notamment, de nombreux systèmes industriels sont surveillés par de simples procédures de traitement d'alarmes. Les alarmes sont des symptômes de comportement anormal exploités au niveau de la

surveillance et de la supervision. Elles vérifient que les variables process restent à l'intérieur d'un domaine de valeurs caractéristiques du comportement normal (le référentiel). Ainsi, dans le cas d'une grandeur physique x_{mes} on vérifie que : $\forall t, x_{min} \leq x_{mes} \leq x_{max} \rightarrow$ sinon il y a l'émission d'une alarme.

Par exemple, pour détecter un débordement de cuve ou une vidange complète, on surveille les valeurs de la rétention liquide $U_1(t)$ dans la cuve : $\forall t, U_{lmin} \leq U_1(t) \leq U_{lmax}$. De même, pour détecter l'emballement d'une réaction exothermique, on travaille sur la variation de température du réacteur $\Delta T(t)$: $\forall t, \Delta T_{min} \leq \Delta T(t) \leq \Delta T_{max}$.

Dans le cas de variables discrètes, on vérifie que l'état atteint appartient bien à l'ensemble des états admissibles du système à un instant donné. On vérifie alors que les durées observées sont conformes aux durées opératoires prévues.

Le dépassement de seuil est donc utilisé pour déclencher de manière automatique des commandes de mises en sécurité : fermeture/ouverture de vannes si le niveau haut limite est atteint, arrêt du chauffage si la température a atteint la température seuil, mise en route d'une pompe de secours si la pompe usuelle est en surchauffe...

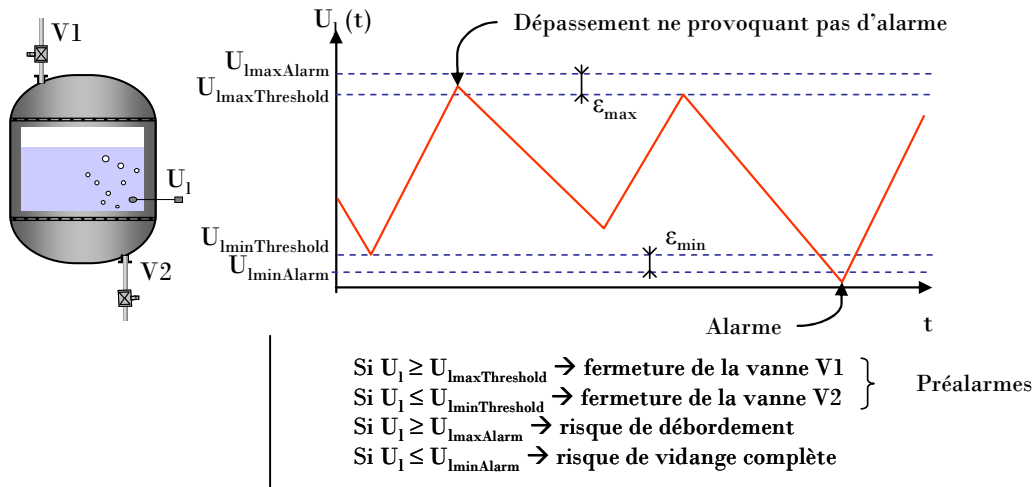
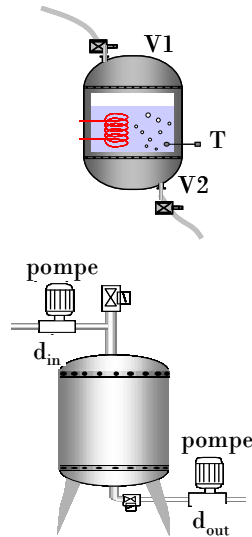


Figure 1.13 Les seuils des alarmes

Le choix des seuils est donc capital. Souvent, quatre valeurs de seuils sont mises en place pour les mesures critiques. Ceci est illustré sur la figure 1.13 qui représente une cuve subissant successivement une série de cycles de remplissage et de vidange. Le choix du seuil de déclenchement de l'alarme est alors un compromis. En effet, un écart ϵ_{min} ou ϵ_{max} trop faible risque de générer fréquemment de fausses alarmes (liées à des perturbations). Au contraire, lorsque les écarts ϵ_{min} ou ϵ_{max} sont élevés, on considère le procédé en fonctionnement normal alors qu'il est en défaut. Ce défaut sera alors détecté tardivement (notamment pour les défauts s'installant lentement).

1^{er} cas :

La surveillance du gradient de température : Un gradient $\Delta T(t) \geq \Delta T_{\max}$ peut être considéré comme normal en chauffe, alors qu'il est le symptôme d'une déviation nécessitant une action correctrice en réaction.

2^{ème} cas :

La surveillance d'un réservoir en phase de remplissage : On considère normale toute situation vérifiant :

- $U_{\min} \leq U_1(t) \leq U_{\max}$
- $d_{\text{in}} \leq \Delta U_1(t) \leq d_{\text{out}}$

Dans le cas d'une fuite, aucun dysfonctionnement n'est détecté tant que la variable d'état $U_1(t)$ et sa variation ne sortent du domaine normal.

Figure 1.14 Les limites de l'observation

Cependant, les phénomènes continus à surveiller sont souvent étroitement liés à la phase de fabrication dans laquelle se trouvent la commande et le procédé. La vérification des conditions sur les variables continues ne peut donc se faire sans connaître l'état (la configuration) du système. Par exemple, dans le premier cas de la figure 1.14, la surveillance conduit à signaler un dysfonctionnement inexistant. Il y a alors génération d'un évènement non pertinent. Dans le deuxième cas de la figure 1.14, la surveillance ne détecte pas la défaillance de la cuve. Ici, il y a perte d'un élément significatif.

À la vue de ces exemples, on peut conclure que les alarmes conviennent essentiellement à la surveillance du procédé en régime établi autour d'un point de fonctionnement constant. Par contre, pour les procédés fonctionnant avec de nombreux régimes transitoires, la surveillance fondée uniquement sur dépassements de seuil donne généralement des résultats non satisfaisants. Par ailleurs, d'autres facteurs limitent cette approche :

- Le système de traitement d'alarmes aide l'opérateur dans sa tâche de surveillance. Néanmoins, toute la partie expertise de la défaillance et prise de décision est sous la responsabilité de l'opérateur : identification et localisation de la défaillance, fonction de reprise (conduite en mode dégradé, activation de la procédure d'arrêt d'urgence, maintenance...). Le comportement de l'opérateur (a-t-il de l'expérience ? est-il entraîné ? est-il attentif ? est-il stressé ?) conditionne alors sa capacité à répondre à une défaillance.
- Les informations disponibles la plupart du temps sont celles utiles pour la commande. Elles ne sont pas toujours celles nécessaires à la surveillance (exemple : mesure de la variation de rétention). Ceci complique la tâche de surveillance de l'opérateur dont la performance est alors limitée.
- De plus, lorsqu'un dysfonctionnement se propage dans toute l'installation, les variables sortent de leur domaine normal dans un ordre aléatoire : les variables ayant des seuils faibles vont déclencher une alarme avant celles qui ont des seuils élevés, même si leur comportement est la conséquence de ces dernières et non leur cause. On obtient alors une séquence temporelle d'alarmes difficilement interprétable.

- Enfin les valeurs limites sont définies par des experts. Ce choix a priori des seuils min et max de chaque variable n'est en rien trivial.

1.4.3 L'intérêt d'un modèle du procédé pour la surveillance

L'utilisation d'un modèle du procédé (quelle que soit sa nature, cf. chapitre 3) comme référentiel permet de résoudre une partie des problèmes soulevés par la supervision fondée sur des alarmes. En effet, si l'évolution du modèle peut être synchronisée avec l'évolution du procédé surveillé, notamment durant les phases transitoires ou les changements de point de fonctionnement, alors les écarts de comportement peuvent être a priori attribués à la présence d'une défaillance [Boillereaux et Flaus, 2003].

Néanmoins, les incertitudes sur les mesures, ainsi que la qualité du modèle utilisé, peuvent aussi conduire à ces écarts. Dans ces conditions, un filtrage peut s'avérer nécessaire afin de réduire leur impact. Par ailleurs, les seuils utilisés dans ce contexte peuvent être beaucoup plus faibles que ceux des alarmes fixes, conduisant de ce fait à une détection plus précoce.

La *redondance analytique* utilise un modèle et est un outil intéressant pour des études de surveillance. Cette notion est illustrée sur la figure 1.15. Grâce à la mesure de l'entrée par le capteur 1 et à ce modèle, la grandeur de sortie est prédite. L'écart entre cette prédiction et la mesure de la sortie par le capteur 2 constitue un indicateur de défaut. Si cet écart atteint une valeur significative, il devient un symptôme du comportement anormal du système : un des composants ou dispositifs du système physique est défectueux.

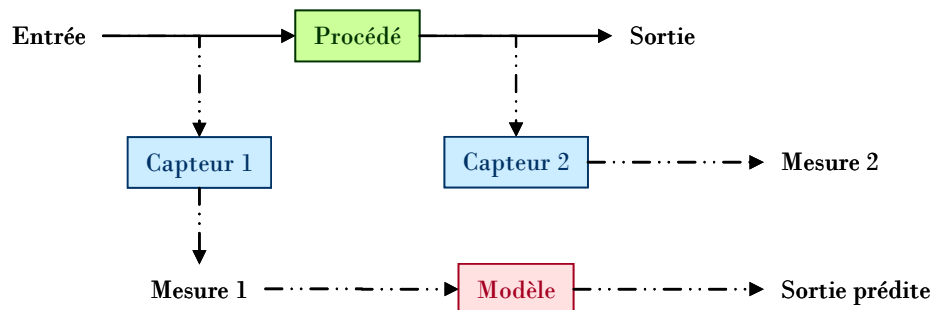


Figure 1.15 La redondance analytique

1.4.4 Le cadre des travaux

Ce point souligne le cadre de ces travaux. Dans un premier temps, les connaissances à avoir sur le procédé sont répertoriées. Ensuite, les caractéristiques d'un système de surveillance sont présentées. Enfin, les objectifs de ces travaux sont exposés.

1.4.4.1 Les bases

Afin de mettre en place un système de surveillance, il faut avoir des connaissances approfondies de l'installation : cela va des connaissances de son comportement normal aux connaissances de son comportement anormal. Ainsi, dans un cas idéal, il faudrait avoir la connaissance de :

- la nature des causes de défaillances à localiser,

- la connaissance des symptômes associés aux défaillances induites par les causes,
- la maîtrise des moyens de mesure et de traitement des symptômes,
- la connaissance des mécanismes physiques entre les causes et les effets,
- l'inventaire du retour d'expérience,
- le recensement des expertises disponibles,
- la définition du niveau de confiance dans le diagnostic,
- l'identification des utilisateurs finaux du diagnostic,
- la modélisation d'un défaut avec les outils pour modéliser le comportement normal du procédé.

Néanmoins dans la réalité, il est parfois difficile d'avoir cette connaissance exhaustive. Seul un sous-ensemble de ces éléments est en général disponible.

1.4.4.2 Les caractéristiques

Selon [Venkatasubramanian *et al.*, 2003], un système de surveillance doit posséder dans l'idéal les propriétés suivantes :

- rapidité de détection et diagnostic (en temps réel),
- identification et localisation fines du défaut,
- capacité à prédire le degré de criticité de la défaillance (important pour la fonction reprise),
- robustesse du système face aux bruits et incertitudes,
- adaptabilité et portabilité (capacité à évoluer devant de nouvelles situations : perturbations, variations des conditions opératoires...),
- capacité à évoluer (à analyser de nouvelles situations et à intégrer de nouveaux cas de défaut...),
- facilité de conception et implémentation informatiques (faible demande de temps CPU...),
- effort de modélisation pour le déploiement rapide et efficace des classificateurs de diagnostic,
- capacité à identifier plusieurs défauts.

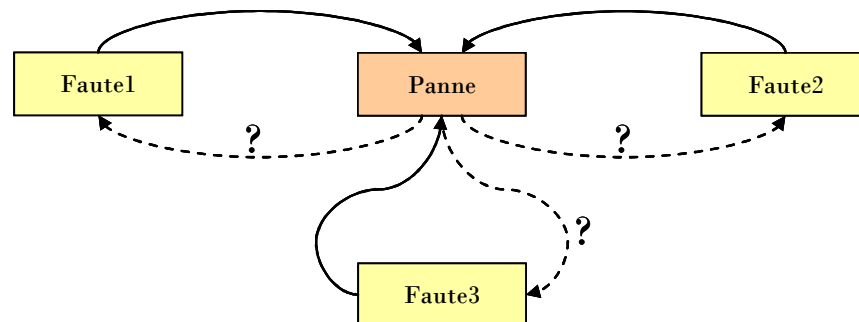
Là encore, les systèmes de surveillance ne possèdent généralement qu'un sous-ensemble de ces propriétés.

1.4.4.3 Les objectifs de nos travaux

Le développement d'une procédure de détection et localisation de défaut en ligne robuste doit pouvoir répondre aux contraintes suivantes [Orantes Molina, 2005] :

- Les caractéristiques temporelles des défauts sont inconnues ;
- Le modèle du système (s'il existe) est incertain ;
- Les bruits du modèle et des mesures sont pris en compte ;
- Il est donc nécessaire de fonctionner en temps réel.
- Le temps de détection et de diagnostic doit être obtenu en temps fini et est fonction de la dynamique du système : plus la dynamique du système sera rapide, plus le temps de réponse requis sera court.

Le problème est donc de développer une méthodologie de diagnostic capable d'appréhender des systèmes dynamiques complexes, ici les procédés. La phase de détection est une phase cruciale pour la fonction de surveillance. En effet, elle permet de gérer et d'analyser une masse importante d'informations disponibles. C'est une étape de décision qui permet de lancer le diagnostic et elle permet l'établissement d'un diagnostic le plus complet possible. La non fiabilité des mesures réalisées sur le procédé rendent ces tâches plus complexes. De plus, les phénomènes sont difficilement interprétables par l'absence de certaines grandeurs pertinentes. Certes, à partir de la faute, il est aisé de prédire la panne, mais la réciproque « trouver la faute à partir des effets » n'est en rien triviale (cf. figure 1.16).



Les fautes, Faute1, Faute2, et Faute3 conduisent à la même panne.
Comment savoir quelle est la cause la plus probable de la panne ?

Figure 1.16 La procédure de diagnostic

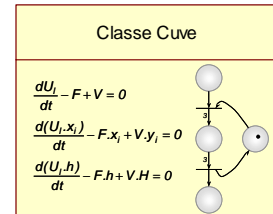
1.5 LE BILAN

Les sections précédentes ont permis de dresser un bref panorama des éléments et concepts classiques caractérisant les systèmes étudiés et leur supervision. Notre approche de diagnostic s'inscrit dans ce cadre :



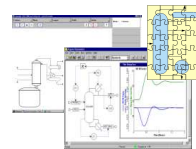
Comme l'a montré la section 1.1, ce travail s'intéresse principalement au domaine du génie des procédés. La première partie du chapitre s'est proposé d'en définir les grandes lignes. Les spécificités des procédés discontinus ont été exposées et une norme internationale pour décrire les procédés a été détaillée : la notion de recette a été introduite.

De plus, cette étude se base sur la modélisation et la simulation des procédés. Leur représentation nécessite l'utilisation de modèles dits « hybrides ». Ces derniers s'avèrent bien adaptés pour modéliser les processus dynamiques régissant les procédés puisque ces processus font intervenir explicitement des phénomènes à dynamique continue et événementielle. La section 1.2 s'est attachée à définir les systèmes dynamiques hybrides et à présenter leur modélisation. Le chapitre suivant présente le modèle utilisé dans ces travaux : les réseaux de Petri différentiels à objets.



La section 1.3 a présenté la hiérarchisation des niveaux de décision d'une entreprise et a ainsi souligné la place de la conduite dans cette hiérarchisation. Cette dernière se décompose en deux sous-niveaux le contrôle/commande et la supervision. Dans le cadre de ces travaux, l'accent est mis sur la supervision et en particulier la fonction de surveillance qui la compose.

Enfin la section 1.4 a mis en avant l'enjeu industriel d'un système de surveillance avancée. Les principales caractéristiques attendues de ces systèmes ont été énoncées. Dans ce mémoire, une méthode à base de modèle est développée pour résoudre les problèmes de détection et diagnostic de fautes.



CHAPITRE 2 LA PLATE-FORME DE SIMULATION *PRODHYS*

*L'unification des travaux de recherche en modélisation et simulation de procédés menés depuis plusieurs années au sein du Laboratoire de Génie Chimique a permis de développer une plate-forme de simulation dynamique hybride nommée ProDHyS (**P**rocess **O**bject **D**ynamic **H**ybrid **S**imulator). Son développement a ainsi conduit à l'intégration de plusieurs modules répondant à des fonctions propres.*

L'objectif de ce chapitre est de décrire les principes fondamentaux de ProDHyS à travers la mise en place d'un système hydraulique utilisé comme exemple didactique.

Ces vingt dernières années, les systèmes dynamiques hybrides ont reçu une attention particulière. La plate-forme *PrODHyS* (**P**rocess **O**bject **D**ynamic **H**ybrid **S**imulator), développée au sein du LGC, s'inscrit dans cette thématique. L'objectif premier est de représenter de façon rigoureuse et explicite, aussi bien la partie continue que la composante événementielle du système physique étudié. L'enjeu suivant consiste en la validation du précédent travail par le biais de la simulation, fonction souvent indispensable pour la vérification du modèle.

Ainsi PrODHyS est une plate-forme évolutive composée de modèles réutilisables. Cette plate-forme adopte une philosophie modulaire en adéquation avec CAPE – OPEN [Belaud et Pons, 2002 ; Gani et Braunschweig, 2002]. Elle propose un environnement capable de simuler aussi bien les composantes continues et discrètes d'un système, grâce à l'utilisation du formalisme *Réseau de Petri Différentiel Objet*.

Ce chapitre est consacré à l'introduction du formalisme utilisé pour la modélisation et la simulation des systèmes dynamiques hybrides. Dans un premier temps, nous présentons les objectifs de la simulation dynamique hybride, desquels découlent les concepts fondamentaux mis en œuvre dans *PrODHyS*. Ensuite, le formalisme et le processus de modélisation d'un procédé sous *PrODHyS* sont décrits. Enfin, ces concepts sont illustrés à travers la simulation d'un exemple didactique.

2.1 LA SIMULATION DYNAMIQUE HYBRIDE

2.1.1 Les concepts

L'architecture générale d'un simulateur dynamique hybride est décrite sur la figure 2.1. La simulation dynamique hybride intègre plusieurs étapes : la *simulation de la partie discrète* qui consiste à jouer le modèle discret, la *simulation de la partie continue* (résolution d'un système d'équations composé traditionnellement d'équations différentielles ordinaires et partielles et d'équations différentielles algébriques) et la *mise en place du modèle continu global*, obtenu par concaténation des sous-modèles actifs du système.

À l'heure actuelle, les recherches concernant les méthodes et outils formels relatifs à l'analyse du comportement des SDH en sont encore à leur début. La simulation reste donc un passage obligé lorsque l'on envisage :

- d'aider à la conception ou au dimensionnement d'une installation et de valider cette conception,
- d'optimiser des procédés existants,
- d'ajuster la recette de fabrication aux contraintes de production (disponibilité des ressources humaines et du procédé, respect des contraintes temporelles, vérification des objectifs de qualité),
- de valider le modèle élaboré (dans un but de prévision, par exemple) pour une installation existante,
- de valider la commande conçue pour une installation,
- de tester le programme de commande exécuté sur PC ou sur automate programmable industriel,
- d'entraîner les opérateurs (arrêt, démarrage, incident...).

Plusieurs objectifs peuvent être poursuivis simultanément.

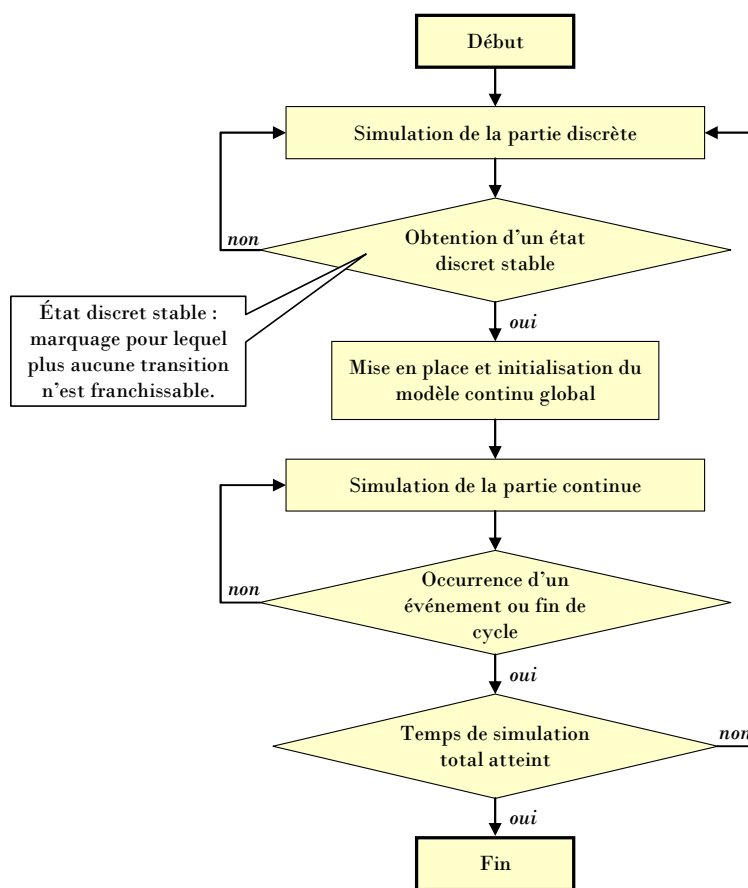


Figure 2.1 L'architecture générale d'un simulateur dynamique hybride

2.1.2 Les spécificités de la simulation des procédés

L'étude des opérations unitaires offre de nombreux avantages tant au niveau de la conception et de l'exploitation du procédé, qu'au niveau de la conduite optimale du procédé. Mais le plus souvent, le procédé à étudier n'offre pas suffisamment d'informations pour comprendre les phénomènes mis en jeu. Une étude par simulation devient alors le seul recours.

La simulation des procédés est aujourd'hui un outil incontournable pour la résolution d'un grand nombre de problèmes soulevés au cours de chaque étape du développement, de la conception, du fonctionnement ou de l'amélioration du procédé. La mise en place d'une étude de procédé par simulation se déroule généralement en deux temps : la mise en place de la représentation mathématique et sa résolution.

2.1.2.1 Un modèle

Une représentation mathématique du procédé doit tout d'abord être établie afin de reproduire aussi fidèlement que possible le comportement réel du système. Cette représentation mathématique constitue le *modèle* du procédé. Ce modèle s'appuie sur les lois fondamentales de la physique et de la chimie telles que la conservation de la masse, le premier et second principe de la thermodynamique, les équilibres entre phase, les lois de transfert... Le modèle qui en résulte conduit alors à un jeu mixte d'équations différentielles ordinaires ou partielles couplées avec des équations algébriques non linéaires.

L'évolution rapide de la puissance des calculateurs a permis de valider l'utilisation de ces modèles pour la représentation des opérations unitaires classiques de génie chimique.

2.1.2.2 La résolution

La deuxième phase de l'étude de procédé par simulation consiste à résoudre le système d'équations résultant de la première étape. Or, la résolution analytique de tels systèmes est souvent impossible et le recours à des méthodes numériques s'avère inévitable. L'informatique apparaît alors comme un outil essentiel pour leur mise en œuvre. La résolution du système permet de suivre l'évolution des variables d'état au cours du temps et d'étudier ainsi le comportement du système représenté. Cette phase d'étude constitue la partie *simulation* du procédé. Deux approches de simulation sont couramment utilisées pour les systèmes complexes : la simulation modulaire et la simulation globale [Mattsson *et al.*, 1993 ; Thévenon, 2000].

2.1.2.2.1 La simulation modulaire

La première démarche pour résoudre ce genre de problèmes correspond à une approche séquentielle-modulaire, c'est-à-dire qu'à chaque opération unitaire correspond un sous-programme auquel on fournit des variables d'entrée (courants d'alimentation, conditions opératoires), et où sont calculées les variables de sortie (courants de sortie). Le comportement global du système résulte alors du comportement local de chaque module déterminé séquentiellement selon un ordre bien défini, les entrées des modules étant modifiées par le biais de relations spécifiques entre modules.

2.1.2.2.2 La simulation globale

Dans le cas de la simulation globale, le système d'équations complet est obtenu par concaténation des systèmes associés aux modules actifs ; il est ensuite résolu globalement par un solveur puissant. Cette approche permet de lever les difficultés liées aux problèmes de dépendance mutuelle des variables. De plus, la séparation de la phase de modélisation et de la phase de simulation permet une meilleure analyse et manipulation du modèle et induit une simulation plus efficace.

En revanche, la difficulté de cette approche repose sur la gestion de la structure du modèle. En effet, celle-ci est variable, c'est-à-dire que le nombre d'équations et de variables d'état varie au cours de la simulation. Le système complet est généralement de grande taille et son intégration nécessite une méthode de résolution puissante et des moyens de calcul importants.

2.1.2.3 Le cas particulier des procédés discontinus

Par ailleurs, lorsque les systèmes de production visés sont des procédés discontinus (ou batch) ou semi-continus, la capacité de ces simulateurs à gérer simultanément et de manière cohérente des comportements continus et discrets devient un atout majeur. En effet, ces unités marient des séries d'opérations qui fonctionnent en continu (qu'il s'agisse d'un régime transitoire ou permanent) avec des opérations purement discontinues. Il en résulte que la recette comporte une forte composante événementielle mais dépend aussi de paramètres à valeurs continues (quantités de matière utilisées, conditions opératoires telles que température, pression, concentration, etc.). De ce fait, elle est souvent définie par des événements d'état (seuil de température ou de composition par exemple) et non par des

durées ou des dates d'occurrence fixées a priori. Par conséquent, la modélisation des opérations et des transformations physico-chimiques des produits nécessite l'implantation de modèles phénoménologiques. Cependant, la complexité de ces opérations conduit parfois à simplifier ou négliger certains phénomènes physiques, selon le niveau de finesse ou le temps de réponse exigé pour la simulation. Par exemple, si l'évolution de l'état physique de la matière est par nature continue, sa modélisation fait souvent apparaître des aspects événementiels (dus au changement de phase, notamment). Il en résulte que les modèles dynamiques apparaissent ici encore comme des modèles continus par morceaux ou même discontinus.

2.2 LES CONCEPTS FONDAMENTAUX DE *PRODHYS*

Aujourd'hui, la simulation dynamique constitue un outil d'études privilégié lors des phases d'analyse et de conception des systèmes industriels. L'apparition de logiciels commerciaux tels que *ProSim Batch* (www.prosim.net) ou *Aspen Dynamics* (www.aspentech.com) en est la preuve. Parallèlement, ces simulateurs ont largement profité de l'évolution rapide de la puissance des calculateurs ainsi que d'avancées considérables dans les stratégies de résolution notamment de grands systèmes d'équations algèbro-différentielles. Dans ce contexte, les travaux de recherche entrepris depuis plus de quinze ans au sein du *Laboratoire de Génie Chimique (LGC)* ont conduit au développement progressif d'une plate-forme de simulation nommée *PrODHyS (Process Object Dynamic Hybrid Simulator)*.

Par ailleurs, l'évolution des procédés de fabrication vers des processus batch confronte les modélisateurs avec des modes de fonctionnement souvent difficiles à gérer avec des modèles purement continus ou purement discrets. Ces modèles sont à structure variable et nécessitent donc un noyau capable de gérer automatiquement ces changements de configuration. Un des moyens pour gérer les séquences légales de commutations entre les différents systèmes d'équations algèbro-différentielles est de mettre en place un modèle à état discret. On parle alors de modèles dynamiques hybrides [Zaytoon, 2001]. Dans ce contexte, les outils classiques de simulation dynamique continue ou de simulation à événements discrets d'ateliers s'avèrent mal adaptés à cette classe de problèmes. Pour ces raisons, les systèmes dynamiques hybrides constituent désormais un intérêt certain [Zaytoon, 2001] et de nombreux outils de simulation ont été développés pour les représenter. Nous pouvons citer par exemple *gPROMS* [Barton et Pantelides, 1994], *shift* [Deshpande *et al.*, 1998], *Omsim* [Andersson, 1994], *Chi* [Fabián *et al.*, 1998], *BaSiP* [Wöllhaf *et al.*, 1996], *ABACUSS II* (www.yoric.mit.edu/abacuss2/abacuss2.html).

Parallèlement, de nombreux formalismes hybrides ont été définis ou obtenus par extension de formalismes discrets ou continus existants. Ils combinent simultanément les aspects discrets et continus. Nous pouvons citer les plus classiques : les *automates hybrides* [Alur *et al.*, 1995], les *réseaux de Petri prédicats-transitions différentiels* [Champagnat *et al.*, 1998], les *réseaux de Petri hybrides* [Le Bail *et al.*, 1991], les *statecharts hybrides* [Kesten et Pnueli, 1992].

De la même manière, *PrODHyS* a adopté une modélisation hybride des procédés en tenant compte simultanément des aspects continus, décrits par des équations différentielles algébriques (EDA) et des aspects discrets, décrits par des réseaux de Petri. La philosophie de *PrODHyS* est donc de permettre la modélisation fine de la structure et du comportement complexe des procédés, dans un objectif de simulation dynamique. L'élaboration d'un modèle dynamique capable de les simuler passe alors par une

décomposition du procédé en sous-systèmes plus simples. Cette vision systémique et hiérarchisée d'un procédé s'adapte particulièrement bien à une *approche orientée objet*. Dès le départ, celle-ci a été exploitée en vue :

- d'améliorer la qualité du logiciel et produire des bibliothèques de composants logiciels extensibles et réutilisables,
- de faciliter la création de ces composants par le programmeur mais aussi par un modelleur grâce à un « vocabulaire » plus riche et plus proche de sa compétence métier,
- de mettre, à terme, ces composants à la disposition de l'utilisateur par l'intermédiaire d'une interface graphique au sein de laquelle chaque objet aura une « image active ».

La représentation orientée objet des procédés industriels est un thème traité dans la littérature depuis plusieurs années. C'est dans ce contexte que les bases fondamentales de *PrODHyS* ont été définies [Jourda *et al.*, 1996 ; Sargousse, 1999 ; Moysse, 2000 ; Perret, 2003]. Ces travaux s'axent, soit sur la représentation des procédés à un haut niveau [Bourseau, 1993 ; Stephanopoulos *et al.*, 1990], soit sur une représentation structurée mais purement mathématique des modèles [Piela, 1989 ; Nilsson, 1993] (en faisant abstraction d'une représentation moins formelle liée à la structure matérielle des appareils). La conception de *PrODHyS* a cherché à concilier ces deux approches, tout en séparant la représentation de la structure des procédés (c'est-à-dire la constitution de chaque appareil et leur connexion au sein du procédé) de la représentation utile à la simulation (c'est-à-dire les modèles mathématiques formels décrivant leur comportement). Notamment, chaque module repose sur l'identification de composants élémentaires généraux et réutilisables dont l'assemblage permet de construire des modèles de simulation élémentaires, eux-mêmes étant des composants de modèles de simulation plus complexes et spécialisés. C'est pourquoi un des principaux apports de ces travaux réside dans la détermination et la conception de ces briques élémentaires initiales et non dans l'élaboration d'une bibliothèque riche de modèles. La dernière évolution majeure de *PrODHyS* a été l'intégration du noyau de simulation dynamique hybride [Perret, 2003] basé sur le formalisme réseau de Petri différentiel objet (RdPDO).

2.3 L'ARCHITECTURE LOGICIELLE DE *PRODHYS*

La plate-forme *PrODHyS* est donc une bibliothèque de classes destinées à être dérivées en exploitant les mécanismes objets (polymorphisme, composition, héritage, généricité). Son développement s'appuie sur un processus de conception et de codage unifié, basé sur le formalisme UML et le langage objet C++. Elle est organisée en deux couches fonctionnelles indépendantes. La couche simulation regroupe les classes correspondant au noyau de simulation (gestionnaire de simulation, formalisme RdPDO, intégrateurs EDA (Équation Différentielle Algébriques), solveurs EANL (Équations Algébriques Non Linéaires)). Elle fournit donc les objets de base utiles à la simulation de tout système dynamique hybride, quel que soit le domaine d'application. Quant à la couche modélisation, elle rassemble les classes utilisées spécifiquement pour la modélisation des procédés. Actuellement, cette bibliothèque regroupe plus de mille classes réparties en sept modules. Ceux-ci sont représentés de façon schématique sur la figure 2.2 qui met en évidence les dépendances (flèches en pointillées) existant entre eux et leur structure en deux couches logicielles.

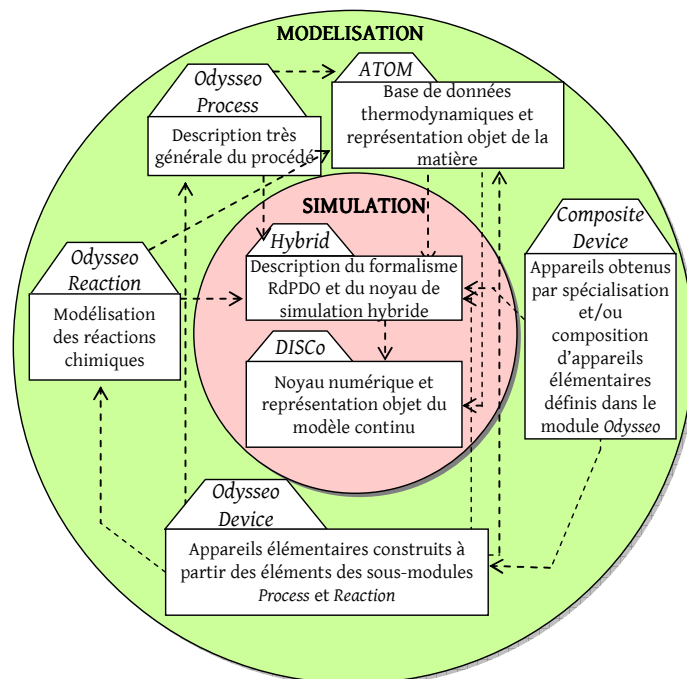


Figure 2.2 L'architecture logicielle de PrODHyS

La couche interne de *PrODHyS* correspond au noyau de simulation. Elle fournit les éléments de base utiles à la simulation de tout système dynamique, quel que soit le domaine d'application. Elle est composée :

- du module *DISCo* [Sargousse, 1999 ; Le Lann, 1999] qui constitue le noyau numérique du système ; il offre un ensemble de solveurs et d'intégrateurs (EDA, EANL) et permet une représentation objet des modèles mathématiques continus. Le module *DISCo* est développé dans l'annexe C.
- du module *Hybrid* qui contient l'ensemble des classes nécessaires à la description du formalisme Réseau de Petri Différentiel Objet (RdPDO) ainsi que du noyau de simulation hybride.

Quant à la couche supérieure, elle rassemble les classes utilisées pour la représentation de la structure des procédés. Cette couche comprend :

- le module *ATOM* [Jourda, 1996 ; Jourda et al., 1996] 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 via le composant *ProPhy32* développé par la société *ProSim SA*.
- le module *Odysseo* [Moyse, 2000] regroupant les entités élémentaires et génériques qui permettent de modéliser un procédé. Il comprend 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 répertorie les appareils élémentaires « concrets » du procédé, construits à partir des éléments des deux précédents sous-modules.

- le module *CompositeDevice* qui rassemble les appareils obtenus par spécialisation et/ou composition d'appareils élémentaires définis dans le module *Odysseo*.

Cette architecture en deux couches fonctionnelles rend ainsi possible l'implémentation de plates-formes de simulation dédiées à des domaines d'ingénierie variés (électronique, productique, etc.) simplement en développant la couche modélisation adaptée. De plus, elle permet 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.). Dans notre cas, cela correspond au développement des modules *Process*, *Reaction* et *Device* dédiés au domaine des procédés.
- 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'utilisateurs 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*.

2.4 LE FORMALISME RDPDO

Dans cette partie, seuls les principes de base sont présentés pour permettre d'appréhender la manière dont les modèles de simulation sont spécifiés via *PrODHyS*. Le lecteur trouvera la définition formelle des RdpDO en annexe (cf. annexe B).

2.4.1 La spécification du comportement dynamique

Pour pouvoir utiliser cette plate-forme dans des domaines variés, elle doit décrire de façon rigoureuse et précise aussi bien les composantes continues que discrètes des systèmes à simuler. C'est pourquoi, le noyau de simulation de *PrODHyS* est fondé sur un formalisme mixte clairement défini.

2.4.1.1 La partie continue

Une façon rigoureuse de représenter les aspects continus est d'utiliser des systèmes d'équations différentielles algébriques. Compte tenu de la complexité des phénomènes physico-chimiques rencontrés en génie des procédés, ils ont donc été adoptés pour simuler l'évolution continue des variables d'état. Cette solution nécessite l'utilisation d'un solveur d'équations différentielles algébriques possédant de nombreuses fonctionnalités (réduction d'index, recherche de conditions initiales cohérentes, détection d'événements, etc.). Rappelons aux lecteurs l'existence du module *DISCO* au sein de la plate-forme. La gestion de la partie continue est donc bien établie.

2.4.1.2 La partie discrète

En revanche, de nombreux phénomènes discrets intrinsèques aux systèmes eux-mêmes (changements de phase par exemple) ou résultant du contrôle, de la commande ou de la supervision, ne peuvent être traités de façon simpliste en intégrant, par exemple, des variables booléennes au sein du modèle continu. Par conséquent, un modèle discret doit être mis en place afin de permettre une description aussi fidèle que possible de la composante événementielle du système. Un modèle discret basé sur les réseaux de Petri a donc été développé afin de représenter efficacement le séquençement, mais aussi les mécanismes de synchronisation, de parallélisme, d'affectation des ressources ou d'exclusion mutuelle.

2.4.1.3 La structure de données

Par ailleurs, les systèmes rencontrés nécessitent souvent de manipuler et de structurer d'abondantes quantités d'informations. Par exemple, en génie des procédés, l'information contenue dans les systèmes est très dense (propriétés physico-chimiques de la matière, spécification des conditions opératoires, paramètres géométriques d'une opération unitaire, etc.).

L'utilisation de l'approche objet est alors apparue justifiée, afin de décrire les aspects statiques et structurels du système. Les mécanismes de spécialisation, de composition et d'encapsulation, qui sont propres à cette approche, engendrent en effet une description de haut niveau et permettent ainsi de réduire la taille du modèle et d'en améliorer la lisibilité.

Enfin, la puissance de description de la structure de données est augmentée par l'utilisation d'un jeton porteur d'informations : ce dernier est défini par une classe d'objet encapsulant à la fois des données sous forme d'attributs et des méthodes qui traitent ces données.

L'intégration de l'ensemble des concepts décrits précédemment a conduit à la définition du formalisme nommé Réseau de Petri Différentiel Objet (RdPDO) [Hétreux *et al.*, 2003]. Dans la suite nous présentons les principes généraux de ce modèle.

2.4.2 Les éléments du formalisme RdPDO

Le RdPDO est constitué d'un ensemble d'éléments qui permettent de décrire à la fois les aspects dynamiques et structurels du système à modéliser (cf. figure 2.3) :

- de classes objet qui représentent les différents composants du système à simuler,
- de variables formelles typées par des éléments de l'ensemble de classes,
- de jetons, instances de classes d'objet formées d'attributs et de méthodes,
- de places discrètes, symbolisées par un simple cercle, qui représentent un état dans lequel les variables continues du système n'évoluent pas au cours du temps,
- de places différentielles, symbolisées par deux octogones concentriques, modélisant le comportement continu (matière, opérations unitaires...) au moyen d'un système d'équations différentielles algébriques,
- de transitions caractérisées ou non par des conditions et des actions,

- des arcs typés (porteur d'informations sur les classes de jetons autorisés à transiter sur l'arc), inhibiteurs (permettant alors de tester l'absence de jeton de type(s) indiqué(s) sur l'arc dans la place en amont) ou non.

La figure suivante illustre la sémantique adoptée :

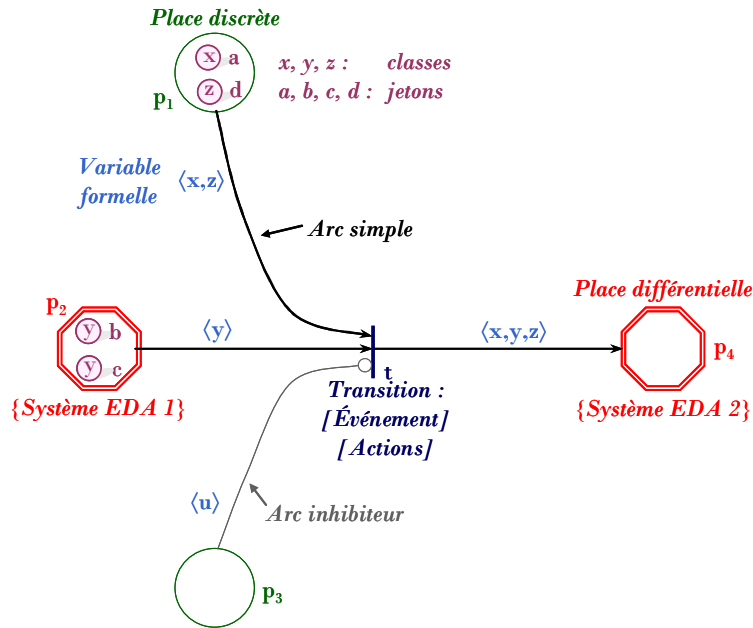


Figure 2.3 La sémantique du formalisme RPDO

Indiquons ensuite que dans ce formalisme, l'approche objet et les réseaux de Petri sont exploités selon une approche dite combinée étendue. Celle-ci consiste à les faire interagir de deux manières.

2.4.2.1 L'introduction des objets dans les réseaux de Petri

Cette approche considère les jetons qui se déplacent sur le réseau de Petri comme des objets. La philosophie sous-jacente consiste à modéliser les états d'un ensemble d'entités identiques par un réseau de Petri unique dans lequel chaque jeton représente une entité particulière. Ces jetons doivent donc être individualisés et porteurs d'informations. La structure des entités modélisées est définie au moyen d'une classe caractérisée par un ensemble d'attributs et un ensemble de méthodes qui traitent ces données. Chaque jeton correspond alors à une instance de cette classe. La figure 2.4 permet d'illustrer les concepts qui se dégagent de cette approche.

Quatre jetons (a et b de classe p_{in} , c de classe p_{out} et d de classe m) marquent le réseau. Deux jeux de jetons sont susceptibles de franchir la transition : il s'agit soit du jeu (a, d) , soit du jeu (b, d) . Le franchissement de la transition n'est possible que si l'un des jeux vérifie les conditions associées, C_1 et C_2 . La condition C_1 est exprimée en fonction de la variable formelle $\langle m \rangle$; par conséquent, elle doit être testée par le jeton d , étant le seul jeton de type m qui peut se substituer à la variable formelle $\langle m \rangle$. Considérons que la condition C_1 est validée et concentrons-nous sur la condition C_2 . Cette dernière est exprimée en fonction de la variable formelle $\langle p_{in} \rangle$; par conséquent, elle doit être testée avec les jetons de type p_{in} qui

peuvent se substituer à la variable formelle $\langle p_{in} \rangle$. Les jetons concernés sont ici les jetons a et b . Si l'on suppose que seul le jeton a vérifie la condition C_2 , seul le jeu (a, d) peut franchir la transition.

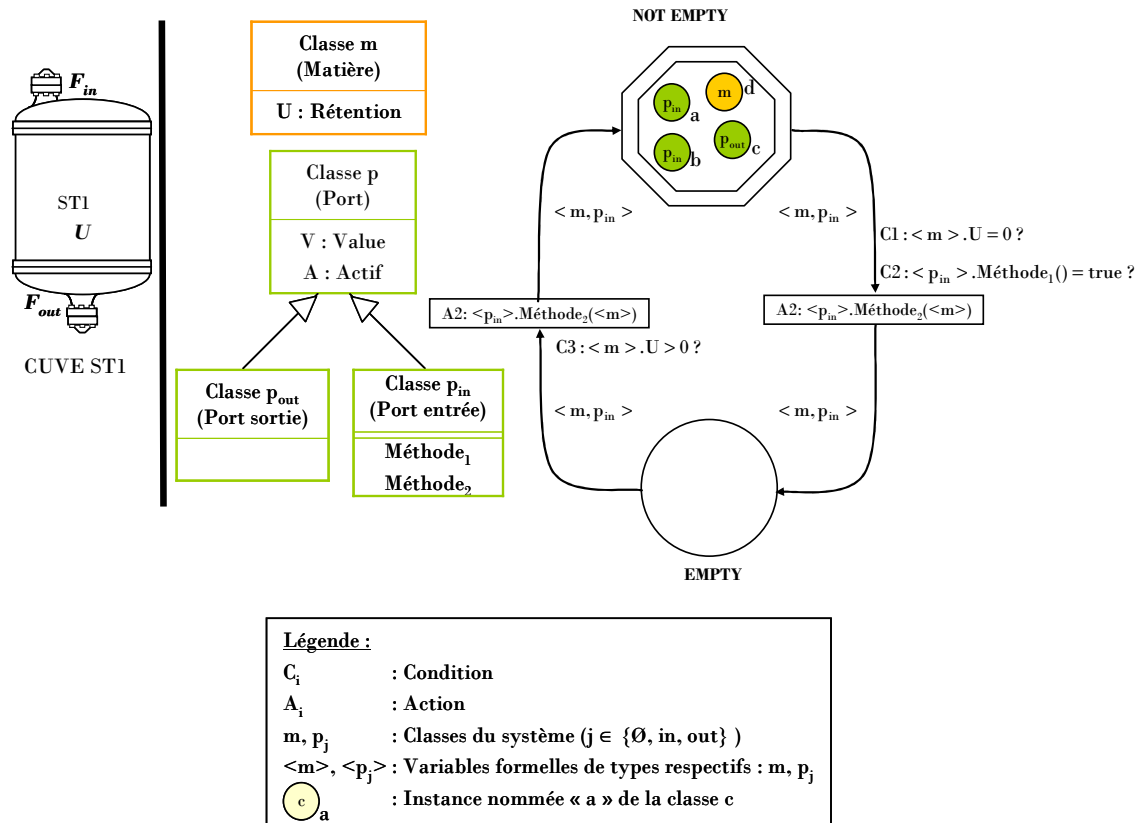


Figure 2.4 L'approche « objet dans les réseaux de Petri »

Le tir de cette dernière implique alors l'exécution de l'action associée. Elle consiste à appliquer la méthode $Méthode_2$ du jeton a substitué à la variable formelle $\langle pin \rangle$ sur le jeton d substitué à la variable formelle $\langle m \rangle$.

2.4.2.2 L'introduction des réseaux de Petri dans les objets

Une autre façon d'aborder le paradigme Réseau de Petri / Objet est d'utiliser les réseaux de Petri pour décrire le comportement interne des objets. Considérons, par exemple, un objet caractérisé par un ensemble d'attributs et de méthodes. Le marquage courant du réseau indique alors l'état interne de l'objet, le franchissement des transitions se traduit par l'exécution d'une ou plusieurs méthodes de l'objet et la structure globale du réseau spécifie les séquences légalés d'exécution des méthodes.

Soit un objet cuve dont le comportement est modélisé par le RdP de la figure 2.5. La méthode `describePetriNet` construit les places, les transitions et les arcs du réseau de Petri ci-dessous. La méthode `loadEvent` décrit les conditions des transitions soit ici C_1 , C_2 et C_3 . La méthode `loadActions` définit les actions associées aux transitions, A_1 et A_2 .

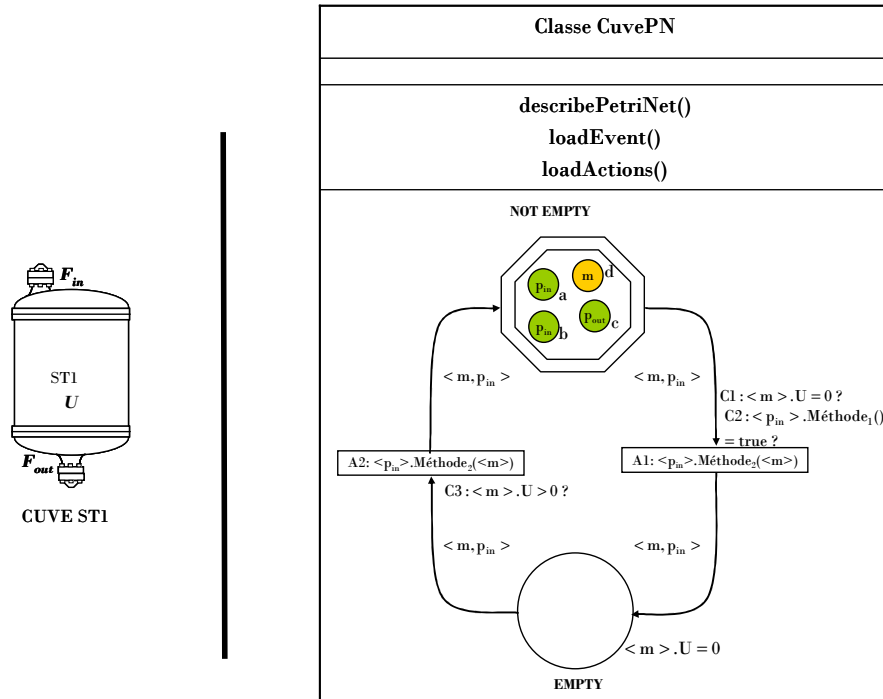


Figure 2.5 L'approche « réseau de Petri dans les objets »

Par ailleurs, l'état privé ou protégé de certains membres de l'objet (attributs et méthodes) rend certaines propriétés de l'objet invisibles depuis l'extérieur. Ils ne peuvent être utilisés que par l'objet lui-même et offrent ainsi une fonction uniquement descriptive de son état et de son comportement internes. Le mécanisme d'encapsulation propre à l'approche objet utilisé ici permet ainsi de découpler distinctement les éléments internes de l'objet et les éléments externes qui lui permettent de communiquer avec les autres objets.

2.4.2.3 L'évolution du modèle RdPDO

Si la partie statique offre un haut niveau d'abstraction grâce aux concepts objets, la partie dynamique fait collaborer au sein d'une même structure :

- des systèmes d'équations différentielles algébriques (EDA) pour décrire l'évolution continue du système,
- des réseaux de Petri utilisés pour spécifier et gérer les séquences légales de commutation entre ces différents systèmes EDA.

La figure 2.6 montre un exemple d'évolution dans les RdPDO. Lorsqu'un jeton marque une place différentielle, le système EDA est instancié par substitution des variables formelles par les attributs du jeton. La résolution induit l'évolution continue de ces attributs ainsi que l'évaluation de la condition associée à la transition (événements d'état ou de temps). Lorsque cette condition est atteinte, l'action est exécutée et la transition est franchie.

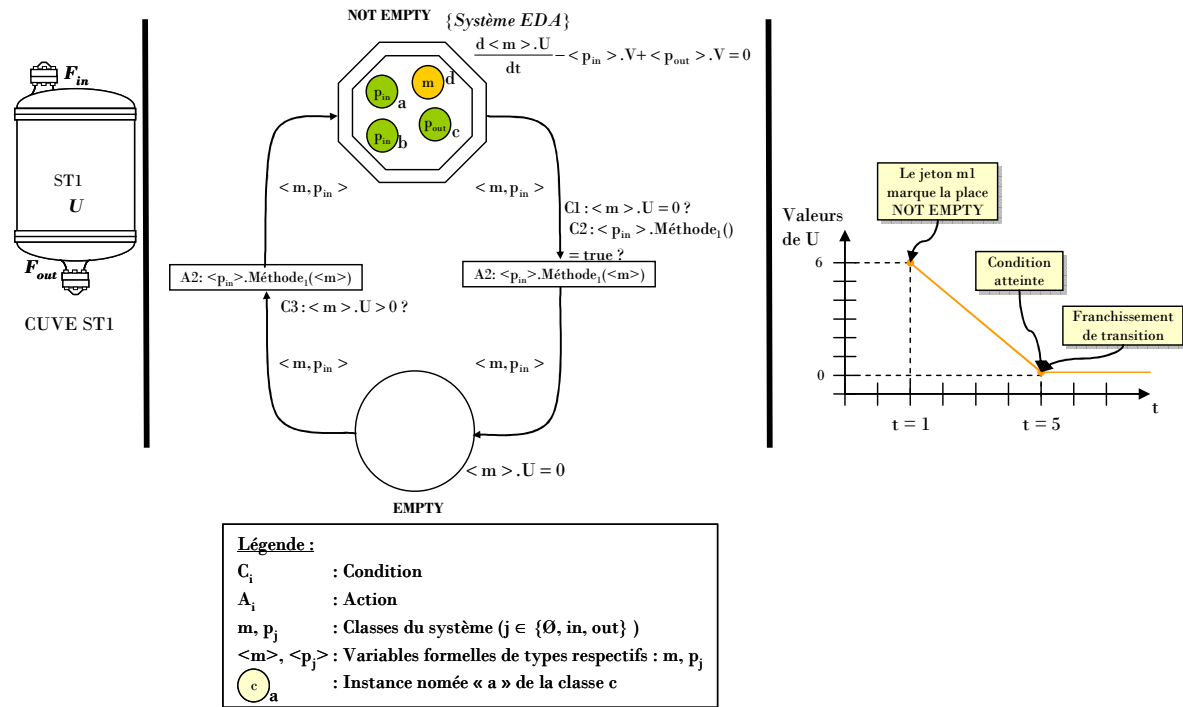


Figure 2.6 Le principe d'évolution des RdPDO dans le modèle simplifié de la cuve

Plusieurs jetons de même type peuvent marquer la place, déclenchant alors l'intégration du même système EDA à partir de la date de marquage. Par ailleurs, l'instanciation du système EDA d'une place différentielle peut nécessiter une combinaison de jetons de même type et/ou de types différents. Dans ce cas, la cohérence de l'ensemble doit être garantie par la modélisation ou validée a posteriori par la simulation. Ainsi, le réseau de Petri peut être vu comme un moniteur d'EDA conduisant à la création dynamique d'un unique modèle de simulation dont la dimension et la structure fluctuent entre deux événements. Outre la résolution des systèmes EDA (basée sur une extension de la méthode de Gear [Gear, 1971]) et des modèles discrets (joueur spécifique à cette classe de RdP), le noyau effectue d'autres traitements fondamentaux tels que le calcul exact des instants de commutation, le recalage des états, la vérification de la cohérence des nouveaux modèles continus générés après commutation, la réinitialisation des variables d'état et de leurs dérivées.

2.5 LA MODÉLISATION D'UN PROCÉDÉ

2.5.1 Le modèle de simulation

La simulation d'un procédé discontinu nécessite de modéliser à la fois la partie commande (le superviseur) et la partie opérative (le procédé). S'appuyant sur les principes de l'approche objet, le modèle relatif au niveau commande (a priori spécifique à la recette et à la topologie du procédé considéré) est complètement dissocié des modèles des appareils qui eux, doivent être réutilisables quel que soit le contexte (notion de composant). La modélisation des appareils et de la matière qu'ils contiennent a déjà été présentée dans de précédentes communications [Olivier *et al.*, 2005 ; Olivier *et al.*, 2006 ; Hétreux *et al.*,

2007]. Rappelons simplement que l'approche combinée Objet / Réseau de Petri est exploitée pour dissocier le modèle de la matière, de l'appareil qui la contient (notion de container). En l'occurrence, le jeton matière permet de gérer les équations du modèle liées à l'état physico-chimique et thermodynamique de la matière (lui-même au moyen d'un RdP) en fonction des conditions locales. De même avec des jetons « port » qui portent les variables de flux et de potentiel. Ces jetons « objet » présentent l'intérêt d'être réutilisables et de réduire notablement la complexité du RdP d'un appareil en ne faisant apparaître que les états inhérents strictement à celui-ci (et non en combinaison avec ceux de la matière qu'il contient, par exemple). Ces différents modèles ne sont finalement fusionnés qu'au moment de l'instanciation du modèle de simulation, en fonction de l'état courant du procédé.

2.5.2 La modélisation de la partie opérative

La spécification structurelle de tout objet de procédé sous *PrODHyS* est toujours définie selon deux axes : un axe topologique et un axe phénoménologique.

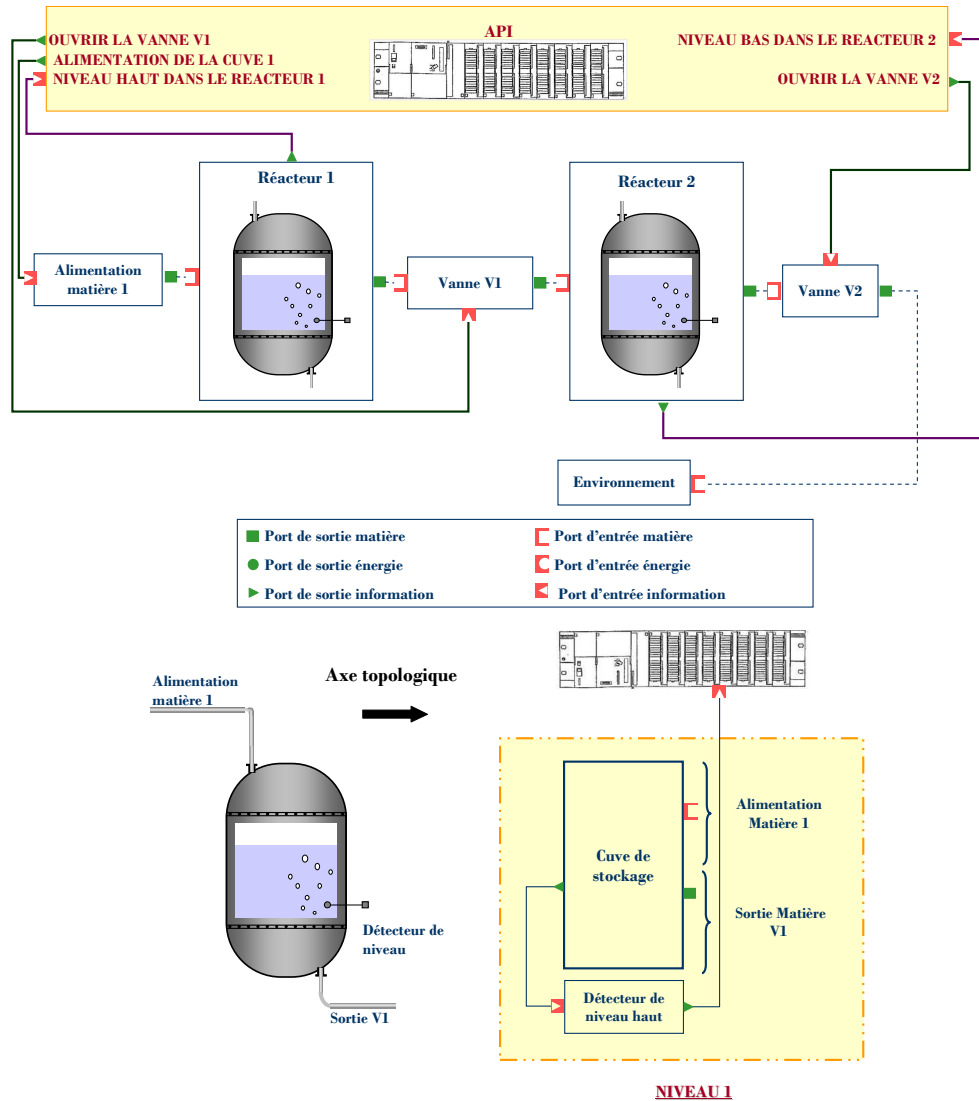


Figure 2.7 La modélisation topologique

2.5.2.1 La modélisation selon l'axe topologique

Il s'agit ici de représenter la structure du procédé selon une vision système, c'est à dire en termes de connexions entre diverses structures composées hiérarchiquement au travers desquelles circulent matière, énergie.

2.5.2.1.1 La modélisation topologique selon l'axe connexion

Cet axe décrit les connexions entre les éléments constitutifs du procédé (cf. figure 2.7). Une connexion correspond simplement à une visibilité offerte, à un élément vers un autre (principe d'encapsulation), par l'intermédiaire de ports, sorte d'interfaces d'entrée-sortie. Ce principe induit qu'il n'existe aucune variable globale et le partage d'une variable entre plusieurs sous-systèmes se fait toujours de manière explicite. Une connexion s'établit par la mise en relation de deux ports mais reste totalement acausale. La notion de *port d'entrée* ou de *sortie* permet seulement de fixer le sens conventionnel du flux (positif si *sortie* \rightarrow *entrée*). Selon cet axe, un procédé est représenté par un graphe orienté.

2.5.2.1.2 La modélisation topologique selon l'axe décomposition/composition

L'axe de *décomposition/composition* décrit la structure interne de chaque entité. Deux types initiaux d'éléments suffisent à son support : les *appareils élémentaires* qui sont non-décomposables et peuvent être spécialisés (fixent la granularité de la modélisation) et les *appareils composés* qui décrivent récursivement la décomposition. Selon cet axe, un procédé apparaît alors sous la forme d'un arbre dont les feuilles sont les appareils élémentaires et dont la racine est un objet général de type Flowsheet. À titre d'illustration, la figure 2.7 montre la structure adoptée, selon ce deuxième axe, pour modéliser un système hydraulique basé sur le principe des vases communicants [Olivier-Maget *et al.*, 2007]. Par ailleurs, chaque appareil élémentaire issu de la décomposition résulte de la spécialisation de composants encore plus généraux définis dans *PRODHYS* [Perret *et al.*, 2004 ; Moysse, 2000].

2.5.2.2 La modélisation d'un procédé selon l'axe phénoménologique

A partir de la topologie du procédé, il faut générer un modèle mathématique (système EDA) représentant le comportement du procédé et de la matière qu'il contient. Il s'agit donc ici d'une vision orientée phénomène reposant sur les lois thermodynamiques et physico-chimiques ainsi que sur la notion de bilan (matière, énergie, etc.). Chaque élément de procédé, quel qu'il soit, est alors considéré comme une frontière de bilans. Nous parlons alors de volume de contrôle pour caractériser l'aspect conteneur de phénomène : un élément de procédé contient directement ou indirectement une portion de matière dans laquelle se produisent un ou plusieurs phénomènes physico-chimiques.

2.5.3 La spécification de la recette

La fabrication d'un produit est constituée d'une suite de tâches qui nécessite de disposer de matières premières, d'appareils, d'utilités énergétiques, de main d'œuvre, etc. La suite des tâches ordonnées dans le temps qui décrit la procédure de fabrication du produit est appelée *recette*. Produire un lot consiste donc à développer la recette de ce lot en spécifiant les affectations d'appareils nécessaires à la réalisation de chaque tâche.

2.5.3.1 La connexion des RdP « appareils » et « recette »

Les signaux échangés entre la partie procédé et la partie commande correspondent soit à l'émission d'une commande soit à la réception d'une information. Un signal est modélisé par une place respectivement dite de commande ou d'information et son état est associé au marquage de cette place. Ces places sont le seul lien entre les appareils commandés et la recette.

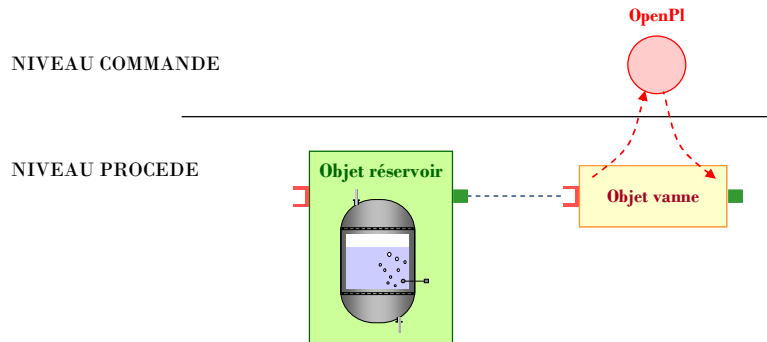


Figure 2.8 La place de signal d'un dispositif actif

Considérés comme des boîtes noires, on distingue alors deux types de dispositif (cf. figure 2.8) :

- les dispositifs actifs : il s'agit des appareils dont le RdP possède une ou plusieurs places de commande et/ou d'information tels que les vannes, les pompes, les alimentations d'énergie, les capteurs, etc.
- les dispositifs passifs : ce sont les entités dont le RdP ne possède aucun lien direct avec le RdP recette tels que les cuves, les réacteurs ou la matière.

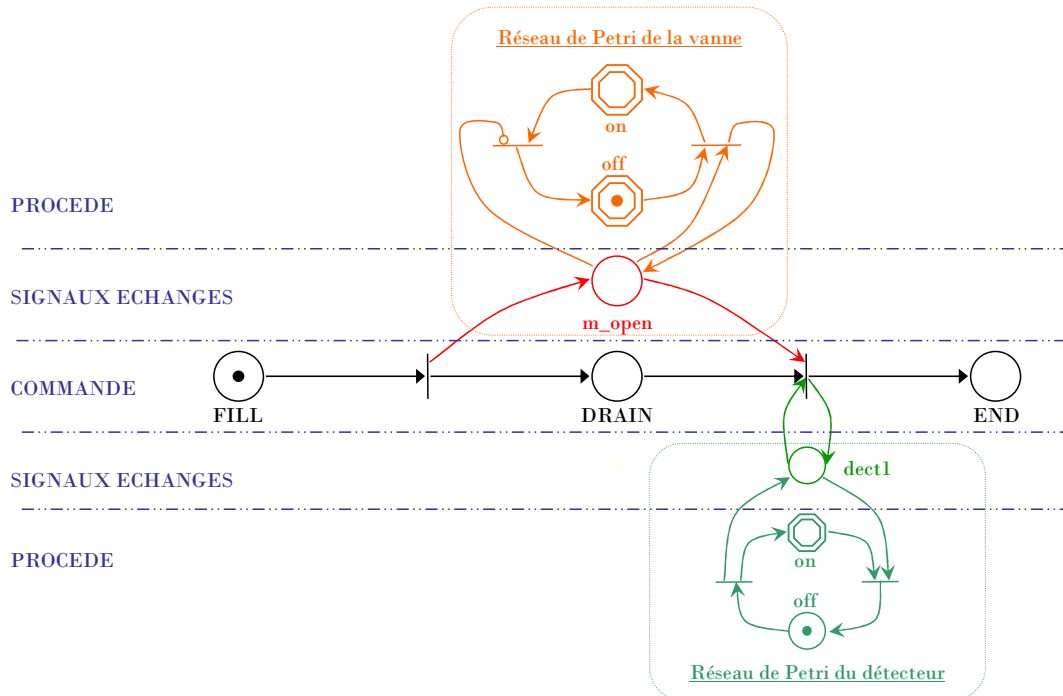


Figure 2.9 Les interactions entre la partie commande et la partie procédé

Les interactions entre RdP recette et RdP procédé sont illustrées sur la figure 2.9. Celle-ci représente une séquence opératoire permettant d'alimenter un réacteur jusqu'à ce qu'un volume désiré soit atteint. L'opération de remplissage du réacteur est commandée par la recette qui envoie un jeton sur la place signal de l'appareil vanne. L'alimentation est maintenue ouverte tant que cette place de commande reste marquée. Afin de détecter l'instant où le volume souhaité de réactif est transféré, un détecteur de niveau est utilisé. Le marquage de la place de l'appareil détecteur indique que le volume de réactif a atteint la valeur cible. La transition est alors tirée. L'absence de jeton sur la place « commande » provoque alors la fermeture de la vanne.

De plus, le marquage d'une place de commande d'un dispositif actif induit en général l'évolution de son RdP, lui-même provoquant éventuellement en cascade l'évolution des dispositifs passifs au travers du réseau constitué par la connexion des différents ports matière ou énergie.

Par souci de clarté, nous utiliserons une vision réduite du réseau de Petri du système. Cette vision ne fait apparaître que les places *signal* des appareils et non leurs réseaux de Petri complets. Ainsi le réseau de Petri du dispositif de la figure 2.9 se résume par ce réseau :

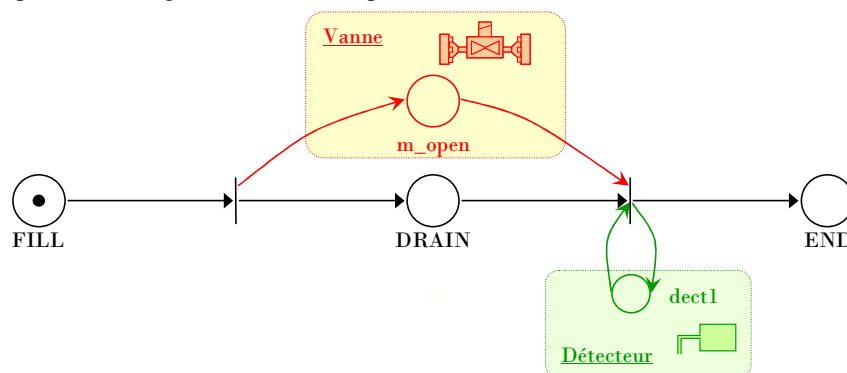


Figure 2.10 La vision du réseau de Petri de commande utilisée par convention

Pour terminer, indiquons que l'évolution des différents modèles est conditionnée par deux types distincts d'événements :

- d'une part, des événements dits externes qui provoquent des commutations commandées. Il s'agit de tous les signaux échangés entre le niveau commande et le niveau procédé (soit les commandes émises depuis le RdP recette afin de piloter les dispositifs actifs, soit à l'occurrence d'un événement d'état (détection d'un seuil) ou temporel. Spécifiés par l'utilisateur, ces événements apparaissent explicitement sur le RdP recette.
- d'autre part, des événements dits intrinsèques (ou encore autonomes) dont l'occurrence dépend uniquement de l'évolution spontanée du procédé. Ces commutations autonomes correspondent au changement d'état de la matière (passage de l'état liquide à l'état liquide/vapeur lorsque la température d'ébullition est atteinte, par exemple) ou d'un dispositif passif. Ceux-ci n'apparaissent donc pas explicitement sur le RdP de la recette (l'utilisateur n'a donc pas à les spécifier) et sont traités exclusivement au sein du modèle de l'entité concernée.

2.5.3.2 La mise en œuvre de macro-place

Nous rappelons aux lecteurs que la norme ISA/SP88 (ou IEC61512-1 : www.isa.org) définit une approche standard pour traiter la complexité du contrôle/commande des procédés batch. Elle est composée actuellement de trois volets dont un concernant la structuration du système de conduite. Au sein de cette partie, quatre vues hiérarchisées sont proposées. La vue qui concerne l'aspect procédural de la recette définit cinq niveaux de description (cf. figure 1.5). Compte tenu de la complexité des recettes considérées, la notion de macro place a été introduite afin de se rapprocher de cette structuration. Il s'agit donc de remplacer une séquence de places/transitions relative à une opération ou une phase par une seule macro-place. Cette séquence est alors délimitée par deux places particulières E et S :

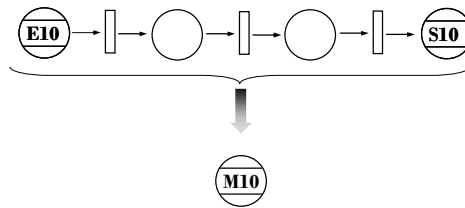


Figure 2.11 La notion de Macro-Place sous PrODHyS

La décomposition présentée précédemment a été adoptée pour la modélisation de la procédure sous PrODHyS. La figure 2.12 présente les modèles UML associées à la procédure. On retrouve ainsi la classe ProcedureRecipe définie comme un ensemble d'opérations (classe OperationRecipe), elles-mêmes définies comme un ensemble de phases (classe PhaseRecipe).

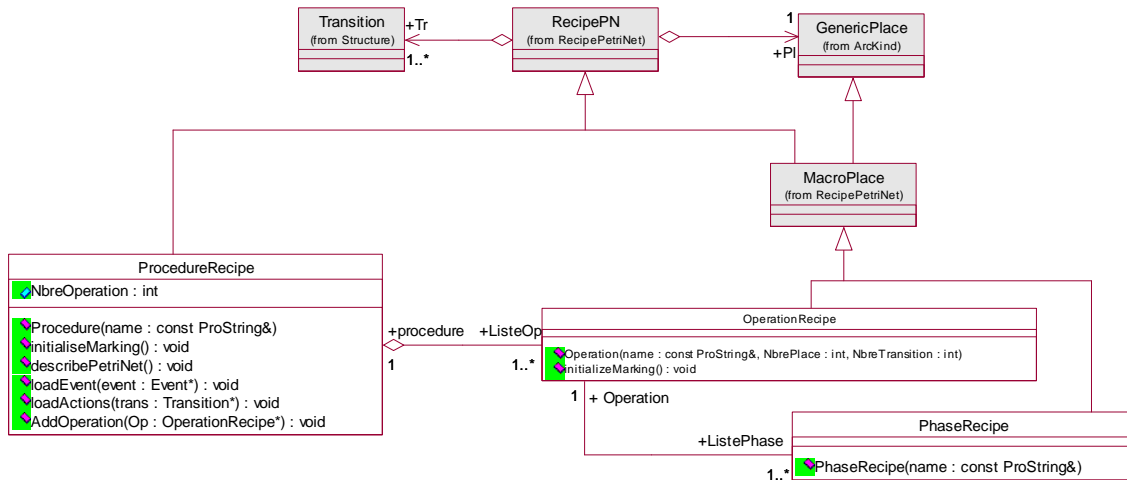


Figure 2.12 La modélisation UML de la procédure

2.6 L'APPLICATION

Afin d'illustrer les concepts fondamentaux de *PrODHyS*, une application est traitée dans ce paragraphe. Elle s'appuie sur le système des vases communicants.

2.6.1 Le procédé étudié

2.6.1.1 La description

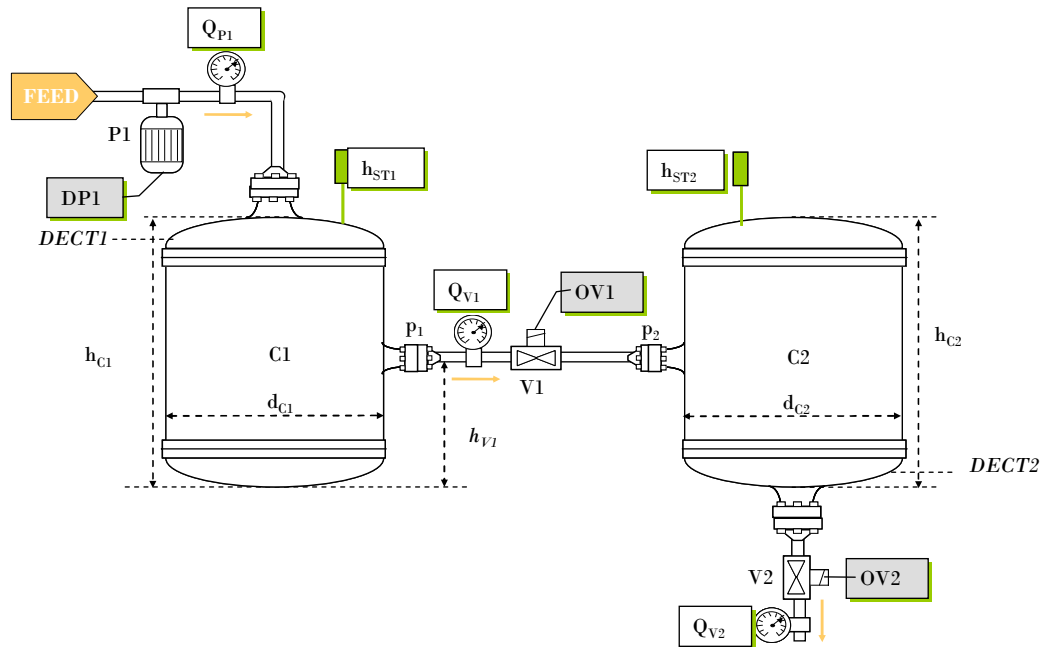


Figure 2.13 Les vases communicants

Le procédé mis en œuvre est décrit sur la figure 2.13. Il comporte deux bacs reliés entre eux par une vanne. Dans cette application, le bac C1 est connecté à une alimentation et un soutirage peut être effectué sur le bac C2. De plus, le procédé est instrumenté de deux détecteurs :

- Le premier contrôle le niveau de liquide dans la cuve C1 et émet une alarme quand le niveau de liquide atteint le niveau haut $h_{11\max}$;
- Le second surveille le niveau de liquide dans la cuve C2 et renvoie une alarme lorsque le niveau de liquide atteint le niveau bas $h_{12\min}$.

Les données techniques sont répertoriées dans le tableau 2.1.

Dans le cadre de cette application, la pression de l'environnement est de 1 atm. Un sens positif est choisi arbitrairement afin de préciser, par la suite, le sens réel de l'écoulement du flux. Les cuves sont similaires et placées à une hauteur identique ($d_{Ci} = 35$ cm, $h_{Ci} = 75$ cm avec $i \in \{1;2\}$). Les ports p_1 et p_2 des cuves C1 et C2 sont élevés de 30 cm ($h_{V1} = 30$ cm) par rapport au fond des cuves.

Le débit de la pompe P1 est de 10 cm³/s.

| | |
|--|----------------------------|
| Hauteur Cuve C1 (h_{C1}) | 75 cm |
| Diamètre Cuve C1 (d_{C1}) | 35 cm |
| Hauteur Cuve C2 (h_{C2}) | 75 cm |
| Diamètre Cuve C2 (d_{C2}) | 35 cm |
| Débit Pompe P1 | 10 cm³/s |
| Section Conduite $\varnothing V1 = \varnothing V2$ | 0,36 cm² |
| Élévation Vanne V1 (h_{V1}) | 20 cm |
| Seuil Maximum DECT1 (h_{l1max}) | 70 cm |
| Seuil Minimum DECT2 (h_{l2min}) | 5 cm |

Tableau 2.1 Les données techniques des vases communicants

2.6.1.2 Le modèle mathématique

Nous allons décrire dans ce point le modèle mathématique caractérisant chaque appareil du procédé.

Ainsi, dans ce système, une vanne V_j est composée d'une entrée et d'une sortie caractérisées respectivement par le lot de données et $(F_{V_j}^{out}, h_{LV_j}^{out}, x_{V_j}^{out})$, où F_f , h_{Lf} et x_f correspondent respectivement au flux, à l'enthalpie liquide et au vecteur de composition liquide du flux f (avec $f = \overset{in}{V_j}$ ou $\overset{out}{V_j}$).

- Dans ces conditions, le modèle de la vanne V_j est décrit de la façon suivante :

- du bilan matière global :

$$F_{V_j}^{in} - F_{V_j}^{out} = 0 \quad (2 - 1)$$

Quand la vanne est fermée, l'équation devient :

$$F_{V_j}^{in} = 0 \quad F_{V_j}^{out} = 0 \quad (2 - 2)$$

- de l'équation de contrainte de flux due aux phénomènes hydrauliques :

$$F_{V_j}^{in} = \frac{S_c}{V_{ml}} \cdot \text{sign}(h_{lin} - h_{lout}) \cdot \sqrt{2 \cdot g \cdot |h_{lin} - h_{lout}|} \quad (2 - 3)$$

Où, S_c est l'aire de la section de conduite, h_l est le niveau de liquide dans la cuve et g est la constante de gravité.

- Ensuite, dans ce système, le modèle de la pompe P_j est composé :

- du bilan matière global :

$$F_{P_j}^{in} - F_{P_j}^{out} = 0 \quad (2 - 4)$$

- de l'équation de contrainte sur le flux :

$$F_{P_j}^{in} = F_{P_j}^{order} \quad (2 - 5)$$

Où, $F_{P_j}^{order}$ est le débit de consigne de la pompe.

- Enfin, une cuve de stockage C_j se décompose d'une entrée et d'une sortie caractérisée respectivement par l'ensemble de données $(F_{C_j}^{in}, h_{LC_j}^{in}, x_{C_j}^{in})$ et $(F_{C_j}^{out}, h_{LC_j}^{out}, x_{C_j}^{out})$ où F_f , h_{Lf} et x_f correspondent respectivement au flux, à l'enthalpie liquide et au vecteur de composition liquide du flux f (avec $f = \begin{smallmatrix} in \\ out \end{smallmatrix}$ ou $\begin{smallmatrix} out \\ in \end{smallmatrix}$). La variable U_l représente la rétention liquide dans la cuve. La variable h_l est le niveau liquide calculé en fonction de U_l , de la section de la cuve S_l et du volume molaire de la phase liquide V_{ml} . Les variables T et P sont respectivement la température et la pression du système. Par conséquent, le modèle de la cuve se traduit de la façon suivante :

- le bilan matière global :

$$\frac{dU_l}{dt} = F_{C_j}^{in} - F_{C_j}^{out} \quad (2 - 6)$$

- les bilans matières partiels :

$$\frac{dU_l \cdot x_i}{dt} = F_{C_j}^{in} \cdot x_{C_j,i}^{in} - F_{C_j}^{out} \cdot x_{C_j,i}^{out} \quad (2 - 7)$$

- le bilan énergétique :

$$\frac{dU_l \cdot h_{Li}}{dt} = F_{C_j}^{in} \cdot h_{LC_j}^{in} - F_{C_j}^{out} \cdot h_{LC_j}^{out} \quad (2 - 8)$$

- le niveau liquide :

$$h_l - \frac{U_l \cdot V_{ml}}{S_l} = 0 \quad (2 - 9)$$

- la contrainte sur l'enthalpie liquide (mh étant le modèle de calcul pour l'enthalpie) :

$$h_L - mh(T, P, x) = 0 \quad (2 - 10)$$

- la contrainte sur le volume molaire liquide (mV_{ml} étant le modèle de calcul pour le volume molaire liquide) :

$$V_{ml} - mV_{ml}(T, P, x) = 0 \quad (2 - 11)$$

- la contrainte sur la pression (mP étant le modèle de calcul pour la pression) :

$$P - mP(p) = 0 \quad (2 - 12)$$

Où, p est l'ensemble des conditions opératoires du système.

L'ensemble de ces modèles sont fusionnés lorsque le modèle de simulation est instancié. La dimension et la structure du système d'équations différentielles algébriques résultant varie alors tout au long de la simulation, en fonction de l'état courant du procédé.

2.6.2 La modélisation du procédé

2.6.2.1 La décomposition

Les mécanismes offerts par l'approche objet permettent de construire des appareils dits complexes obtenus par spécialisation et composition d'appareils dits élémentaires. La figure 2.14 illustre le

processus de décomposition adopté pour représenter le procédé de l'exemple. La décomposition topologique du procédé est illustrée sur la figure 2.7. Le modèle comporte :

- deux cuves (de type StorageTank) ;
- une alimentation matière (de type MaterialFeed) délivrant un débit de $2 \text{ cm}^3/\text{s}$;
- une vanne V2 (de type OneWayValve) pour contrôler le soutirage du bac C2 ;
- une vanne à inversion de flux V1 (de type TwoWayValve) ;
- deux détecteurs de niveau (de type LevelDetector) indiquant respectivement un dépassement du niveau haut dans la cuve C1 et du niveau bas dans la cuve C2 ;
- l'environnement (de type Surroundings).

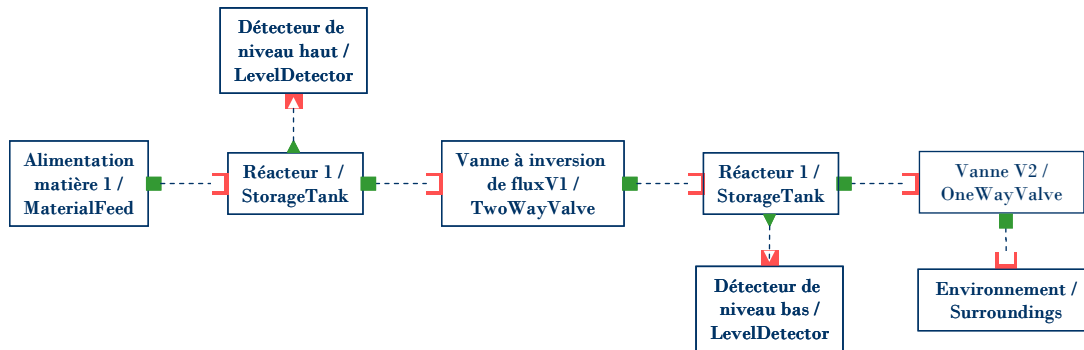


Figure 2.14 Le schéma bloc associé au système des vases communicants

Le procédé est ainsi *décomposé* jusqu'à n'obtenir que des appareils élémentaires spécifiques. De plus, chaque appareil élémentaire spécifique résulte de la *spécialisation*, à des degrés différents, d'entités abstraites et génériques (cf. figure 2.15).

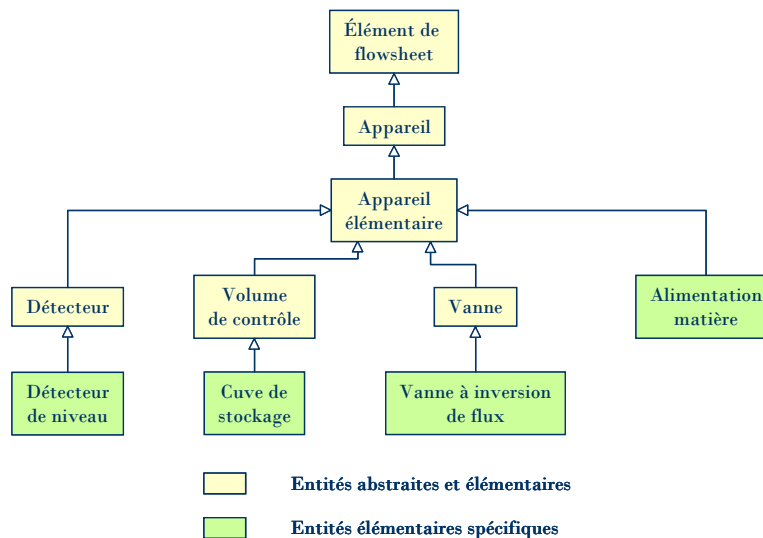


Figure 2.15 Le processus de spécialisation d'appareils élémentaires

2.6.2.2 La modélisation des appareils

2.6.2.2.1 L'alimentation matière

Une *alimentation matière instantanée*, décrite par la classe `MaterialFeed`, désigne un appareil capable de fournir une certaine quantité de matière. Elle est caractérisée par un flux (`maxFlow`) correspondant au débit maximum susceptible d'être délivré. Son comportement est par ailleurs décrit par le réseau de Petri de la figure 2.16. Les conditions et actions associées au réseau de Petri de l'alimentation sont résumées dans le tableau 2.2.

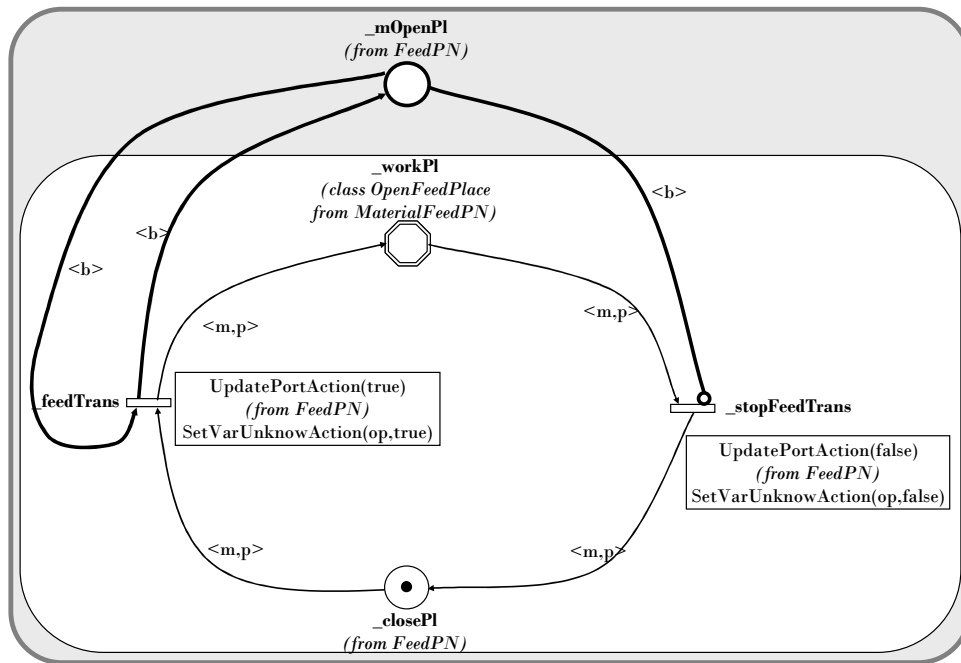


Figure 2.16 Le réseau de Petri de l'alimentation instantanée

| Transition | Evénement | Action |
|-----------------------|-----------|--|
| _feedTrans | | <p>Comportement hérité de FeedPN \Leftrightarrow UpdatePortAction(true) : lorsque la place de commande est marquée, l'action permet l'activation du port de sortie associé à l'alimentation matière. L'activation de ce port conduit à l'activation du port d'entrée auquel il est connecté. L'alimentation passe alors dans l'état actif (place _workPI)</p> <p>Comportement spécifique à MaterialFeedPN \Leftrightarrow SetVarUnknowAction(Op,true) : l'action donne le statut de variable à la grandeur <i>Op</i> afin de pouvoir l'intégrer dans une équation du modèle.</p> |
| _stopFeedTrans | | <p>Comportement hérité de FeedPN \Leftrightarrow UpdatePortAction(false) : lorsque la place de commande n'est plus marquée, l'action permet la désactivation du port de sortie associé à l'alimentation matière. La désactivation de ce port conduit à la désactivation du port d'entrée auquel il est connecté. L'alimentation passe alors dans l'état inactif (place _closePI)</p> <p>Comportement spécifique à MaterialFeedPN \Leftrightarrow SetVarUnknowAction(Op,false) : l'action donne le statut de paramètre à la grandeur <i>Op</i> afin qu'elle apparaisse dans le second membre d'une équation du modèle.</p> |

Tableau 2.2 Les conditions et actions associées aux transitions du réseau de Petri de l'alimentation instantanée

2.6.2.2.2 Le détecteur de niveau

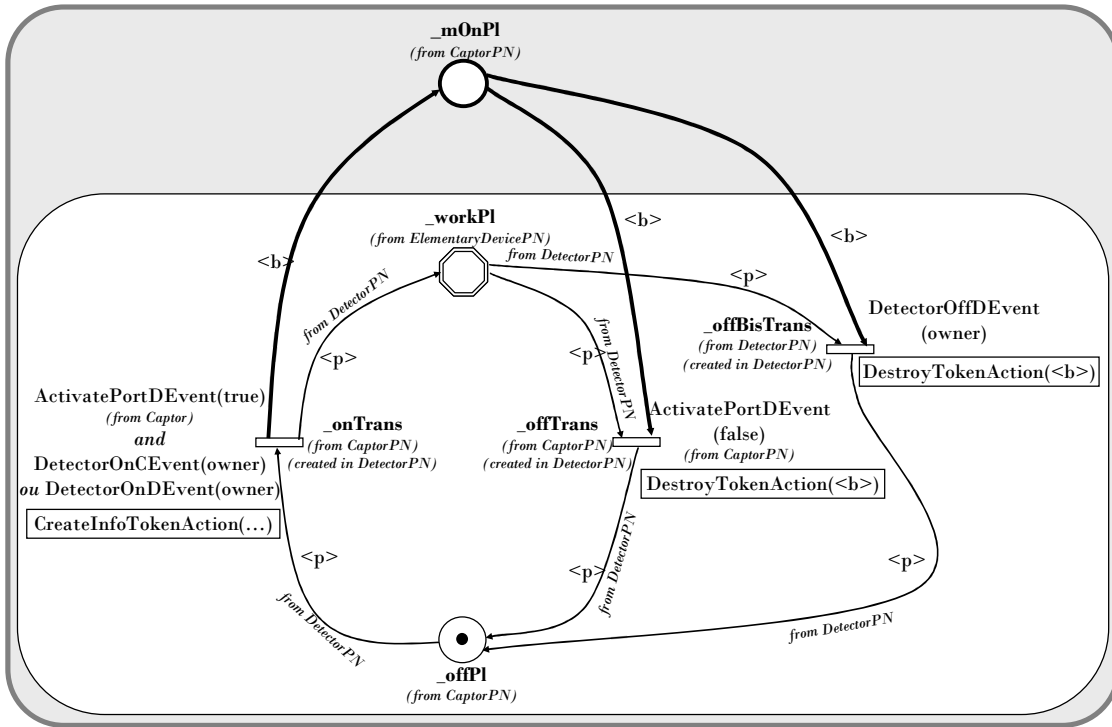


Figure 2.17 Le réseau de Petri d'un détecteur

| Transition | Événement | Action |
|---------------------|---|--|
| _onTrans | <p>Comportement hérité de CaptorPN ⇔ Teste si le port d'entrée de communication du détecteur est actif. Celui-ci est actif si le port de sortie auquel il est connecté est actif. Ce dernier est actif si la grandeur captée existe.</p> <p>Comportement spécifique de DetectorPN ⇔ Teste si la condition de détection (==, <,>,<=,>=) est vérifiée.</p> <p>Si ces 2 événements apparaissent, alors le détecteur passe dans l'état actif (place _workPI)</p> | Création du jeton binaire permettant de marquer la place _mOnPI |
| _offTrans | <p>Comportement hérité de CaptorPN ⇔ Teste si le port d'entrée de communication du détecteur est inactif. Celui-ci est inactif si le port de sortie auquel il est connecté est inactif. Ce dernier est inactif si la grandeur captée n'existe pas ou plus. Si l'événement apparaît, alors le détecteur passe dans l'état inactif (place _offPI)</p> | Destruction du jeton binaire provenant de la place _mOnPI |
| _offBisTrans | <p>Comportement spécifique de DetectorPN ⇔ Teste si la condition de détection (==, <,>,<=,>=) n'est plus vérifiée.</p> <p>Si l'événement apparaît, alors le détecteur passe dans l'état inactif (place _offPI)</p> | Destruction du jeton binaire provenant de la place _mOnPI |

Tableau 2.3 Les conditions et actions associées aux transitions du réseau de Petri du détecteur

Le détecteur a été conçu pour suivre une grandeur physique du procédé. Cependant, cette grandeur n'est pas transmise en tant que telle à la recette. Elle est évaluée par rapport à une valeur seuil

spécifiée au préalable par le niveau commande. Le détecteur peut ainsi transmettre le résultat de cette évaluation (qui se traduit par une expression logique) en émettant un signal binaire sur sa place *signal*. La communication entre la recette et le procédé se fait alors uniquement en termes de messages binaires.

Le comportement du détecteur est décrit par la classe *DetectorPN* (cf. figure 2.17). Des événements sont ajoutés afin de tester la condition logique (équivalence, supériorité, infériorité, etc.) sur les transitions *_onTrans* et *_offtrans*. Le tableau 2.3 résume ces différents points.

2.6.2.2.3 La cuve de stockage

La cuve de stockage est un volume de contrôle particulier caractérisé par une forme géométrique spécifique et une hauteur de liquide. La géométrie de la cuve, et plus précisément, la géométrie du fond de la cuve, est définie dans la configuration associée à l'appareil ; elle peut être : plate, hémisphérique, ellipsoïdal, torisphérique, sphérique, toriconique, conique, etc. Selon le modèle choisi, la méthode *VolumeFormula()*, redéfinie dans la classe *StorageTank*, détermine, en fonction des paramètres géométriques spécifiés, le volume de la cuve. De plus, la hauteur de liquide permet d'une part, de calculer l'aire d'échange entre la cuve et l'extérieur et d'autre part, de déterminer les pressions hydrauliques associées aux ports hydrauliques du bac. Une variable supplémentaire a été introduite au niveau du bac ; il s'agit de la hauteur de liquide. Celle-ci est associée à la place différentielle *Work*.

Le comportement de ce type de bac est décrit par le réseau de Petri suivant :

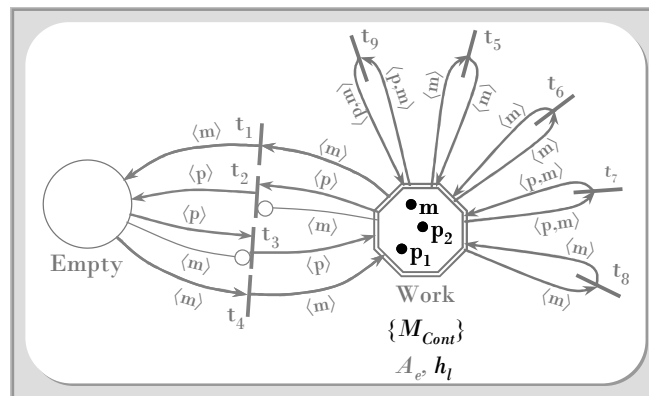


Figure 2.18 Le réseau de Petri de la cuve de stockage

De manière générale, le réseau de Petri associé à la cuve de stockage permet de gérer :

- le transfert des jetons de la place *Work* vers la place *Empty* lorsque la matière disparaît de l'appareil (transitions t_1, t_2),
- le transfert des jetons de la place *Empty* vers la place *Work* lorsque la matière apparaît dans l'appareil (transitions t_3, t_4),
- la désactivation automatique des ports de sortie matière portant la phase amenée à disparaître (transitions t_5, t_6, t_7, t_8),
- l'activation du port de sortie matière sensibilisant la transition lorsque celui-ci est supposé s'activer automatiquement en présence d'une phase de même nature que celle qu'il porte (transition t_9).

Les variables qui décrivent l'état de la cuve de stockage sont réparties d'une part, dans les jetons du réseau et d'autre part, au niveau du réseau lui-même. En effet, les variables concernant les propriétés qualitatives et quantitatives des phases sont portées par les jetons *Phase* ; les variables de *pression* et *température* sont portées par le jeton *Matière* ; les variables de *flux* et éventuellement de *pression* et *d'énergie hydrauliques* sont portées par les jetons *Port*. En revanche, les variables décrivant les pertes thermiques éventuelles et l'aire d'échange sont spécifiques à la cuve de stockage. Les conditions et actions associées aux transitions sont décrites dans le tableau 2.4.

L'état vide et occupé de l'appareil constitue une information intéressante surtout en gestion de production où le problème de l'occupation des ressources est important. C'est pourquoi, les places *Empty* et *Work* ont été créées afin de représenter respectivement l'état *vide* et l'état *occupé* de la ressource. La place *Empty* est une place discrète à laquelle aucune équation n'est associée. La place *Work* est une place différentielle pour laquelle un système algèbro-différentiel a été défini.

| | Conditions | Actions |
|----------------|--|---|
| t ₁ | place marquée de la matière = place Unknown | ∅ |
| t ₂ | ∅ | ∅ |
| t ₃ | ∅ | ∅ |
| t ₄ | place marquée de la matière ≠ place Unknown | ∅ |
| t ₅ | Phase liquide disparaît pas de ports d'entrée matière liquide actifs somme des débits des ports de sortie liquide actifs = 0 | phase à supprimer ← phase liquide |
| t ₆ | Phase gazeuse disparaît pas de ports d'entrée matière vapeur actifs somme des débits des ports de sortie vapeur actifs = 0 | phase à supprimer ← phase vapeur |
| t ₇ | le port p sensibilisant la transition est un port de sortie matière actif portant la phase à supprimer | désactivation du port p |
| t ₈ | il n'existe plus de ports de sortie matière actifs portant la phase à supprimer | ∅ |
| t ₉ | il existe une phase dont le type est identique à celui du port p encore inactif sensibilisant la transition le port p est supposé s'activer automatiquement | activation du port p franchissant la transition |

Tableau 2.4 Les conditions et actions associées aux transitions du réseau de Petri de la cuve de stockage

Ainsi, lorsque la matière devient absente de l'appareil, le jeton *Matière* associé est transféré sur la place *Empty* via la transition t₁. L'absence de jeton *Matière* sur la place *Work*, détectée au niveau de la transition t₂, provoque alors le transfert de tous les jetons *Port* de la place *Work* vers la place *Empty*. Au contraire, lorsque la matière apparaît dans la cuve, les jetons sont transférés selon le même mécanisme de la place *Empty* vers la place *Work*. Le nombre de ports associés à l'appareil n'étant *a priori* pas connu, l'utilisation d'arcs inhibiteurs est ici très intéressante puisqu'ils permettent de tester l'absence du jeton *Matière* sur les places situées en amont des transitions t₂ et t₃.

Si la phase disparaît, la transition t₅ (pour la phase liquide) ou t₆ (pour la phase vapeur) est franchie. La transition t₇ est validée tant qu'il existe un port de sortie matière *actif* transportant la phase à supprimer (l'action associée consiste alors à la désactiver). Lorsqu'il ne reste plus de ports de sortie actifs transportant cette phase, la transition t₈ est alors franchie.

De plus, si l'utilisateur l'a au préalable spécifié, l'apparition d'une phase au sein du volume de contrôle peut générer l'activation automatique des ports de sortie exportant cette phase. Ce cas se présente, par exemple, lorsque le point de bulle est atteint dans une cuve ouverte contenant initialement un système liquide. La transition t_9 a ainsi été définie dans ce contexte.

2.6.2.2.4 La vanne

La fonction d'une vanne est d'agir sur un débit en le réglant manuellement ou par l'intermédiaire d'un système de régulation. La vanne entre ainsi dans la catégorie des appareils *actifs*.

Le modèle RdPDO qui lui est associé est alors caractérisé par une place de commande (place `mOpen`) pilotant l'ouverture ou la fermeture de la vanne. Les conditions et actions associées au réseau de Petri de la vanne sont résumées dans le tableau 2.5.

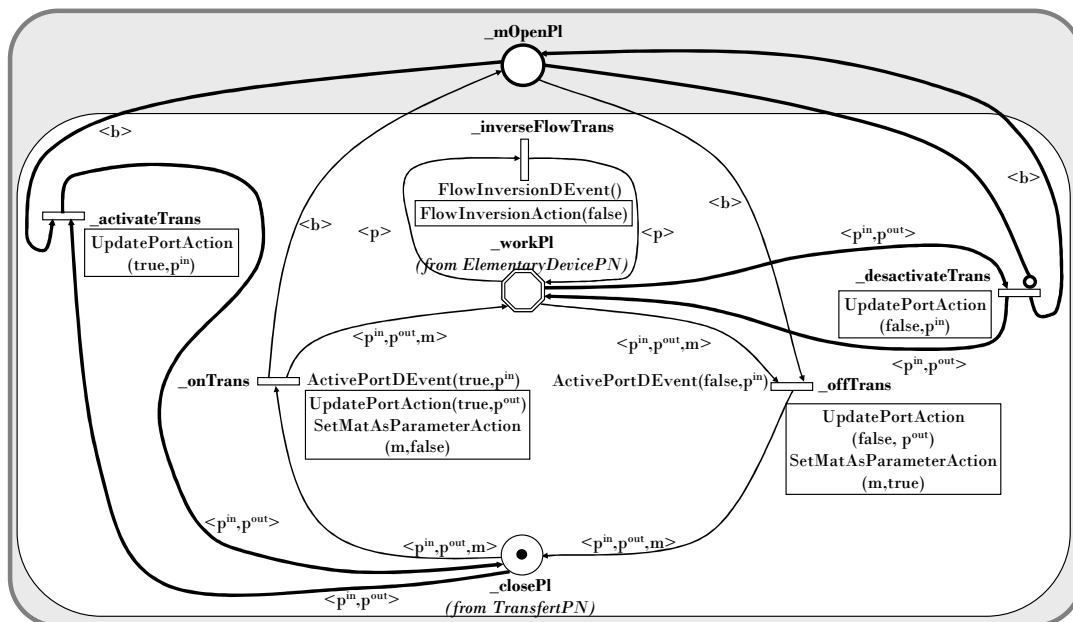


Figure 2.19 Le réseau de Petri de la vanne

Le tir des transitions `_activateTrans` et `_desactivateTrans` est effectif dès qu'une commande est émise depuis le niveau *commande*. Elles s'accompagnent respectivement de l'activation et la désactivation du port d'entrée.

De plus, le phénomène d'inversion de flux est fréquent. Or à ce niveau, aucun élément ne précise si la vanne est anti-retour ou si elle autorise une inversion de flux. Dans ce contexte, deux vannes spécifiques sont définies : la *vanne anti-retour* décrite par la classe `OneWayValve` et la *vanne à inversion de flux* décrite par la classe `TwoWayValve`.

L'état ouvert de la vanne à inversion de flux est désormais décrit par deux places différentielles. La première, `Open`, correspond à l'état ouvert de la vanne lorsque l'écoulement se fait dans le sens positif. La seconde, `OpenInv`, correspond à l'état ouvert de la vanne lorsque l'écoulement se fait dans le sens contraire au sens positif. Ces deux places proposent une équation supplémentaire contraignant la pression à celle de l'appareil recevant le flux de la matière. Mais celle-ci dépend maintenant du sens de l'écoulement.

Dans le premier cas, il s'agit de la pression de l'appareil situé en aval du port de sortie ; dans le second cas, il s'agit de la pression de l'appareil situé en amont du port d'entrée.

Lorsqu'une commande d'ouverture est lancée (la vanne étant initialement fermée), les transitions t_3 ou t_5 sont sensibilisées, mais en fonction du sens de l'écoulement, seule l'une des deux est franchissable.

| Transition | Événement | Action |
|--------------------------|---|---|
| _onTrans | Hérité de TransfertPN \Leftrightarrow ActivePortDEvent(true,p ⁱⁿ) Retourne la valeur true si le port d'entrée p ⁱⁿ est actif (c.a.d., si la place ActivePl du port est marquée) | Hérité de TransfertPN \Leftrightarrow UpdatePortAction(true,p ^{out}) Cette action permet l'activation du port de sortie p ^{out} associé à cet appareil. L'activation de ce port conduit à l'activation du port d'entrée auquel il est connecté (en cascade) SetMatAsParameterAction(m,false) Spécifie que les attributs de la matière deviennent des variables du modèle associé |
| _offTrans | Hérité de TransfertPN \Leftrightarrow ActivePortDEvent(false,p ⁱⁿ) Retourne la valeur true si le port d'entrée p ⁱⁿ est inactif (c.a.d., si la place ActivePl du port n'est plus marquée) | Hérité de TransfertPN \Leftrightarrow UpdatePortAction(false,p ^{out}) Cette action permet la désactivation du port de sortie p ^{out} associé à cet appareil. La désactivation de ce port conduit à la désactivation du port d'entrée auquel il est connecté (en cascade) SetMatAsParameterAction(m,true) Spécifie que les attributs de la matière deviennent des paramètres du modèle associé |
| _activateTrans | | UpdatePortAction(true,p ⁱⁿ) Cette action permet l'activation du port d'entrée p ⁱⁿ associé à cet appareil. Ceci permet ensuite le franchissement de la transition _onTrans |
| _desactivateTrans | | UpdatePortAction(false,p ⁱⁿ) Cette action permet la désactivation du port d'entrée p ⁱⁿ associé à cet appareil. Ceci permet ensuite le franchissement de la transition _offTrans |
| _inverseFlowTrans | FlowInversionDEvent() True si l'attribut FlowInversion a été mis à true, indiquant l'occurrence d'une inversion de flux | FlowInversionAction(false) Cette action permet replacer l'attribut FlowInversion à false (pour éviter de tourner en boucle sur la transition) |

Tableau 2.5 Les conditions et actions associées aux transitions du réseau de Petri de la vanne

Lorsque le sens de l'écoulement est positif (le flux entre dans la vanne par le port d'entrée), la transition t_3 est tirée et l'action associée consiste à activer le port d'entrée de la vanne. Au contraire, lorsque le sens de l'écoulement est négatif (le flux entre dans la vanne par le port de sortie), la transition t_5 est tirée et l'action associée consiste à activer le port de sortie de la vanne. La vanne n'autorisant aucune accumulation de matière, les transitions t_1 et t_7 se chargent d'activer respectivement le port de sortie et le port d'entrée de la vanne. Enfin, les conditions associées aux transitions t_9 et t_{10} surveillent les états actifs des ports d'entrée et de sortie de la vanne (marquage de la place Active ou ActiveInv). En effet, un changement de configuration au niveau des ports (passage de l'état Active à l'état ActiveInv ou inversement) implique un changement de configuration de la vanne qui passe de l'état Open à OpenInv ou de l'état OpenInv à Open.

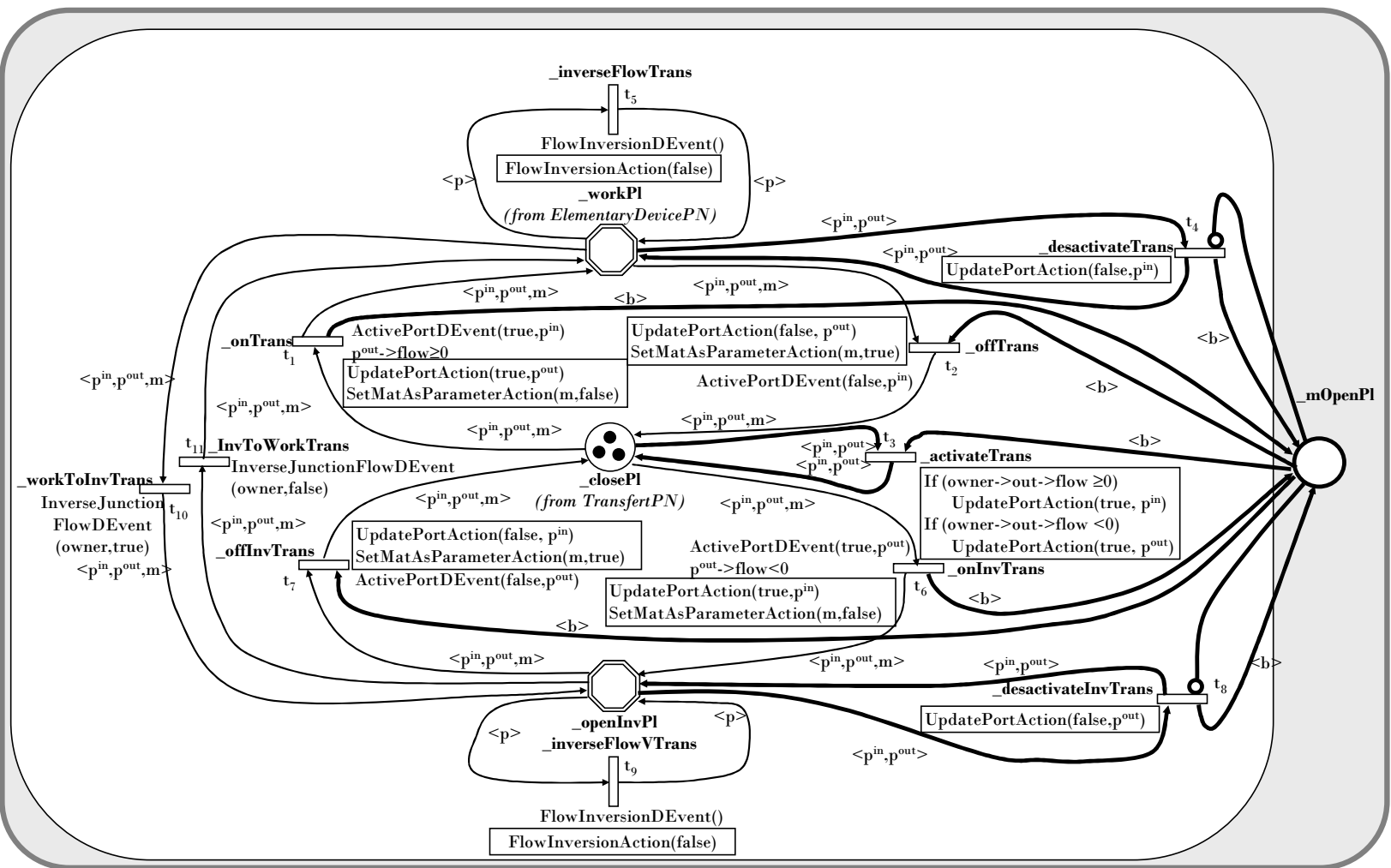


Figure 2.20 Le réseau de Petri de la vanne à inversion de flux

| Transition | Événement | Action |
|--|---|--|
| t_1 \Leftrightarrow _onTrans | Hérité de TransfertPN \Rightarrow ActivePortDEvent(true,p ⁱⁿ) p ^{out} ->flow \geq 0 Franchit cette transition si le débit au niveau du port de sortie de la vanne est positif | Hérité de TransfertPN \Rightarrow UpdatePortAction(true,p ^{out}) Hérité de ValvePN \Rightarrow SetMatAsParameterAction(m,false) |
| t_2 \Leftrightarrow _offTrans | Hérité de TransfertPN \Rightarrow ActivePortDEvent(false,p ⁱⁿ) | Hérité de TransfertPN \Rightarrow UpdatePortAction(false,p ^{out}) Hérité de ValvePN \Rightarrow SetMatAsParameterAction(m,true) |
| t_3 \Leftrightarrow _activateTrans | | If (owner->out->flow \geq 0) UpdatePortAction(true, p ⁱⁿ) If (owner->out->flow <0) UpdatePortAction(true, p ^{out}) Si le sens du flux du port de sortie de la vanne est positif (resp. négatif), activation du port d'entrée (resp. de sortie) |
| t_4 \Leftrightarrow _desactivateTrans | | Hérité de ValvePN \Rightarrow UpdatePortAction(false,p ⁱⁿ) |
| t_5 \Leftrightarrow _inverseFlowTrans | Hérité de ValvePN \Rightarrow FlowInversionDEvent() | Hérité de ValvePN \Rightarrow FlowInversionAction(false) |
| t_6 \Leftrightarrow _onInvTrans | ActivePortDEvent(true,p ^{out}) p ^{out} ->flow<0 Franchit cette transition si le débit au niveau du port de sortie de la vanne est positif | UpdatePortAction(true,p ⁱⁿ) Cette action permet l'activation du port d'entrée p ⁱⁿ associé à cet appareil. SetMatAsParameterAction(m,false) Spécifie que les attributs de la matière deviennent des variables du modèle associé |
| t_7 \Leftrightarrow _offInvTrans | ActivePortDEvent(false,p ^{out}) | UpdatePortAction(false,p ⁱⁿ) Cette action permet la désactivation du port de sortie p ⁱⁿ associé à cet appareil. SetMatAsParameterAction(m,true) Spécifie que les attributs de la matière deviennent des paramètres du modèle associé |
| t_8 \Leftrightarrow _desactivateInvTrans | | UpdatePortAction(false,p ^{out}) Cette action permet la désactivation du port d'entrée p ^{out} associé à cet appareil. |
| t_9 \Leftrightarrow _inverseFlowVTrans | FlowInversionDEvent() True si l'attribut FlowInversion a été mis à true, indiquant l'occurrence d'une inversion de flux | FlowInversionAction(false) Cette action permet de replacer l'attribut FlowInversion à false (pour éviter de tourner en boucle sur la transition) |
| t_{10} \Leftrightarrow _workToInvTrans | InverseJunctionFlowDEvent(owner,true) Indique que le flux dans la jonction est inversé | |
| t_{11} \Leftrightarrow _InvToWorkTrans | InverseJunctionFlowDEvent(owner,true) Indique que le flux dans la jonction est en sens normal | |

Tableau 2.6 Les conditions et actions associées aux transitions du réseau de Petri de la vanne à inversion de flux

2.6.3 La modélisation de la recette

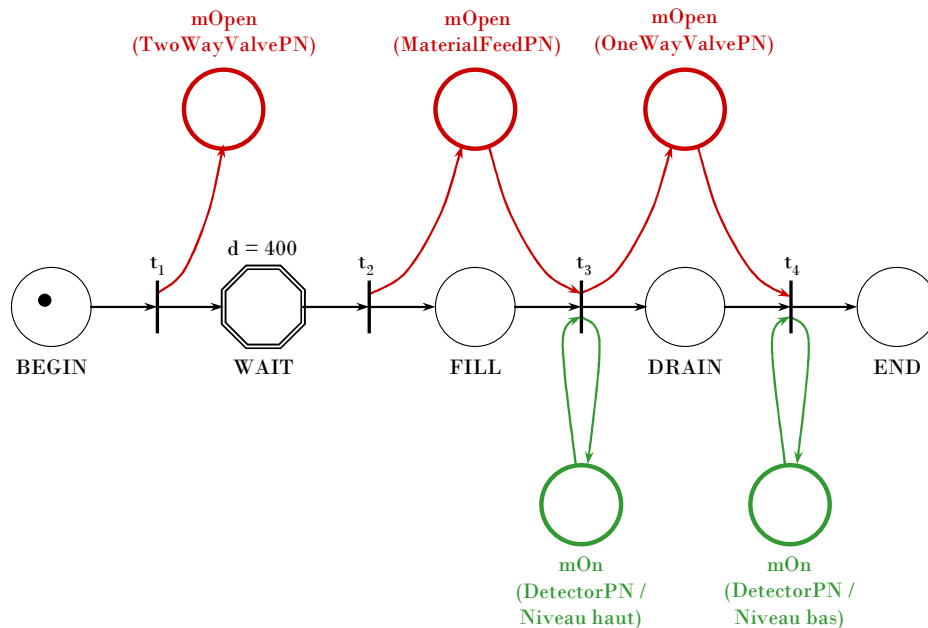


Figure 2.21 La recette associée au système des vases communicants

Dans le cadre de cette application, la séquence opératoire à réaliser est décrite sur la figure 2.21. Initialement, chaque bac contient un mélange binaire eau-méthanol et l'alimentation, le soutirage ainsi que la vanne sont supposés fermés. La hauteur de liquide contenu dans le bac 1 est inférieure à celle du port p_1 ; au contraire, celle du liquide contenu dans le bac 2 est supérieure à celle du port p_2 . Elle est composée de trois étapes principales :

- La première étape consiste à ouvrir la vanne à inversion de flux reliant les deux cuves. La recette envoie alors un jeton binaire sur la place de commande $mOpen$ du réseau de Petri associé à la vanne. Dès lors, un flux apparaît entre les deux bacs. Si, à l'initialisation, la hauteur de liquide de la cuve C2 est supérieure à celle de cuve C1, alors le flux s'écoule du bac 2 vers le bac 1, et inversement, si, à l'initialisation, la hauteur de liquide de la cuve C2 est inférieure à celle de la cuve C1, alors le flux s'écoule du bac C1 vers le bac C2. Cette configuration est maintenue pendant un temps égal à la temporisation d associée à la place différentielle WAIT.
- Lorsque la temporisation est écoulée, la cuve C1 est alimentée en eau jusqu'à ce que le niveau de liquide maximal, h_{1max} , soit atteint. L'ouverture de l'alimentation est pilotée par la recette qui émet un signal binaire au niveau de la place de commande $mOpen$ du réseau de Petri associé à l'alimentation ; l'occurrence de l'événement « niveau haut atteint » est en revanche signalée par le détecteur de niveau haut qui envoie un jeton binaire sur sa place d'information mOn . Pendant cette période, le sens de l'écoulement varie en fonction des hauteurs de liquide présentes dans les deux bacs.
- Le dépassement du niveau haut dans la cuve C1 entraîne le tir de la transition t_3 . Ceci s'accompagne d'une part, de la fermeture de l'alimentation (la place $mOpen$ de l'alimentation n'est plus marquée) et d'autre part, de l'introduction d'un soutirage au niveau du bac C2 (marquage de la place $mOpen$ de la

vanne agissant sur la vidange du bac C2). La place DRAIN décrit alors la vidange partielle des deux bacs. Au niveau de la cuve C1, la vidange cesse lorsque la hauteur de liquide h_{11} atteint celle du port h_p ; en effet, la différence de pression aux bornes de la vanne s'annule et le flux qui la traverse disparaît. La vidange du bac 2 se poursuit jusqu'à ce que le détecteur de niveau bas informe la recette du dépassement du seuil minimal.

2.6.4 La simulation

Dans le cadre de cette application, le fluide utilisé est un binaire eau-méthanol liquide. Les conditions initiales sont répertoriées dans le tableau 2.7.

| | Cuve C1 | Cuve C2 | Alimentation |
|-----------------------|---------|---------|--------------|
| T (K) | 298,15 | 298,15 | 298,15 |
| x_{eau} | 0,8 | 0,3 | 0,5 |
| $x_{\text{méthanol}}$ | 0,2 | 0,7 | 0,5 |
| U_1 (mol) | 990 | 400 | - |
| h_1 (cm) | 23,26 | 14,07 | - |

Tableau 2.7 Les conditions initiales de la simulation des vases communicants

Analysons les résultats obtenus par simulation :

Le graphe de la figure 2.22 représente l'évolution du flux qui transite à travers la vanne V1 (F_{V_1}), l'évolution du flux qui transite à travers la vanne V2 (F_{V_2}), l'évolution du flux de l'alimentation matière de la cuve C1 (F_F) et l'évolution des niveaux de liquide dans les deux cuves (h_{11} et h_{12}) dans chacune des deux cuves. Cette simulation met en exergue les phénomènes hydrauliques intervenant dans ce système. Ainsi, les hauteurs de liquide et le flux qui traverse la vanne V1 sont intimement liés. Par exemple, le changement de signe de la pente de la courbe associée au flux F_{V_1} , à la date 400 min, entraîne instantanément un changement de pente des hauteurs de liquides h_{11} et h_{12} .

Le graphe de la figure 2.23 représente l'évolution de la fraction molaire en eau et méthanol dans les cuves C1 et C2. Il met l'accent sur les phénomènes d'inversion de flux et en particulier, souligne la capacité du modèle à choisir, de façon adéquate, la phase à exporter. Ainsi, initialement, la fraction molaire en méthanol dans la cuve C2 est de 0,7. Lorsque la vanne V1 s'ouvre, un flux apparaît de la cuve C1 vers la cuve C2 ($h_{11} > h_{12}$). La fraction molaire en méthanol au sein de la cuve C2 diminue alors (cf. figure 2.23). Ceci est dû à une concentration en méthanol au sein de la cuve C1 ($x_{\text{méthanol}} = 0,2$) inférieure à celle dans la cuve C2 ($x_{\text{méthanol}} = 0,7$).

2.6 L'application

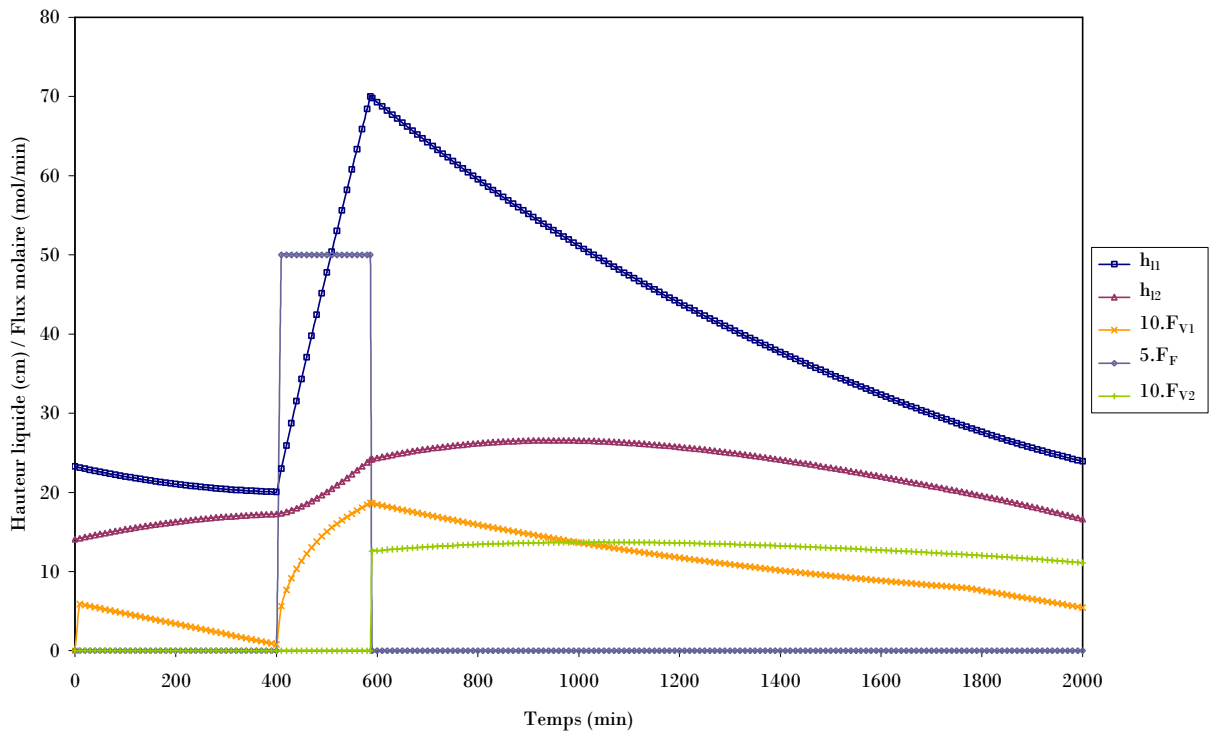


Figure 2.22 Les évolutions du flux et des niveaux de liquide dans les bacs C1 et C2

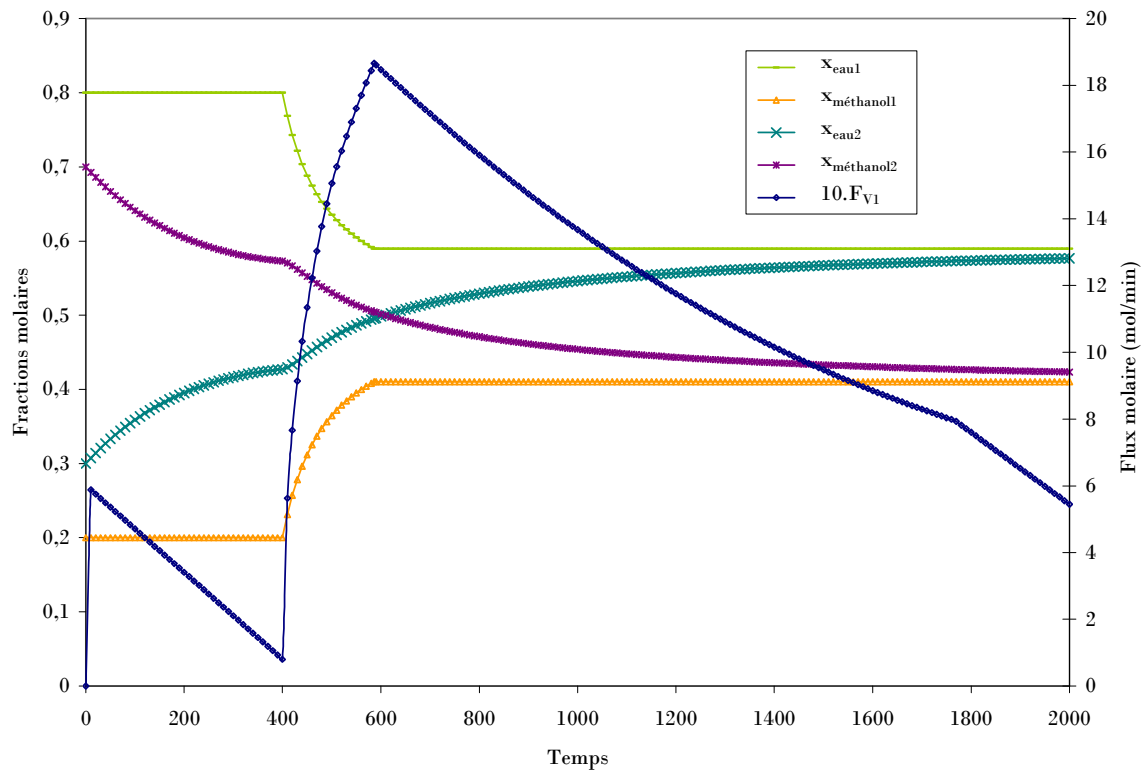


Figure 2.23 Les évolutions des fractions molaires liquide en eau et méthanol dans les bacs C1 et C2

2.7 LA CONCLUSION

Afin de bien situer l'environnement dans lequel est mise en œuvre l'approche de surveillance développée dans ce manuscrit, ce chapitre s'est proposé de dresser un état de l'existant. Cette description reste néanmoins assez générale et le lecteur pourra consulter les documents référencés pour plus d'informations.

L'approche orientée objet a largement été évoquée dans ce chapitre. Elle apporte de nombreux avantages en termes de qualité logiciel (extensibilité, réutilisabilité, flexibilité), mais surtout en termes de modélisation grâce à une description hiérarchique à la fois abstraite et proche de la réalité. Elle permet de représenter plus simplement des entités de plus en plus complexes en masquant progressivement cette complexité.

Dans cet objectif, *PrODHyS* permet tout d'abord une représentation rigoureuse et explicite des composantes continues et discrètes ; en effet, les réseaux de Petri engendrent une description efficace des contraintes de synchronisation, de parallélisme et de séquençement et les équations différentielles algébriques conduisent à une représentation précise et fiable de la dynamique continue (à condition de disposer d'un solveur suffisamment puissant notamment pour la recherche et la gestion des événements ainsi que pour le calcul des conditions initiales).

L'« approche combinée étendue » permet une intégration sur plusieurs niveaux des réseaux de Petri et des objets. Ceci permet de réduire la taille du modèle, de découpler la complexité du système, d'organiser l'information (autour des jetons notamment), d'améliorer la lisibilité du modèle et par conséquent d'en faciliter sa réutilisation.

PrODHyS fournit des composants logiciels destinés à modéliser et à simuler plus spécifiquement les procédés chimiques. Par ailleurs, la modélisation de nombreux procédés nécessite la prise en compte de phénomènes hybrides. L'utilisation d'un modèle à haut niveau d'abstraction associé à des méthodes d'intégration numérique performantes permet de construire un simulateur dynamique hybride numériquement robuste.

Dans le dernier point de ce chapitre, les potentialités de *PrODHyS* ont été illustrées à travers la modélisation et la simulation d'un procédé semi-continu. Ces résultats permettent d'envisager l'utilisation de ce modèle de simulation comme modèle de référence dans un système de surveillance.

PARTIE 2 :
L'ÉTAT DE L'ART

CHAPITRE 3 LES TECHNIQUES DE SURVEILLANCE

Par un souci de clarté, ce chapitre propose tout d'abord de définir la terminologie employée dans ce mémoire pour les études de détection et diagnostic de fautes, afin de positionner le cadre de notre travail. Dans une seconde section, un état de l'art des différentes approches de détection et de diagnostic est établi. Une analyse comparative de ces approches est ensuite présentée. Enfin, nous soulignons la théorie sur laquelle repose l'approche de diagnostic de défauts proposée dans cette thèse.

Les enjeux économiques actuels imposent une croissance continue de la productivité. Dans de telles circonstances, toute défaillance est considérée comme nuisible. Si les Politiques de Maintenance Préventive (TPM) permettent de réduire considérablement ces aléas, elles ne peuvent pas les éliminer complètement. De ce fait, la conduite du procédé doit donc assurer une surveillance permanente, afin de détecter au plus tôt ces défaillances. Cela nécessite alors la mesure de nombreuses variables process caractéristiques, délivrées par des capteurs. Quel que soit le composant incriminé, l'occurrence d'une défaillance entraîne souvent un comportement non désiré du procédé et donc une détérioration de ses performances. De ce fait, le module de surveillance doit être capable de traiter aussi bien les défauts du système physique (fuites, pertes thermiques, etc.) que ceux des appareils de contrôle/commande (actionneurs, capteurs, etc.). De plus, la complexité et la taille des systèmes industriels actuels induisent la surveillance d'un nombre croissant de variables process et rendent ainsi difficile le travail des opérateurs. De ce fait, la conception d'un système de supervision couplé à un outil d'aide à la décision apparaît judicieuse.

Dans ce contexte, depuis plusieurs années, le diagnostic de défauts est un vaste domaine qui fait l'objet d'un engouement particulier auprès de la communauté scientifique et des industriels. Ceci est dû à la volonté de ne plus subir les pannes mais de les maîtriser. La littérature cite de nombreux exemples de systèmes de détection de défauts dans des systèmes industriels divers : les procédés, les transports, les télécommunications, l'agroalimentaire, la production d'énergie, l'aéronautique... Dans tous les cas, ces exemples soulignent la complexité de la fonction de supervision des systèmes industriels. Dans ce contexte, les systèmes de surveillance automatique sont des outils indispensables en production afin de soutenir et d'aider le conducteur d'atelier dans leur tâche [Dubuisson, 2001].

L'objectif de ce chapitre est dans un premier temps, d'établir la terminologie utilisée dans la suite de ce mémoire,. Dans un second temps, les principales techniques existantes pour la surveillance des systèmes dynamiques sont présentées, puis classées dans un tableau comparatif répertoriant leurs principes, leurs points forts et leurs faiblesses. Après ce bilan, les hypothèses de travail adoptées sont abordées.

3.1 LA TERMINOLOGIE

Dans la littérature, de nombreuses définitions d'un même mot sont trouvées, parfois différentes selon le domaine d'application considéré. À titre d'exemple, le terme *diagnostic* est employé dans des contextes variés :

- Au sens médical, il représente la démarche qui identifie la maladie à l'origine des symptômes ou observations constatés sur le patient ;
- En finance, le diagnostic désigne un bilan.
- En automatique, le diagnostic désigne toute méthode amenant à déterminer si un système est en fonctionnement normal ou non, et en cas de détection, à discriminer l'origine de la défaillance.

Afin de lever toute ambiguïté, il nous a semblé primordial de rappeler la terminologie utilisée dans ce rapport. Ces définitions reposent sur les travaux de : [Dubuisson, 1990 ; Zwingelstein, 1995 ;

3.1 la Terminologie

Isermann, 1997 ; Ploix, 1998 ; Isermann et Phalle, 2000 ; Lefebvre, 2000 ; AFNOR ; IEE]. Ces définitions sont illustrées à travers l'exemple de la figure 3.1 (exemple présenté au point 2.6.1).

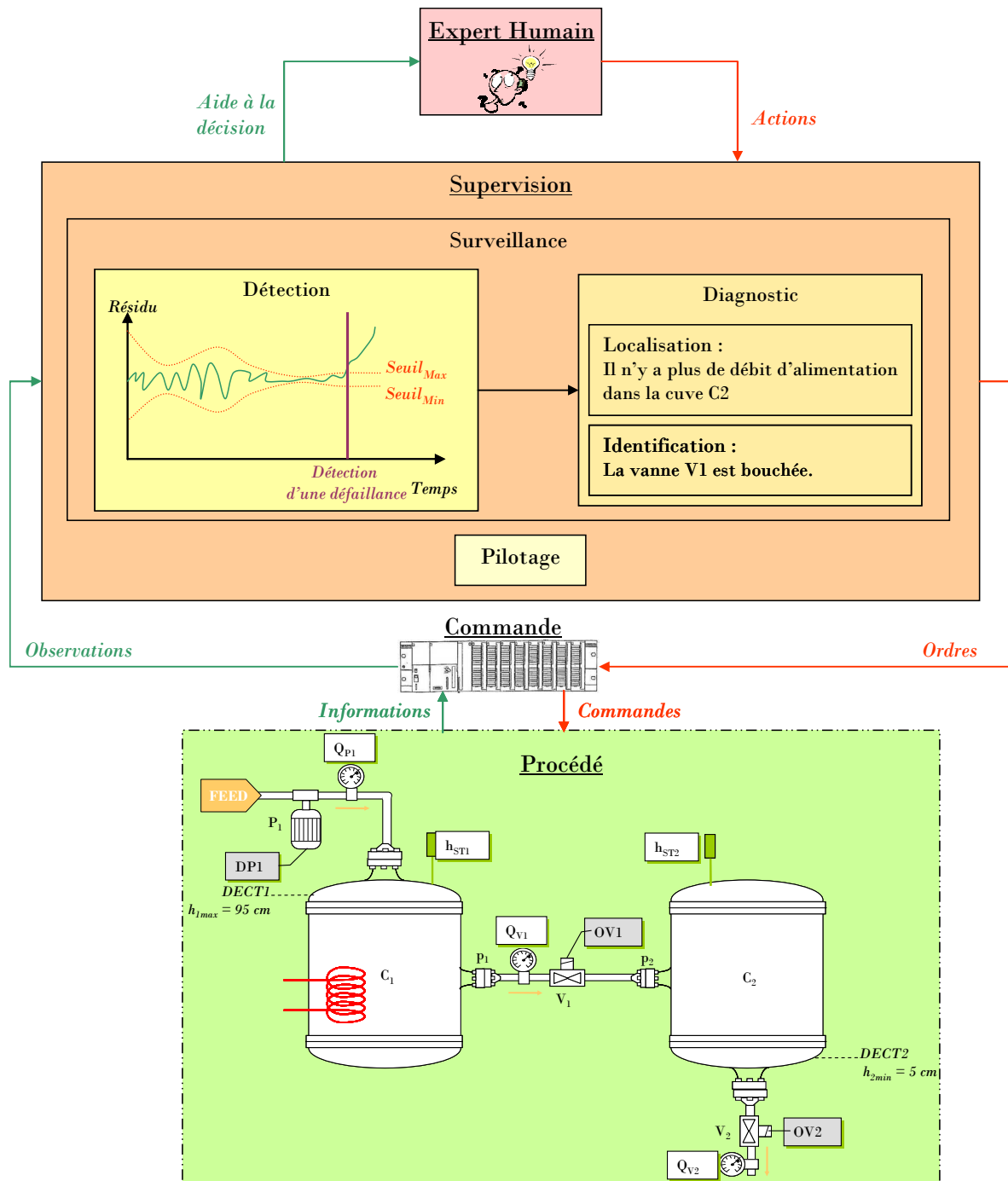


Figure 3.1 La boucle de supervision

3.1.1 Le système physique

Un **système physique** est un ensemble structuré d'éléments (composants, constituants) qui accomplit une fonction.

Exemple : Le système physique est ici un procédé, composé de deux cuves, deux électrovannes, une pompe et cinq capteurs (basé sur le *principe des « vases communicants »*).

L'**environnement** est le milieu dans lequel un système physique fonctionne.

Exemple : L'air ambiant, l'opérateur font partie de l'environnement.

Un **composant** est un élément discret du système étudié. Il peut être connecté ou en interaction avec d'autres composants. Il correspond au niveau minimal de granularité dans la structure composants / composés. Il s'agit donc de l'entité élémentaire de la description systémique choisie et son comportement est un fondement du comportement du système.

Exemple : L'électrovanne V1 est considérée comme un composant dynamique, le détecteur DECT1 est un composant électronique analogique, la conduite est un composant mécanique statique (cf. figure 3.1). Avec un autre niveau de granularité, l'électroaimant de V1 pourrait être un composant et l'électrovanne V1 serait un dispositif (cf. définitions suivantes).

Un **dispositif** est une entité composée d'éléments (composants) qui évoluent dans le temps dans le but de remplir une fonction définie.

Exemple : Le réacteur C1 est un dispositif dans lequel s'effectue les réactions chimiques ; celui-ci est composé de deux composants : une cuve de stockage et une alimentation d'énergie (cf. figure 3.1).

Chaque dispositif évolue en interaction avec son environnement. Les interactions entre le système et son environnement apparaissent de plusieurs manières possibles :

- L'environnement agit sur le système de façon explicite ou implicite.

Exemple :

- Explicite : un opérateur active la commande d'ouverture de la vanne V1 (cf. figure 3.1).

- Implicite : les réactions chimiques d'un procédé sont sensibles aux conditions opératoires (température, pression de l'environnement). Si celles-ci varient au cours du temps, cela aura un impact sur le système, plus précisément sur le rendement de la réaction chimique.

- Le système agit sur l'environnement de façon explicite et implicite.

Exemple :

- Explicite : quand le niveau de la cuve C1 atteint le niveau h_{1max} , alors le détecteur DECT1 déclenche une alarme (cf. figure 3.1). L'opérateur va alors fermer manuellement l'alimentation de la cuve C1. Ici le système « agit » sur l'opérateur.

- Implicite : une réaction exothermique rejette de la chaleur dans l'environnement et entraîne donc une augmentation de sa température.

Un **modèle** caractérise un système physique par un ensemble de propriétés qui en facilitent la description et la compréhension. Il peut correspondre à une représentation structurelle, comportementale et/ou fonctionnelle de chacun de ses composants et dispositifs. Le niveau structurel traduit les connections entre les différents composants et dispositifs du système physique. La représentation comportementale est constituée de relations dites de « causes à effets » (relations entre les variables du système), tandis que la représentation fonctionnelle est plus abstraite et correspond aux fonctions que le système doit accomplir.

On va alors distinguer deux types d'approches : les modèles *phénoménologiques* et les modèles *comportementaux* [Walter et Pronzalto, 1997].

On distingue deux types de modèle quantitatif :

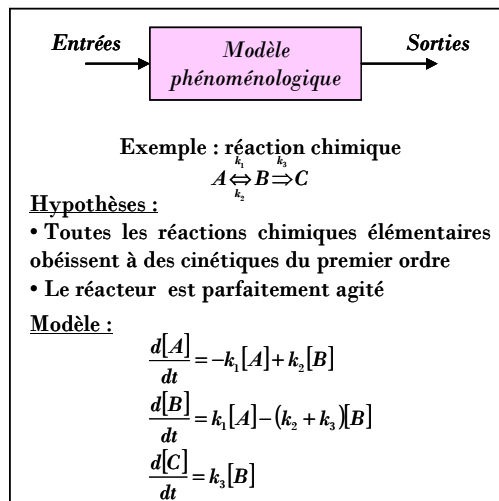


Figure 3.2 Le modèle phénoménologique

- Les modèles *phénoménologiques* (cf. figure 3.2), appelés aussi modèles de connaissance, reposent sur une compréhension plus approfondie des phénomènes régissant le système physique. Ils se basent sur les principes et les lois physico-chimiques (par exemple les équations bilan, réaction, thermodynamique...);

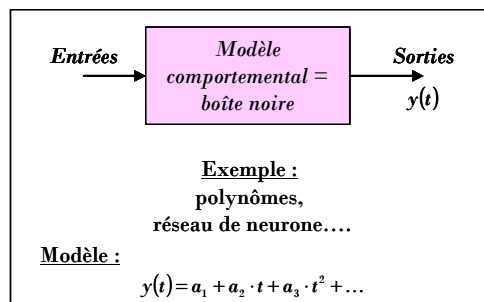


Figure 3.3 Le modèle comportemental

- À l'inverse, les modèles *comportementaux* (cf. figure 3.3) se limitent à reproduire le comportement observé sans connaissances particulières sur le processus. Ils sont fondés sur l'expérience, et

représentent la dynamique du système ou d'un sous système autour d'un point de fonctionnement par une fonction de transfert ou une représentation d'état.

3.1.2 Les observations

Une **observation** est une information obtenue sur le dispositif réel. Cette information peut être obtenue soit à une entrée, soit à une sortie du dispositif ou encore entre deux composants de celui-ci.

Exemple : Les mesures obtenues par le détecteur DECT1 constituent une observation du système (cf. figure 3.1).

Un **symptôme** est un phénomène qui survient sur un dispositif et qui révèle un dysfonctionnement.

Exemple : La diminution anormale du rendement de la réaction endothermique est un symptôme.

Le **syndrome** est un ensemble de symptômes caractéristiques d'un dysfonctionnement. Ainsi certains dysfonctionnements sont diagnostiqués plus rapidement s'ils sont associés à un syndrome caractéristique.

Exemple : Supposons que les deux symptômes ont été qu'on a constaté que :

- le rendement de la réaction endothermique est plus faible que le rendement nominal.
- la température en sortie de procédé a diminué.

La combinaison de ces deux symptômes est un syndrome. Elle permet de faire un diagnostic rapide : la température du réacteur a baissé.

3.1.3 Les anomalies

Une **perturbation** est un bruit, une entrée non contrôlée qui agit sur le système.

Exemple : Le bruit inhérent aux mesures est une perturbation du système physique.

Un **défaut** est un écart de comportement du système, une dérive par rapport au comportement normal. Il n'entraîne pas nécessairement le dysfonctionnement du système mais laisse envisager une probable défaillance.

Exemple : L'encrassement de la vanne V2 (cf. figure 3.1) est un défaut.

On distingue alors les défauts *additifs* des défauts *multiplicatifs* :

- Le défaut *additif* représente un signal parasite qui s'ajoute au signal principal (offset constant sur un capteur, débit de fuite sur un réservoir, ouverture incomplète d'une vanne).
- Le défaut *multiplicatif* représente une modification de paramètres (taux d'encrassement d'une vanne ou d'une pompe, etc.)

Une **panne** (fault) est la cause de l'apparition des symptômes. Elle peut être définie comme un état anormal (fonctionnement dégradé ou défaillant) d'une unité fonctionnelle (soit le système complet soit un de ses dispositifs logiciels ou matériels) la mettant dans l'impossibilité d'accomplir une de ces fonctions.

Exemple : Le blocage de la vanne V2 (cf. figure 3.1) en position ouverte est une panne possible du système physique.

Une **dégradation** est une baisse des performances d'une des fonctionnalités d'un appareil. À partir d'un certain seuil, il n'y a plus de dégradation mais défaillance.

Exemple : Le débit de la vanne V2 (cf. figure 3.1) est moins important à un epsilon près que celui donné par la consigne.

Une **défaillance** (failure) est une altération ou une interruption avérée des performances d'une des fonctionnalités d'un appareil. Il constitue le passage d'un état normal à un état de panne. Il est la conséquence d'un défaut ou dans le pire des cas d'une panne.

Exemple : Le débit de la vanne V2 (cf. figure 3.1) est beaucoup moins important que la consigne dans le cas d'un défaut, voire nul dans le cas d'une panne.

On peut classer les anomalies¹ selon un *degré de criticité* :

- Une anomalie est *assimilable*, lorsqu'elle n'a aucun impact sur les performances des fonctionnalités de l'appareil.

Exemple : Le débit de la vanne V2 (cf. figure 3.1) est plus faible à un epsilon près du débit nominal.

- Une anomalie est *caractéristique*, lorsqu'elle peut être contournée par une action corrective provenant de la fonction de reprise (défini dans le point 3.1.5).

Exemple : Considérons un défaut de la vanne V1 : son débit est trop faible pour permettre la vidange de la cuve C2 vers la cuve C1, dans un temps imparti (cf. figure 3.1). Dans ce cas là, la vanne V2 est ouverte pour corriger la déviation.

- Une anomalie est *critique*, lorsqu'elle nécessite une action d'urgence de la fonction de reprise.

Exemple : Considérons une panne de la vanne V2 (cf. figure 3.1) : elle reste bloquée dans l'état fermé. La vanne V1 ne pouvant, à elle toute seule, faire diminuer le niveau de liquide de la cuve C1, l'alarme de niveau maximum atteint est lancée par le détecteur DECT1. Une action d'urgence est mise en place : arrêt de la pompe P1 qui alimente la cuve C1.

Le terme **mode de fonctionnement** est utilisé pour décrire les différentes situations de fonctionnement d'un dispositif. On distingue les modes de fonctionnement normal, dégradé et défaillant.

- Le mode de fonctionnement est *normal*, lorsque les variables du système restent dans le domaine des valeurs nominales. Le système physique accomplit ses fonctions sans diminution de ses performances.
- Le mode de fonctionnement est *dégradé*, lorsque les variables du système ne sont plus dans le domaine des valeurs nominales et que le système accomplit partiellement sa fonction ou que ses performances sont amoindries.

Exemple : Ainsi, le système dont la vanne V2 (cf. figure 3.1) est encrassée (débit plus faible) est en mode dégradé : le débit est plus faible mais le système ne nécessite pas d'arrêt.

¹ La notion d'anomalie couvre indifféremment la notion de défaut, de panne, de dégradation et de défaillance.

- Le mode de fonctionnement est *défaillant*, lorsque les variables du système ne sont plus dans le domaine des valeurs nominales et que le système n'est plus apte à accomplir sa fonction ou que ses performances sont fortement amoindries.

Exemple : C'est le cas du système dont la vanne V2 (cf. figure 3.1) est bloquée fermée : cette défaillance a un impact sur le fonctionnement du système.

La distinction entre ces définitions est établie au travers du tableau 3.1. Il représente les anomalies selon leur degré de criticité. Par ailleurs, selon [Ploix, 1998], on distingue les anomalies comportementales et les anomalies fonctionnelles (altération des performances des fonctionnalités d'un dispositif).

| Notions | Type | Degré de criticité | Mode de fonctionnement | Exemple |
|--------------|--------------|-------------------------------|------------------------|--|
| Perturbation | Comportement | Assimilable | Normal | Bruit des mesures |
| Défaut | | Caractéristique | Dégradé | Encrassement de la vanne |
| Panne | | Critique | Défaillant | Blocage de la vanne en position fermée |
| Dégradation | Fonction | Assimilable à caractéristique | Dégradé | Débit de la vanne plus faible → utilisation d'une vanne de secours |
| Défaillance | | Caractéristique à critique | Défaillant | Débit de la vanne nulle → arrêt d'urgence |

Tableau 3.1 Les anomalies

3.1.4 La surveillance

La **surveillance** consiste en l'analyse de l'état du système et en l'extraction d'informations permettant de détecter les défaillances des systèmes et de les diagnostiquer.

3.1.4.1 La détection

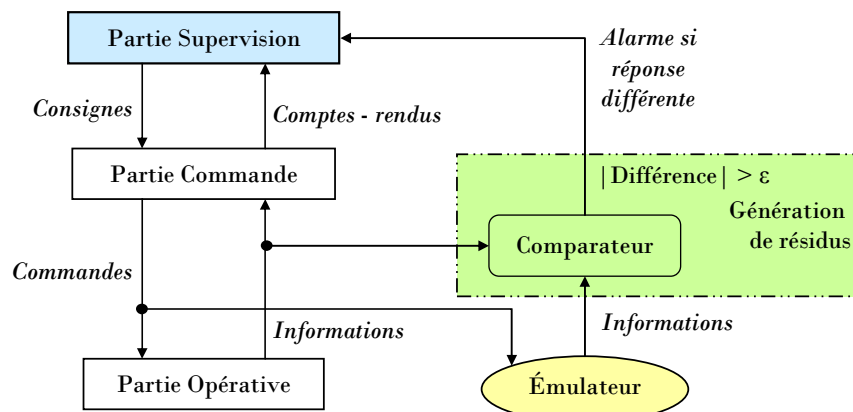


Figure 3.4 Un exemple de détection grâce à un émulateur

La **détection** (cf. figure 3.4) consiste à reconnaître qu'un dispositif est dans un mode de dysfonctionnement (mode de fonctionnement dégradé ou défaillant) à partir de la connaissance de certaines de ses caractéristiques. Elle génère alors des symptômes grâce aux observations et aux modèles de référence.

Le **résidu** est établi à partir de la comparaison entre le comportement du système observé et le comportement attendu obtenu à l'aide du modèle de référence. Il traduit la présence ou non du défaut.

3.1.4.2 Le diagnostic

Le **diagnostic** consiste à déterminer les composants ou organes défaillants du système physique à l'aide d'un raisonnement logique basé sur les symptômes et les observations du système. Dans la littérature scientifique, on rencontre aussi bien les termes « diagnostic de défaillances » (failure diagnosis) que « diagnostic de pannes » (fault diagnostic). Même s'il est logique de s'intéresser aux défaillances, la recherche de défauts apporte plus d'informations pour trouver les éléments défaillants et peut amener au diagnostic prédictif de défaillances.

La **localisation de défaut** consiste à déterminer les causes physiques ou fonctionnelles d'une défaillance. Ce raisonnement s'appuie sur les symptômes qui se sont révélés lors de la phase de détection.

L'**identification d'un défaut** permet de déterminer l'amplitude et l'évolution probable au cours du temps du défaut.

Ces notions sont représentées sur la figure 3.1.

3.1.4.3 La surveillance prédictive

La **surveillance prédictive** consiste en l'analyse de l'état présent et passé du système physique afin d'établir les dégradations et futures défaillances du système. On retrouve dans cette fonctionnalité les sous parties détection et diagnostic.

La **détection prédictive** détecte une dégradation du système et ainsi prédit la possibilité d'une future défaillance.

Le **diagnostic prédictif**, appelé aussi pronostic, consiste à déterminer les composants ou organes qui ont entraîné une dégradation du système mais pas encore une défaillance.

3.1.5 Les corrections

La **reprise** consiste à trouver le remède de la panne. Dans le cas le plus simple, c'est le remplacement de l'élément défaillant.

La **maintenance** est la fonction qui permet le remplacement ou la réparation des équipements usagés ou défaillants. Nous distinguons deux types de maintenance :

- La maintenance préventive imposée par la sûreté de fonctionnement. Elle peut être :
 - soit *systématique*, c'est-à-dire effectuée selon un échéancier établi suivant le temps d'usage ou à partir du nombre d'unités d'usage ;

- soit *conditionnelle* : elle intervient, lors de la prédiction d'une future défaillance du système physique, c'est-à-dire lors d'une dégradation des performances du système.
- La maintenance corrective, effectuée après défaillance. Elle intervient au cours de la fonction reprise. Elle est :
 - soit *curative* : elle consiste en la remise en l'état initial, ce qui peut correspondre au remplacement du composant défectueux.
 - soit *palliative* : dans ce cas une solution de secours provisoire, permettant au composant défaillant d'assurer au moins une partie de ses fonctionnalités ; il doit toutefois être suivi d'une action curative dans les plus brefs délais.

Exemple : Reprenons l'exemple suivant : la vanne V1 est encrassée et son ouverture ne permet plus la vidange de la cuve C1 vers la cuve C2 (cf. figure 3.1). Dans ce cas là, la maintenance curative a été l'ouverture de la vanne V2.

3.2 LA CLASSIFICATION DES MÉTHODES DE DIAGNOSTIC

Le développement de nouvelles technologies a introduit une complexité supplémentaire au niveau des systèmes temps réel par le fait qu'ils combinent à la fois des éléments matériels et des éléments logiciels. De nos jours, le maintien des performances et de la productivité des systèmes industriels est un défi important. Dans ce contexte, le rôle de la fonction de surveillance est de détecter et d'identifier, à partir de symptômes observés, les causes d'une dégradation ou d'une défaillance d'un composant ou plusieurs composants du système physique. Le panel de méthodes de diagnostic est considérable, et de nombreuses classifications sont proposées dans la littérature : [Franck et Köppen-Seliger, 1997], [Isermann, 1997], [Travé-Massuyès *et al.*, 1997], [Chantler *et al.*, 1998], [Dash et Venkatasubramanian, 2000] [Gentil, 2003], [Venkatasubramanian *et al.*, 2003]. Ces classifications sont influencées par les contextes et les domaines d'application particuliers à chaque communauté et ne sont donc pas homogènes.

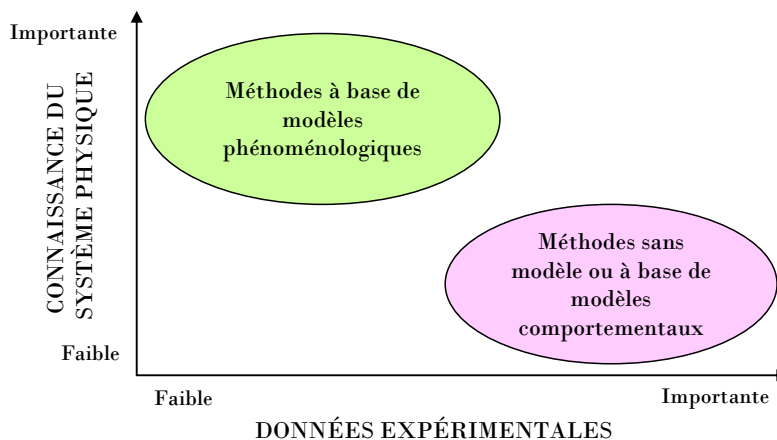


Figure 3.5 Les méthodes selon leur degré de connaissance et de nécessité de données expérimentales

De ce fait, il est souvent délicat de déterminer la méthode la plus appropriée pour la résolution du problème de détection et diagnostic [Biswas *et al.*, 2004]. La section 1.4 a montré que les méthodes de diagnostic fondées sur l'utilisation de modèle sont souvent préférables, notamment dans un contexte dynamique. Ces méthodes et le type de modèle associé se différencient non seulement par la nature de la connaissance dont on dispose sur le système physique et ses défaillances, mais également par la manière dont cette connaissance est exploitée. Néanmoins, à un premier niveau nous pouvons les classer en deux grandes familles (cf. figure 3.5).

- les méthodes qui nécessitent une connaissance approfondie du système physique ; ce sont des méthodes à base de modèles phénoménologiques.
- les méthodes ne nécessitant pas de connaissances accrues du système physique, mais qui utilisent une connaissance puisée dans les expériences passées, dite aussi superficielle ; on distingue alors les méthodes sans modèle ou à base de modèles comportementaux.

La figure 3.6 représente la typologie des méthodes de diagnostic abordées dans ce manuscrit :

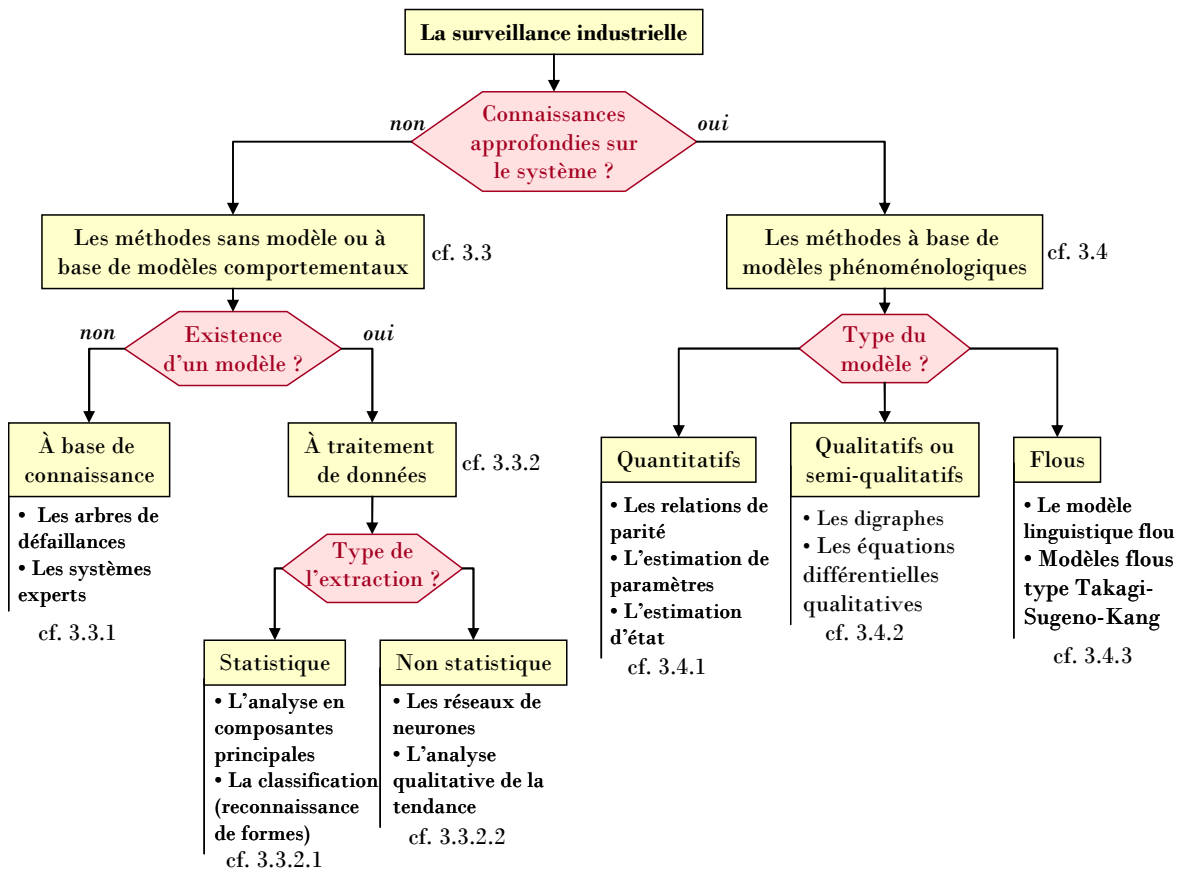


Figure 3.6 La typologie des méthodes de diagnostic

Les sections suivantes ont pour objectif de décrire plus en détail ces différentes méthodes.

3.3 LES MÉTHODES SANS MODÈLE OU À BASE DE MODÈLES COMPORTEMENTAUX

Nombreuses sont les applications industrielles dont la représentation (modèle phénoménologique) est difficile, voire impossible, à obtenir, du fait de la complexité des phénomènes mis en jeu ou des nombreuses reconfigurations intervenants dans le processus de production. L'information fournie par l'instrumentation du procédé reste néanmoins disponible. Il existe alors des méthodes n'utilisant aucune connaissance approfondie du procédé. On distingue dans ce type d'approche deux types de méthodes :

- des méthodes sans modèle, des méthodes dites qualitatives, appelées méthodes à base de connaissances,
- des méthodes à base de modèles comportementaux, des méthodes dites quantitatives, appelées méthodes basées sur le traitement de données.

3.3.1 Les méthodes à base de connaissance

Ces méthodes sont mises en œuvre, lorsque la construction d'un modèle est difficile et lorsque peu de mesures du système sont disponibles. L'expérience et la connaissance de l'opérateur sont alors exploitées pour assurer le bon fonctionnement du processus.

3.3.1.1 Les méthodes d'analyse des défaillances

Généralement, ces méthodes sont issues du contexte de la Sécurité de Fonctionnement. Elles se basent sur une analyse fonctionnelle et matérielle du système afin d'associer des symptômes à des causes. Ce sont donc des approches relationnelles ne nécessitant aucun traitement de données.

3.3.1.1.1 L'AMDE et ses dérivées

L'AMDE (Analyse des Modes de Défaillance et de leurs Effets) a été introduite au cours des années soixante, dans le domaine de l'aéronautique pour la sûreté des avions [Recht, 1966]. Elle consiste en l'analyse systématique et complète, composant par composant, de tous les modes de défaillance possibles en précisant leurs effets sur le système global [Kempowski, 2004]. Les étapes sont les suivantes :

- définition du système, de ses fonctions et de ses composants,
- établissement de l'ensemble des modes de défaillance des composants,
- pour chaque mode de défaillance, recherche des causes possibles de son apparition,
- étude des effets sur le système et sur l'opérateur pour chaque combinaison (cause, mode de défaillance).

L'AMDE est une procédure déductive qui utilise des tableaux comme outil d'identification des causes de défaillances à partir des effets observés. Cette démarche est aujourd'hui très répandue

[Villemeur, 1988] et fait aujourd'hui partie intégrante des méthodes de conception dans de nombreux secteurs industriels tels que l'automobile, la chimie...

| Fonction | Élément | Modes de défaillances | Effets | Causes | Citation | | | |
|---|----------|-------------------------------|--------------------------|--|----------|---|----|-----|
| | | | | | D | F | G | C |
| 100 Système de remplissage | Pompe P1 | Moteur cassé | Débit d'alimentation nul | Age de la pompe | 5 | 2 | 8 | 80 |
| | | Moteur essoufflé | Débit trop faible | Pompe non révisée | 1 | 3 | 5 | 15 |
| | Commande | Pas d'alimentation électrique | Débit nul | Panne réseau électrique | 10 | 2 | 3 | 60 |
| | | Opérateur absent | Pompe sur off | Mauvaise organisation | 6 | 3 | 3 | 54 |
| | | Automatisme défectueux | Pompe sur off | Mauvaise programmation ou rupture de ligne | 2 | 1 | 3 | 6 |
| 200 Système de vidange | Vanne V2 | Fuite vanne | Défaut d'étanchéité | Vanne sous dimensionnée | 2 | 4 | 5 | 40 |
| | | Vanne grippée | Échauffement | Manque de lubrifiant | 1 | 4 | 2 | 8 |
| | | Vanne bloquée | Position Ouverte | Vanne forcée | 5 | 2 | 9 | 90 |
| | | Vanne bloquée | Position Fermée | Vanne forcée | 5 | 2 | 10 | 100 |
| 300 Système de recirculation des fluides | Conduite | Partiellement obstruée | Débit affaibli | Nettoyage peu fréquent | 5 | 3 | 6 | 90 |
| | | Totalement bouchée | Débit nul | Nettoyage peu fréquent | 1 | 2 | 10 | 20 |
| | Vanne V1 | Fuite vanne | Défaut d'étanchéité | Vanne sous dimensionnée | 2 | 4 | 5 | 40 |
| | | Vanne grippée | Échauffement | Manque de lubrifiant | 1 | 4 | 2 | 8 |
| | | Vanne bloquée | Position Ouverte | Vanne forcée | 5 | 2 | 4 | 40 |
| | | Vanne bloquée | Position Fermée | Vanne forcée | 5 | 2 | 10 | 100 |
| 400 Système de stockage | Cuve C1 | Réservoir troué | Épanchement de fluide | Matière cuve ou épaisseur cuve | 2 | 1 | 9 | 18 |
| | | Réservoir encrassé | Contenance diminuée | Nettoyage peu fréquent | 8 | 6 | 1 | 48 |
| | Cuve C2 | Réservoir troué | Épanchement de fluide | Matière cuve ou épaisseur cuve | 2 | 1 | 9 | 18 |
| | | Réservoir encrassé | Contenance diminuée | Nettoyage peu fréquent | 8 | 6 | 1 | 48 |

Légende :

C : La criticité avec $C = G * F * D$

F : L'indice de fréquence (probabilité d'apparition)

G : L'indice de gravité (conséquences de la défaillance)

D : L'indice de détection (traduit la facilité ou non à détecter)

} Note de 1 à 10 où 1 indique un indice faible et 10 un indice élevé

Tableau 3.2 L'AMDEC

Cette approche est complétée parfois par une analyse de criticité. L'AMDEC (Analyse des Modes de Défaillance et de leurs Effets et de la criticité) permet alors d'extraire les modes de défaillance les plus critiques. La criticité est évaluée à partir du couple probabilité-gravité.

Le tableau 3.2 illustre cette approche. Il est issu de l'Analyse des Modes de Défaillance de leurs Effets et de leurs Criticité du procédé illustré sur la figure 3.1. L'utilisation de ce tableau à des fins de diagnostic industriel, conduit à utiliser ce tableau comme un outil d'identification de la cause de la défaillance à partir des observations sur le procédé [Zwingelstein, 1995].

Notons que l'AMDEC est la traduction de l'anglais FMECA, *Failure Modes, Effects and Critically Analysis*.

Une autre extension de l'AMDE est l'HAZOP (Hazard and operability study). C'est une analyse de fonctionnement. Elle consiste en l'exploration systématique de 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.

3.3.1.1.2 Les arbres de défaillances ou arbres de causes

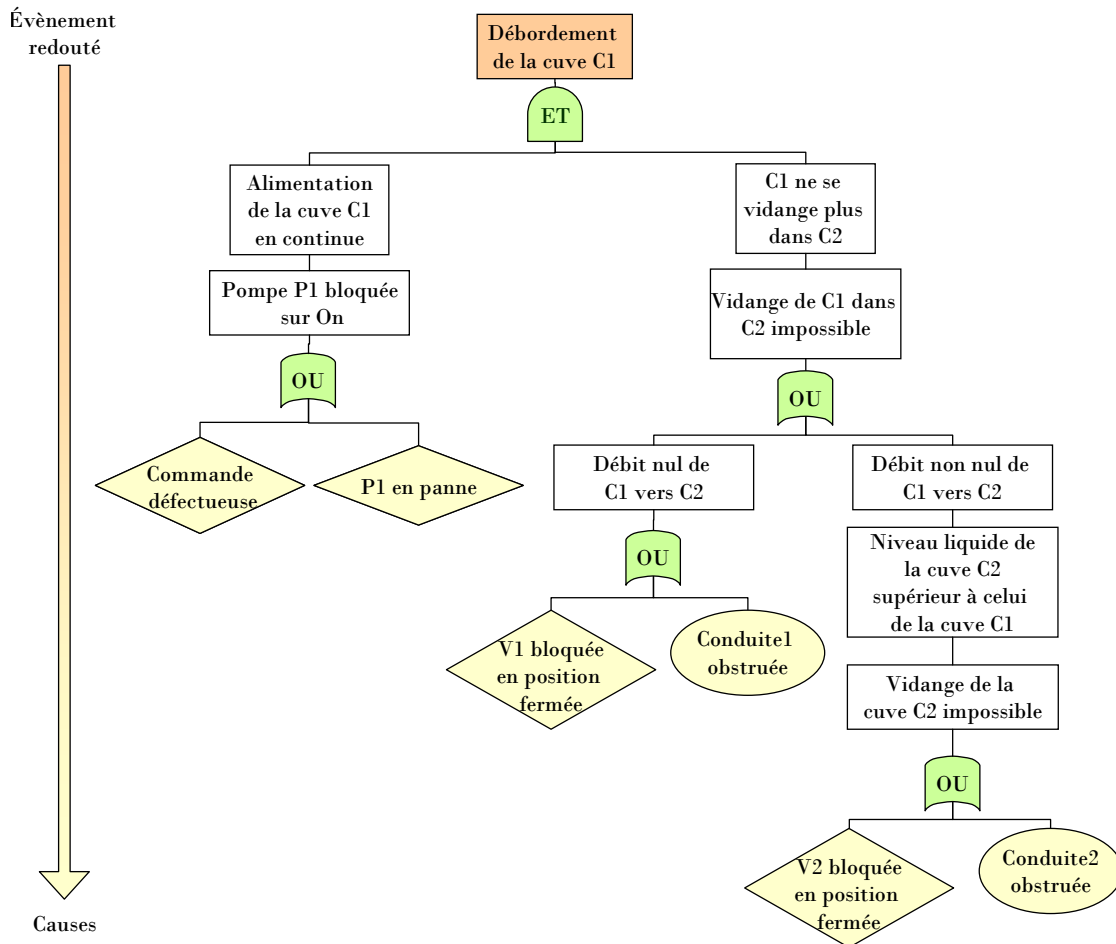


Figure 3.7 L'arbre de défaillances

Conçue au début des années soixante aux États-Unis, cette approche est un outil majeur de l'analyse de risque technologique [Villemeur, 1988]. Elle est employée communément dans les analyses de fiabilité, de disponibilité ou de sécurité des systèmes. [Kempowski, 2004]. Cette méthode, optimisée, détermine les chemins critiques dans un système, c'est-à-dire les diverses combinaisons possibles d'évènements qui entraînent la réalisation d'un évènement indésirable unique. Cette méthode déductive est basée sur l'identification des points faibles de la conception. Sa représentation graphique est composée d'une structure arborescente (arbre) permettant un traitement à la fois qualitatif et quantitatif.

L'arbre est construit sous forme de diagramme logique où un sommet correspond à un évènement indésirable et où à la base se trouve l'ensemble des évènements considérés comme élémentaires. Les causes immédiates produisant l'évènement redouté sont hiérarchisées à l'aide de symboles logiques « ET » et « OU ». À partir d'un ensemble de symptômes, le principe du diagnostic est alors de construire une chaîne d'inférence pour obtenir un ensemble de défaillances à l'origine de ces symptômes.

Une méthode dérivée des arbres de défaillances est le diagramme de causes-conséquences. Cette méthode allie les principes utilisés par l'analyse déductive de l'arbre de défaillances et l'analyse inductive de l'arbre des conséquences.

La figure 3.7 représente un arbre de défaillances, dont l'élément redouté est le débordement de la cuve C1 du procédé illustré sur la figure 3.1.

3.3.1.2 Les systèmes experts

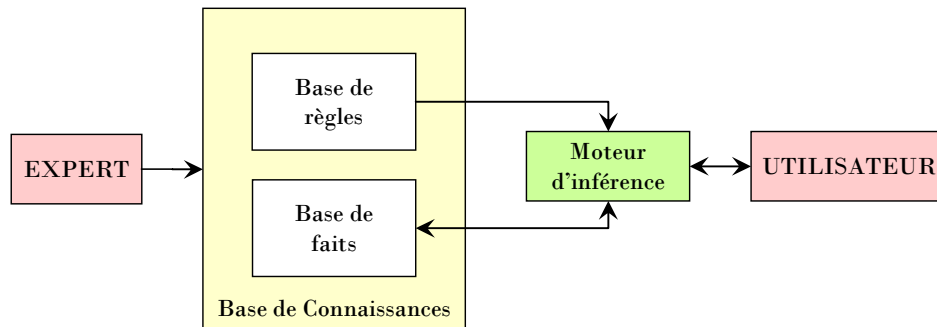


Figure 3.8 La structure d'un système expert [Kempowski, 2004]

Les systèmes experts sont des outils issus de l'intelligence artificielle et constituent un autre moyen de traiter automatiquement les connaissances établies dans la section 3.3.1.1. Ils codifient les mécanismes cognitifs d'un expert humain. Ils ont la capacité de représenter et de restituer les connaissances acquises par les spécialistes d'un domaine particulier. Les connaissances utilisées reposent sur l'apprentissage des relations entre les causes et les effets observés pour chaque défaillance. Ils ne font pas seulement qu'emmagasiner des données : ils accumulent également des règles qui peuvent être utilisées pour manipuler ces données, ceci dans le but de répondre à des questions les concernant. Les systèmes experts sont composés de deux parties indépendantes (cf. figure 3.8) :

- une base de connaissances composée :

- d'une base de règles qui regroupe les règles de production modélisant la connaissance du domaine considéré,
- et d'une base de faits qui contient les symptômes ou les événements traduits sous forme de symboles,
- un moteur d'inférences capable de raisonner à partir des informations contenues dans la base de connaissances, de faire des déductions, etc. et de rajouter de règles dans la base de faits.

Le moteur d'inférence est capable d'utiliser faits et règles pour produire de nouveaux faits, jusqu'à parvenir à la réponse à la question experte posée. En général, les systèmes experts reposent sur des mécanismes de logique formelle et utilisent le raisonnement déductif. Ils exploitent la règle d'inférence suivante :

« Si P est vrai (fait ou prémisse) et si on sait que P implique Q (règle) alors, Q est vrai (nouveau fait ou conclusion). »

La figure suivante représente un exemple de base de connaissance d'un système expert :

Base de faits :

- F1 : Si le débit en entrée de la cuve C1 est supérieur à 0, alors envisager fortement une défaillance de la pompe (pompe reste activée).
- F2 : Si le débit entre C1 et C2 est non nul alors envisager fortement la défaillance de la vanne V2 : celle-ci est bloquée en position fermée.
- F3 : Si le débit entre C1 et C2 est nul alors envisager fortement la défaillance de la vanne V1 : celle-ci est bloquée en position fermée.
- F4 : Si le débit en entrée de la cuve C1 est supérieur à 0 et que la révision de la pompe a été faite récemment, envisager une défaillance de commande.
- F5 : Si le débit entre C1 et C2 est non nul et que la révision de la vanne V2 a été faite récemment, envisager une conduite défectueuse (bouchage).
- F6 : Si le débit entre C1 et C2 est nul et que la révision de la vanne V1 a été faite récemment, envisager une conduite défectueuse (bouchage).
- ...

Base de règles :

- R1 : Si le niveau d'alerte maximale est atteint dans la cuve C1 alors envisager F1, F2, F3, F4, F5 et F6.
- R2 : Si le produit utilisé est une suspension liquide solide et si F1, F2, F3, F4, F5 et F6 sont envisagées, alors envisager en priorité la défaillance d'une conduite (bouchage)
- ...

Figure 3.9 Un exemple de base de connaissance d'un système expert

Il existe de nombreux types de moteurs, capables de traiter différentes formes de règles logiques pour déduire de nouveaux faits à partir de la base de connaissance. Le parcours de règles se fait en général selon trois modes :

- le chaînage avant : on part des faits et règles de la base de connaissance, et on tente de s'approcher des faits recherchés par le problème,
- le chaînage arrière : on part des faits recherchés par le problème, et on tente par l'intermédiaire des règles, de « remonter » à des faits connus,

- le chaînage mixte : une combinaison des deux précédentes approches est ici utilisée.

L'intérêt de ce type de système est de pouvoir faire évoluer facilement la base de connaissance (acquisition de nouvelles connaissances) sans avoir à retoucher aux mécanismes d'inférence.

3.3.2 Les méthodes basées sur le traitement de données

Les méthodes basées sur le traitement de données s'appuient sur des modèles dits comportementaux, et nécessitent généralement un large éventail de données historiques. L'idée est alors d'analyser ces données par différentes techniques afin de proposer un modèle de comportement. Ces données sont donc transformées en une source de connaissance pour le système de diagnostic, lors d'une étape dite *d'extraction de caractéristiques*. À ce niveau, on distingue habituellement deux types de techniques d'extraction selon qu'elle soit statistique ou non.

3.3.2.1 Les techniques d'extraction statistique de données

3.3.2.1.1 L'analyse en composantes principales

L'Analyse en composantes principales est une technique statistique multi-variable utilisée avec succès dans les études de diagnostic. Cet outil permet de compresser des données et de réduire leur taille, afin de ne conserver que l'information utile et ainsi, rendre son analyse plus aisée que dans son ensemble. Le principe est donc de trouver un ensemble de facteurs (composantes) de dimension inférieure à celle de l'ensemble initial qui décrivent correctement les tendances principales et expliquent alors les liaisons statistiques entre les phénomènes.

L'information est représentée par la dispersion d'une population de n individus décrits par p variables. L'objectif est de produire un résumé de cette information. Pour cela, une technique mathématique d'orthogonalité linéaire est utilisée : les points sont projetés du sous-espace de dimension m (m est le nombre de variables observées) dans un espace de dimension $q < m$, en maximisant la variance des projections (ou en minimisant l'erreur quadratique d'estimation) [Harkat, 2003]. La solution de ce problème de maximisation décrit simultanément la projection du sous-espace de dimension q et la projection inverse (du sous-espace de dimension q vers le sous-espace de dimension m) afin d'estimer les variables initiales.

Les composantes principales sont déterminées grâce aux calculs des vecteurs propres de la matrice de covariance.

Considérons un groupe de données de dimension n : $X = [X_1, \dots, X_n]$ avec $X_i \in \mathfrak{R}^m$.

On calcule alors la matrice de covariance Cov associée à cet ensemble de points :

$$Cov = \frac{1}{n} \sum_{i=1}^n X_i \cdot X_i^T \quad (3 - 1)$$

Les valeurs propres de la matrice de covariance de données sont déterminées par un algorithme de détermination des vecteurs propres :

$$Cov \cdot V = \lambda \cdot V \quad (3 - 2)$$

Avec λ les valeurs propres et V est le vecteur propre associé.

Les valeurs propres de la matrice de covariance représentent les variances des projections des données sur les directions représentées par les vecteurs propres [Harkat, 2003]. Les vecteurs propres avec les plus grandes valeurs propres sont utilisés comme vecteurs de base sur lesquels les données sont projetées [Jolliffe, 1986 ; Oja *et al.*, 1992]. Les q composantes principales des données sont les directions orthogonales qui capturent la plus grande variation de données (cf. figure 3.10).

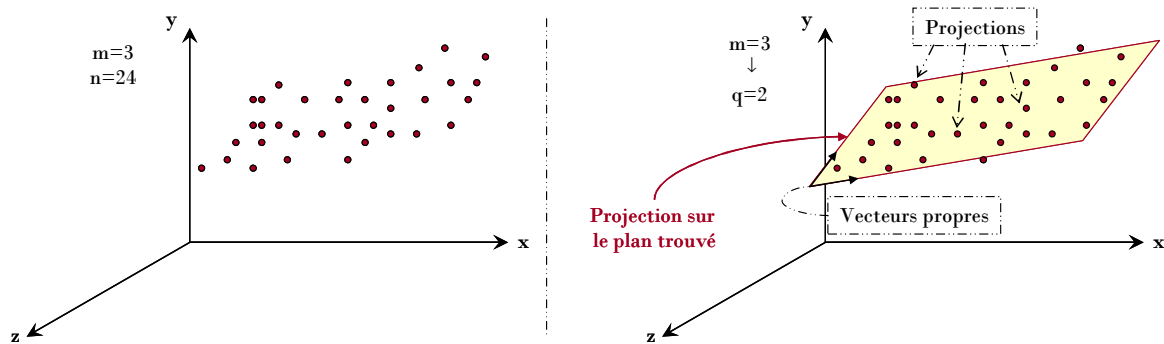


Figure 3.10 Une illustration de l'analyse en composantes principales

L'analyse en composantes principales ne prend en compte que les variables du processus. Il peut être toutefois intéressant d'ajouter à ses données des données liées à la qualité du produit. Dans ce cas, la méthode des moindres carrés partiels est un outil envisageable [Venkatasubramanian *et al.*, 2003]. Cette méthode modélise le rapport entre deux blocs de données tout en les comprimant simultanément [Kempowski, 2004]. Les variables expliquant la variation de données du processus sont alors extraites.

3.3.2.1.2 La classification de données et reconnaissance de formes

Cette approche consiste en la reconnaissance d'une forme parmi différentes possibilités à partir d'observations bruitées. Le diagnostic de défaillances devient alors un problème de classification. En effet, il consiste à construire un bloc de correspondance : le regroupement d'observations forme des classes qui caractérisent une situation ou un mode de fonctionnement du processus (mode normal, dégradé ou défaillant). Une nouvelle observation sera alors comparée à ces classes pour être identifiée. Ainsi, à partir d'un ensemble d'informations décrivant la situation courante du processus, il est possible d'obtenir les causes probables des situations anormales.

Principe de base

Les observations (ou objets ou individus) qui ont des caractéristiques similaires sont regroupées. Une observation Z_i est définie par un vecteur noté :

$$Z_i = [z_1, z_2, \dots, z_D]^T$$

dans un espace, appelé *espace de représentation* ou de *description*, à D dimensions, où D correspond au nombre d'attributs ou caractéristiques.

Le regroupement des observations selon leurs caractéristiques permet la définition des classes décrivant des régions particulières de l'espace de représentation : $C = \{C_k, k = 1..K\}$. Ce regroupement constitue l'espace de décision. Il est fait à partir d'une phase d'apprentissage en utilisant une mesure de similitude par le biais des méthodes de classification.

Les paramètres caractérisant le vecteur d'observation, et par conséquent l'espace de représentation, sont les mesures disponibles issues des capteurs et des actionneurs du processus ou bien des informations extraites de ceux-ci. Seule la connaissance a priori sur le système permet de choisir les caractéristiques. Les classes correspondent aux différents modes de fonctionnement du processus

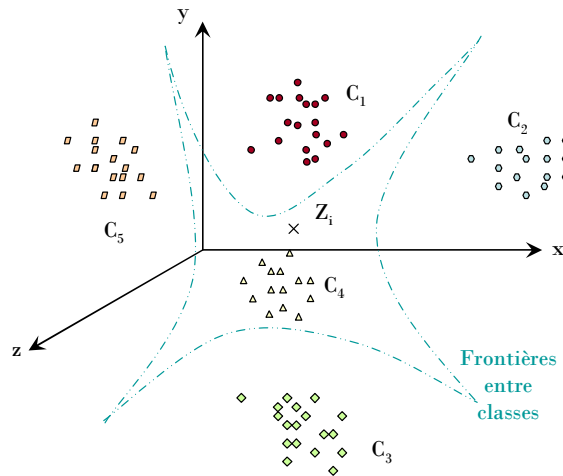


Figure 3.11 La reconnaissance de formes

La figure 3.11 illustre la notion de reconnaissance de forme : l'espace de description à trois dimensions et l'espace de représentation est composé de cinq classes. La connaissance des frontières entre classes permet l'affectation d'une nouvelle observation à l'une de ces cinq classes ; ici l'opération de classification affecte l'observation Z_i à la classe C_4 .

L'objectif est d'associer toute nouvelle observation Z_i à une classe de l'espace de décision. L'affectation d'une observation Z_i à l'une des K classes notées C_k , avec $k = 1..K$ indique une opération de classement ou de discrimination en se basant sur des connaissances a priori ou sur de l'information statistique puisée dans les formes. Une forme est une représentation simplifiée du monde extérieur. Elle correspond généralement à des ensembles d'observations, qui définissent des points dans un espace multidimensionnel approprié, et peuvent former des figures géométriques, descriptibles par une formule mathématique. Par ailleurs, une classe est caractérisée par une ou plusieurs formes. Le système en évaluant un critère de décision (exemples : calcul de distance, de barycentre...) peut alors assigner une forme à une classe mais il peut aussi déterminer avec quelle confiance il effectue cette décision. Si le critère de décision prend des valeurs très proches pour plusieurs classes, la confiance dans la décision est faible.

Déroulement

La classification se déroule en trois étapes [Dubuisson, 1990] :

- *une phase d'analyse* : il s'agit d'étudier les informations fournies par les différents capteurs, les historiques ainsi que l'expérience des opérateurs, afin de définir l'espace de représentation du système ainsi que les classes qui représenteront les différents états de fonctionnement. Notons ici l'intérêt d'utiliser la méthode d'analyse en composantes principales afin de ne conserver que l'information utile. Il s'agit donc de la phase de classification ;
- *une phase de choix du système de détection* : elle consiste à construire une règle de décision qui établira des frontières entre les différentes classes. La règle de décision (travail sur les barycentres, les distances, les frontières...) permettra d'affecter ou non une nouvelle observation à l'une des classes connues ;
- *une phase d'exploitation* : elle consiste à implémenter le système afin de proposer une décision pour toute observation effectuée.

De nombreux travaux soulignent l'intérêt de ces méthodes dans les systèmes complexes [Ribes *et al.*, 2002 ; Kempowski, 2004]. Il existe de nombreuses méthodes se basant sur cette approche. Parmi les plus classiques, citons les classificateurs statistiques tels que le classificateur bayésien, les k plus proches voisins (clustering, K-means) [Marie-Joseph, 2003 ; Casimir, 2003].

Soulignons l'existence d'une méthode dérivée de la reconnaissance de formes appliquée au domaine des télécommunications : la reconnaissance de scénarios (ou chroniques) [Boufaied, 2003 ; Guerraz, 2005]. Cette méthode considère les observations comme un ensemble d'évènements avec relations temporelles (observations reçues dans un ordre ou un autre avec des délais différents ou non). Une chronique est alors représentée par un graphe d'instant (ensemble d'observations et ensemble de contraintes temporelles entre les instants d'occurrences de celles-ci). À chaque observation, un nombre de chroniques candidates est proposé : c'est un ensemble d'instances de modèle de chronique pour lesquelles l'ensemble des observations reçues à l'instant t vérifie le graphe des instants de ces chroniques. À chaque nouvelle observation, on élimine de l'ensemble des chroniques compatibles celles qui ne le sont plus et on ajoute les chroniques qui peuvent débiter avec cette nouvelle observation.

3.3.2.2 Les méthodes d'extraction non statistique

3.3.2.2.1 L'approche fréquentielle

L'analyse fréquentielle est très utilisée pour la détection de phénomènes périodiques comme en analyse vibratoire. Elle analyse le spectre des signaux issus des capteurs et détermine très efficacement l'état de l'installation sous surveillance. Dans un premier temps, les signaux sont analysés en état normal de fonctionnement. Ensuite, toute déviation des caractéristiques fréquentielles d'un signal est reliée à une situation de panne : l'analyse de ces signaux par une procédure de décision (tests d'hypothèses) permet de détecter et d'interpréter tout comportement anormal du système surveillé et de localiser son origine. Nous pouvons citer comme test : le test empirique de franchissement de seuil, le test de moyenne ou encore le

test de variance. Pour plus d'informations sur les tests applicables sur un échantillon de mesures, le lecteur peut se reporter aux travaux de Basseville [Basseville, 1988, 1997].

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 [Orantes Molina, 2005]. Une étude détaillée de ces méthodes se trouve dans [Flandrin, 1998].

3.3.2.2 Les réseaux de neurones

L'utilisation de cette approche est envisagée quand la connaissance sur le procédé n'est pas suffisante et que le développement d'un modèle de connaissance du procédé est impossible. C'est une méthode dite « boîte noire » : elle peut être assimilée à un système informatique constitué d'un nombre de processeurs élémentaires interconnectés entre eux qui traite l'information qui lui arrive à partir des signaux extérieurs. Les premiers travaux sur ce sujet ont été développés par McCulloch et Pitts en 1943 [McCulloch et Pitts, 1943]. Un réseau de neurones définit implicitement une fonction non-linéaire, c'est-à-dire approche une fonction non linéaire, dont on ne connaît que quelques points. Dans le cadre du diagnostic, cela consiste à approcher la relation inconnue reliant les symptômes aux défaillances.

Un neurone formel possède plusieurs entrées ; un poids et une sortie sont affectés à chacune d'entre elles (cf. figure 3.12). De ce fait, un neurone peut avoir plusieurs neurones prédécesseurs. La sortie est calculée à partir des entrées et des poids synaptiques : une fonction d'entrée calcule le potentiel du neurone, celle-ci est très souvent la somme pondérée des entrées augmentée d'un seuil. Une fonction de transfert génère alors la sortie grâce à ce potentiel. Cette fonction de transfert est très importante, et détermine le fonctionnement du neurone et du réseau. Elle peut prendre de nombreuses formes, peut être linéaire, binaire, saturée ou sigmoïdale.

On définit donc un neurone formel comme suit (cf. figure 3.12) :

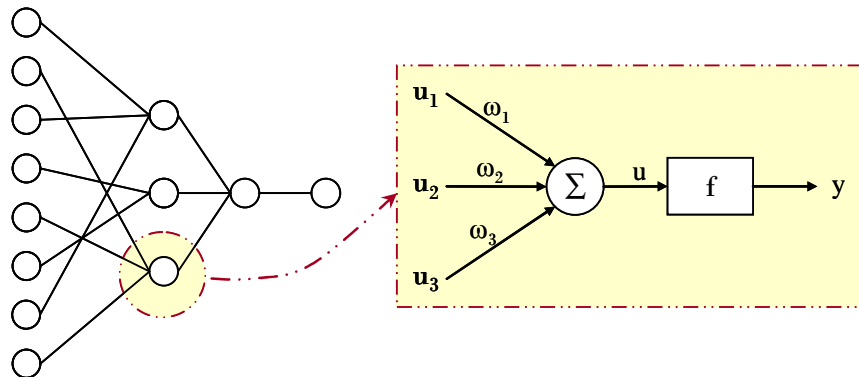


Figure 3.12 Un neurone formel

Chaque nœud calcule la somme de ses entrées (u_1, \dots, u_n) pondérées par les poids synaptiques correspondants ($\omega_1, \dots, \omega_n$).

Cette valeur représente l'état interne u du neurone :

$$u = \sum_{i=1}^n \varpi_i \cdot u_i \quad (3 - 3)$$

Ce résultat est alors transmis à une fonction d'activation f (fonction de transfert). La sortie y constitue l'activation du neurone :

$$y = f(u) \quad (3 - 4)$$

Un neurone formel réalise donc une fonction f à partir de l'état interne. L'interconnexion de plusieurs neurones forme alors un réseau de neurones. La réalisation de la synthèse du réseau s'effectue en différentes étapes :

- choix du type de réseau,
- type de neurones,
- nombre de couches,
- méthodes d'apprentissage : le paramétrage du réseau est réalisé à partir de la base d'exemples au moyen d'un algorithme d'apprentissage. Il est conçu pour minimiser un critère quadratique sur l'erreur d'approximation réalisée sur le modèle non-linéaire.

Ainsi, sur la base de l'optimisation d'un critère, l'apprentissage permet de reproduire le comportement du système à modéliser. Il consiste en la recherche d'un jeu de paramètres (les poids synaptiques) et peut s'effectuer de deux manières : supervisée (le réseau utilise les données d'entrée et de sortie du système à modéliser) et non supervisée (seules les données d'entrée du système sont fournies et l'apprentissage s'effectue par comparaison entre exemples).

Lors de la phase dite de test, de nouveaux exemples sont présentés au réseau de neurones artificiels pour juger de sa capacité à prédire les comportements du système modélisé.

3.3.2.2.3 L'analyse qualitative des tendances

L'analyse qualitative des tendances est une des principales techniques pour la supervision et le diagnostic [Cheung et Stephanopoulos, 1990]. Elle a pour but l'analyse et la représentation qualitative des tendances du système physique. Pour cela, les signaux sont exploités à travers leurs dérivées du premier, voire du second ordre correspondant à l'information de type tendance obtenue par les mesures issues de capteurs.

Elle est constituée de deux étapes principales :

- l'identification des tendances dans les mesures,
- et l'interprétation de ces tendances en termes de scénarios.

L'abstraction qualitative permet une représentation compacte de la tendance en ne prenant en compte que les événements significatifs [Kempowski, 2004]. Elle fournit des informations satisfaisantes pour l'aide au diagnostic et à l'interprétation du comportement du système physique. En effet, généralement un comportement défaillant conduira à une tendance ostensible des signaux issus des

capteurs. Ces tendances sont exploitées pour l'identification des situations anormales ; ceci permet la détection au plus tôt d'une défaillance [Vedam *et al.*, 1997b].

3.4 LES MÉTHODES À BASE DE MODÈLES PHÉNOMÉNOLOGIQUES

Les approches à base de modèles phénoménologiques ont largement été étudiées dans la littérature [Willisky, 1976 ; Chow et Willisky, 1984 ; Basseville, 1988 ; Patton *et al.*, 1989 ; Gentil, 1997 ; Isermann, 1997 ; Maquin *et al.*, 1993, 1997 ; Fussel et Isermann, 1998]. Elles s'appuient sur des modèles explicites du système physique étudié. Elles ont pour principe de comparer les mesures effectuées sur le système aux informations fournies par le modèle.

3.4.1 Les modèles quantitatifs

Diverses approches pour la détection de défaillances à partir des modèles quantitatifs ont été développées depuis les années soixante-dix [Isermann, 1984]. Le *modèle quantitatif* est un modèle mathématique représentant le comportement du système ou du sous-système, c'est-à-dire un dispositif ou un composant.

Le modèle peut être statique ou dynamique, linéaire ou non linéaire. Il est utilisé pour générer des indicateurs de défaut, appelés résidus. Ces indicateurs représentent des symptômes de comportement anormal du système physique, c'est-à-dire une divergence entre le comportement réel et celui prévu par le modèle de référence. Franck décompose ces méthodes en trois grands groupes [Franck, 1998] :

- les approches par espace de parité,
- les approches d'estimation de paramètres,
- et les approches d'estimation d'état.

3.4.1.1 Les relations de parité

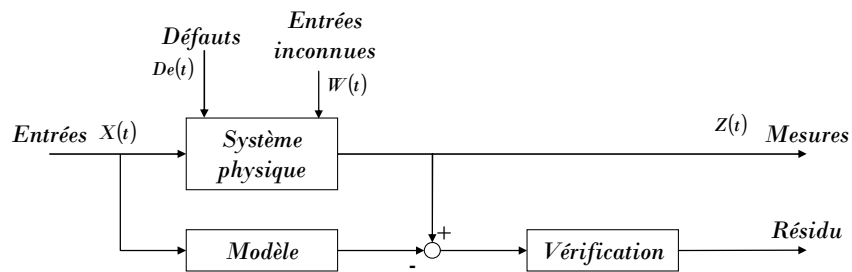


Figure 3.13 Les relations de parité [Touaf, 2005]

La relation de parité est une des premières méthodes utilisées. Le principe est basé sur la vérification de la cohérence existant entre les entrées et les sorties du système surveillé. Les équations du modèle sont projetées dans un espace particulier, dit espace de parité. Cette transformation permet alors

d'éliminer les inconnues à l'aide de redondances. Les équations projetées sont appelées équations de redondance analytique. Elles testent la cohérence des mesures par rapport à leurs estimées fournies par le modèle.

D'après [Ripoll, 1999], la définition formelle d'une relation de parité est la suivante :

Définissons tout d'abord le modèle général d'équations de mesure ;

$$Z(t) = H \cdot X(t) + W(t) + Dir \cdot De(t) \quad (3 - 5)$$

Où Z est le vecteur des mesures, $Z \in \mathfrak{R}^m$, avec m le nombre de mesures,

X est le vecteur des variables à mesurer, $X \in \mathfrak{R}^n$, avec n le nombre de variables

De est le vecteur des défaillances, $De \in \mathfrak{R}^d$, où d est le nombre de défaillances du système, considérées

W est la vecteur des bruits de mesures, $W \in \mathfrak{R}^m$,

H est la matrice caractérisant le système de mesure, $H \in \mathfrak{R}^{m \times n}$,

et Dir est la matrice qui traduit la direction des défaillances, $Dir \in \mathfrak{R}^{m \times d}$.

Le vecteur de parité P est défini par projection des mesures Z :

$$P(t) = P_r \cdot Z(t) \quad (3 - 6)$$

Où P_r est la matrice de projection.

La propriété d'orthogonalité de cette matrice avec la matrice H entraîne :

$$P(t) = P_r \cdot W(t) + P_r \cdot Dir \cdot De(t) \quad (3 - 7)$$

Le vecteur de parité est défini comme la somme de deux composantes :

- Le premier terme $P_r \cdot W(t)$ apporte une contribution aléatoire, de statistique connue si celle du bruit $E(t)$ est connue ;
- Le second terme $P_r \cdot Dir \cdot De(t)$ est une composante déterministe dont la direction est fixée par le rang du capteur en défaut, et l'amplitude est fonction de celle de la défaillance du capteur.

En l'absence d'erreurs de mesure $W(t)$ et de défaillances $De(t)$, le vecteur de parité devient nul.

L'ensemble des redondances entre les mesures se traduit par :

$$P_r \cdot Z(t) = 0 \quad (3 - 8)$$

W ne découple pas parfaitement les entrées inconnues. Il est donc nécessaire de travailler sur l'observabilité.

3.4.1.2 L'estimation de paramètres

Les premières applications de cette approche ont été faites par Isermann [Isermann, 1984, 1993, 1994 ; Isermann et Freyermuth, 1991]. Cette approche estime les paramètres du modèle, c'est-à-dire les constantes physiques du système (température, pression...) ou une agrégation de plusieurs paramètres physiques. On mesure l'influence des défauts sur les paramètres et non plus, comme précédemment, sur les variables du système physique. Le principe est donc d'estimer en continu des paramètres du procédé en

utilisant les mesures d'entrée/sortie et ensuite évaluer la distance qui les sépare des valeurs de référence de l'état normal du procédé.

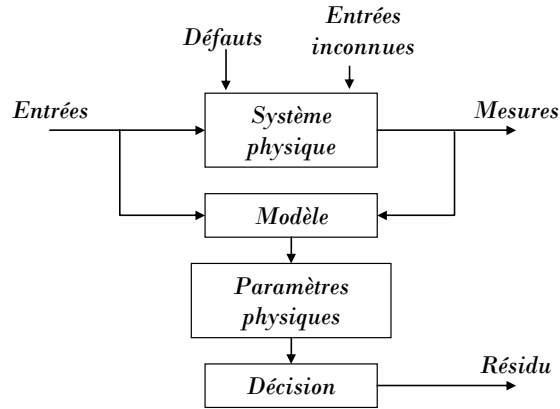


Figure 3.14 L'estimation de paramètres [Frank, 1996]

L'estimation de paramètres comprend cinq grandes étapes [Isermann, 1984] :

- La modélisation mathématique du procédé selon des équations du type :

$$Z(t) = f(u(t), \theta) \quad (3 - 9)$$

Où $u(t)$ représente les commandes du système et θ les paramètres du modèle ;

- La description des relations entre les constantes physiques p supposées connues et les paramètres du modèle θ :

$$\theta = g(p) \quad (3 - 10)$$

- L'estimation $\hat{\theta}$ des paramètres du modèle à partir de l'équation (3 - 10) et des mesures des entrées et des sorties du système :

$$\hat{\theta} = h(Y, U) \quad (3 - 11)$$

Avec $Z = \{Z(1), \dots, Z(t)\}$ et $U = \{U(1), \dots, U(t)\}$;

- L'estimation \hat{p} des paramètres du système à partir de l'équation (3 - 10) :

$$\hat{p} = g^{-1}(\hat{\theta}(t)) \quad (3 - 12)$$

- La théorie de la décision pour détecter une défaillance ou non, en prenant en compte les aspects non déterministes (bruits de capteurs ou erreurs de modélisation). Le vecteur résidu est obtenu en faisant la différence entre les paramètres estimés et les valeurs nominales, en fonction des paramètres du modèle ou des paramètres physiques :

$$r(t) = \theta(t) - \hat{\theta}(t) \quad \text{ou} \quad r(t) = p(t) - \hat{p}(t) \quad (3 - 13)$$

3.4.1.3 L'estimation d'état

Les méthodes d'estimation d'état ont pour principe de reconstruire les états et les sorties du système à partir des entrées et des sorties. Par opposition aux méthodes en boucle ouverte (espace de parité), cette stratégie fonctionne en boucle fermée (cf. figure 3.15).

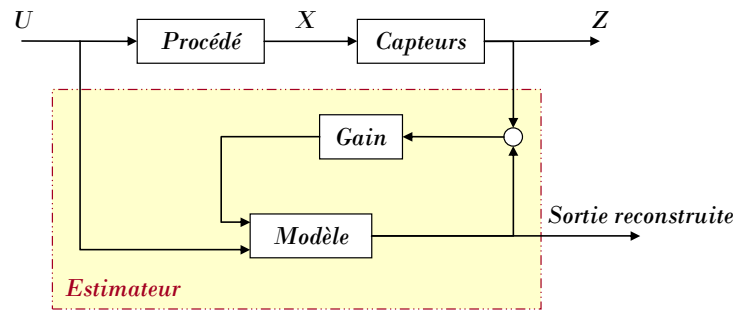


Figure 3.15 L'estimation d'état

Ce sont des méthodes de génération de résidus indirectes : elles calculent l'erreur d'estimation de la sortie. L'estimation de l'état peut être effectuée :

- dans le cas déterministe à l'aide d'observateurs : par exemple dans le domaine fréquentiel, les observateurs généralisés ou dans le domaine temporel, les observateurs de Luenberger [Luenberger, 1971 ; Adjallah, 1993]
- ou dans le cas stochastique à l'aide de filtres : par exemple filtre de Kalman [Willsky, 1976] ou de filtres détecteurs de défauts [Massoumnia, 1986].

Présentons, par exemple, l'observateur de Luenberger, dans le cas d'un système linéaire :

$$\dot{X} = F \cdot X + G \cdot U \tag{3 - 14}$$

$$Z = H \cdot X \tag{3 - 15}$$

Un observateur dynamique à la forme suivante :

$$\dot{\hat{X}} = F \cdot \hat{X} + G \cdot U + K \cdot (Z - \hat{Z}) \text{ avec } \hat{Z} = H \cdot \hat{X} \tag{3 - 16}$$

La correction de l'évolution de l'état grâce au modèle est fonction de l'écart constaté entre la sortie observée et la sortie reconstruite par l'observateur : $(Z - \hat{Z})$.

On peut réécrire l'observateur de la manière suivante :

$$\dot{\hat{X}} = (F - K \cdot H) \hat{X} + G \cdot U + K \cdot Z \tag{3 - 17}$$

K est la matrice de gain et doit être choisie de façon à ce que $(X - \hat{X}) \rightarrow 0$.

Pour cela, il suffit de choisir K telle que la matrice $(F - K \cdot H)$ soit une matrice Hurwitz, c'est à dire que ses valeurs propres soient à parties réelles négatives dans le cas continu ou possèdent un module inférieur à 1 dans le cas discret.

Ces deux types de méthodes présentent des analogies dans leur formulation et leur fonctionnement peut être représenté par la figure 3.15. La différence réside dans le mode de calcul des paramètres du reconstruteur en fonction du contexte choisi (cas continu ou discret, déterministe ou stochastique).

La théorie des estimations d'état est applicable aux systèmes linéaires ou non linéaires. Si l'intégralité du vecteur d'état est estimée, c'est-à-dire si le système est complètement observable, on parle d'estimation d'ordre plein, dans le cas contraire d'ordre réduit.

Frank et Wünnenberg définissent une classe d'observateurs dit observateurs à entrées inconnues dont les sorties du système sont indépendantes des incertitudes structurées auxquelles il est soumis [Frank et Wünnenberg, 1989 ; Ripoll, 1999]. Des travaux de Ding traitent d'une approche unifiée de méthodes à base d'observateurs [Ding et Frank, 1994]. Dans [Venkatasubramanian *et al.*, 2003], l'utilisation de banc d'observateurs pour estimer les variables inconnues est exposée. Cette méthode est constituée d'un ensemble d'observateurs, chacun sensible à un ensemble de défauts et insensibles aux autres. Les résidus générés caractérisent chaque défaut, ce qui facilite la localisation.

La méthode de génération de résidus dépend du type de modèle et de procédé ou du type de défaut que l'on souhaite détecter (défauts additifs, multiplicatifs, multiples ou pas). Deux types de résidus sont distingués : les résidus structurés où la propagation d'un défaut affecte seulement une partie des résidus [Chen *et al.*, 1995] et les résidus directionnels, pour lesquels la présence d'un défaut entraîne le vecteur résidu dans une direction fixe [Gertler, 1992].

3.4.2 Les modèles qualitatifs ou semi-qualitatifs

Dans certains cas, il est difficile de disposer des connaissances complètes pour construire un modèle analytique du système physique en raison de sa complexité structurelle ou de son comportement. Pour remédier à ce problème, des méthodes à base de modèle qualitatif (ou semi-qualitatif) ont été mis en place [Travé-Massuyès et Milne, 1997]. De plus, les systèmes à base de modèles quantitatifs et les approches de classification ont une capacité explicative quasi-inexistante. La stratégie consiste alors en l'établissement de relations de causes à effets. Ces méthodes sont basées sur le raisonnement qualitatif et décrivent la structure du système physique étudié (connexions entre les composants) [Kempowski, 2004]. Ces dernières années, l'intérêt pour ces approches a eu un enclin certain [Travé-Massuyès *et al.*, 1997 ; Escobet *et al.*, 2001]. En effet, elles ont une capacité explicative et adductive nécessaire au diagnostic et notamment à la localisation [Rocha-Loures, 2006].

Le raisonnement qualitatif exprime le lien entre un composant et les formules décrivant son comportement, décrit donc qualitativement l'effet ou l'influence qu'ont les entités du système les unes sur les autres, les entités pouvant représenter aussi bien des variables que des défauts.

Le diagnostic basé sur la théorie du raisonnement logique utilise les termes *suspect* et *candidat* :

- Un suspect fait référence à tout composant que le système de diagnostic a identifié comme responsable possible de l'apparition d'une divergence.

- Un candidat fait référence au composant ou ensemble de composants dont le fonctionnement incorrect expliquerait toutes les divergences observées.

L'approche est composée de trois phases :

- la détection de conflits : recherche de l'ensemble de composants suspects en fonction des divergences observées
- le diagnostic : recherche des candidats parmi les suspects
- la discrimination des hypothèses : raffinement de l'ensemble de candidats dans le cas où il y en aurait plusieurs.

Parmi les méthodes les plus populaires, citons :

3.4.2.1 Les équations différentielles qualitatives

Le raisonnement qualitatif met en œuvre des méthodes symboliques qui permettent d'intégrer ces différentes caractéristiques. Les équations différentielles qualitatives font partie des méthodes qui utilisent des modèles physiques qualitatifs pour la détection et le diagnostic de fautes. Elles sont constituées de variables, de l'espace des quantités associées à ces variables, de contraintes et de l'ensemble de transition. Elles prédisent le comportement du système physique. Ce sont des abstractions des équations différentielles ordinaires : basées sur un système de contraintes qualitatives, elles visent à généraliser les équations ordinaires. Elles utilisent des intervalles de variables et de paramètres. On distingue trois grandes étapes pour ces approches :

- la description du mécanisme physique,
- la construction du modèle,
- et enfin la résolution.

La résolution s'effectue grâce à l'utilisation d'un algorithme qui détermine le comportement du système en n'utilisant aucune information précise sur les paramètres ni même sur les relations fonctionnelles existantes. Citons l'exemple de l'algorithme QSIM [Kuipers, 1985, 1986] qui représente le comportement du système par une séquence d'états en simulant le modèle composé d'équations différentielles qualitatives.

3.4.2.2 Les graphes d'influence/causaux

C'est un outil proposé par l'Intelligence Artificielle. Il détermine les composants défectueux qui peuvent expliquer le fonctionnement anormal observé. Les mécanismes sont expliqués en termes de causalité nécessaire au moment de l'analyse des régimes perturbés. La structure causale décrit les effets que les variables peuvent avoir les unes sur les autres.

Le diagnostic basé sur un graphe d'influence tend à identifier la variable source dont la déviation est suffisante pour expliquer toutes les déviations détectées sur d'autres variables [Travé-Massuyès *et al.*, 1997]. L'algorithme de diagnostic est, en général, une procédure de recherche avant/arrière ayant pour origine la variable incohérente. La recherche arrière permet de réduire l'espace de défaut en supprimant les observations cohérentes en amont du chemin causal. À chaque déviation, dite primaire, une hypothèse est

générée. Cette hypothèse conduit à une recherche avant qui permettra de déterminer la cause de la déviation de comportement.

Les *digraphes* sont une autre représentation de l'information causale [Iri *et al.*, 1979]. La représentation du système est un graphe orienté où les nœuds représentent les variables, et où les arcs représentent les relations normales et déterministes entre les variables [Gentil *et al.*, 2004]. La représentation graphique pour la visualisation de la propagation d'un défaut est obtenue par une analyse physique soignée et une analyse fonctionnelle descendante du processus.

Deux types principaux de structure causale sont proposés [Rocha-Loures, 2006] :

- le digraphe représentant la calculabilité issue de la connaissance des relations mathématiques (équations différentielles ou à différences) [Iwasaki et Simon, 1986 ; Pons et Travé-Massuyès, 1997 ; Gehin *et al.*, 2000 ; Blanke *et al.*, 2003].
- le digraphe représentant une connaissance fonctionnelle du procédé; les nœuds sont liés aux variables considérées significatives par la surveillance et les arcs sont liés aux phénomènes physiques [De Kleer, 1986].

Issu d'une connaissance profonde des mécanismes physiques, le *bond-graph* est le modèle de base de nombreuses approches de diagnostic. Le diagnostic des défauts abrupts y est considéré. Historiquement l'approche a été plus particulièrement envisagée pour la modélisation de la dynamique de systèmes mécaniques et électroniques [Rocha-Loures, 2006]. C'est une représentation graphique d'un système dynamique physique (mécanique, électrique, hydraulique, pneumatique...). Elle ne nécessite pas l'écriture de lois générales de conservation. Elle repose essentiellement sur la caractérisation des transferts d'énergie dans le système.

3.4.3 Les modèles flous

Les systèmes flous utilisent les concepts de la théorie des ensembles flous et de la logique floue, introduite par Zadeh [Zadeh, 1965]. Un modèle flou est une représentation des caractéristiques d'un système à l'aide de règles floues qui décrivent son comportement. Le but est de généraliser l'information imitant le raisonnement approximatif exécuté par l'homme en introduisant l'imprécision. Les systèmes à base de règles floues sont des systèmes où les relations entre variables sont représentées au moyen de règles floues de la forme : SI prémisse ALORS conclusion. La décision en logique floue est basée sur la notion d'expertise, qui permet de quantifier le flou à partir de connaissance a priori ou acquise antérieurement.

3.4.3.1 La logique floue

Un univers de discours possède des fonctions d'appartenance des variables d'entrée et de sortie à des classes. Ces fonctions peuvent avoir différentes formes, les plus usuelles étant des formes triangulaires, trapézoïdales et gaussiennes. Un ensemble flou est totalement déterminé par sa fonction d'appartenance. Soit U l'univers de discours, A un sous ensemble de U , la fonction d'appartenance μ_A de l'ensemble flou A est définie comme suit :

$$\forall x \in U, \mu_A(x) \in [0;1] \quad (3 - 18)$$

$$\forall x \in U, \mu_A(x) = 0 \text{ si } x \notin A \quad (3 - 19)$$

La figure 3.16 donne un exemple de la caractérisation d'un ensemble flou. La logique floue est basée sur des variables floues dites variables linguistiques à valeurs linguistiques dans l'univers du discours U . L'univers du discours est dans notre exemple la gamme de température de 0°C à 40°C, la variable linguistique est la température et les valeurs linguistiques sont « froide », « tiède » et « chaude ».

La fuzzification est l'étape qui consiste en la quantification floue des valeurs réelles d'une variable. C'est une phase délicate du processus mis en œuvre par la logique floue. Elle est souvent réalisée de manière itérative et requiert de l'expérience.

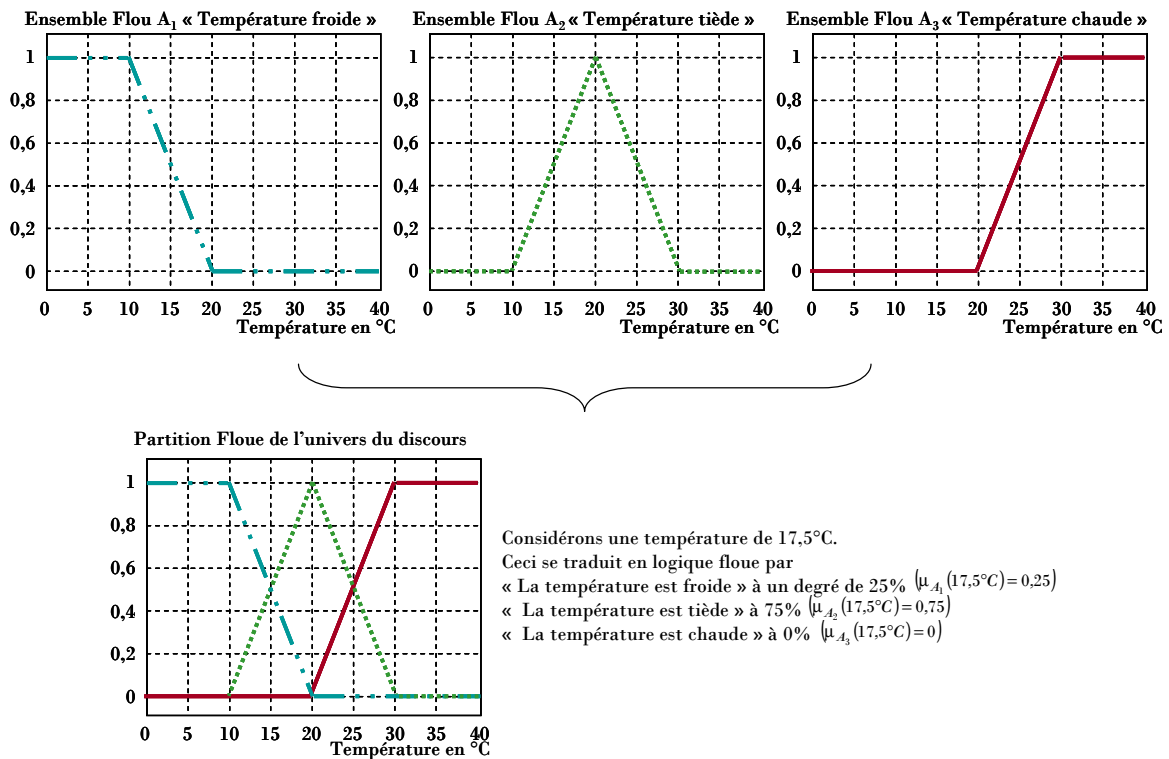


Figure 3.16 La fonction d'appartenance

Deux types de modèle exploitent ce formalisme [Kempowski, 2004] :

- Les modèles linguistiques flous ou modèles de Mamdani,
- Les modèles flous de type Takagi-Sugeno-Kang.

3.4.3.2 Les modèles linguistiques flous

Ces modèles décrivent le système à partir des règles SI-ALORS où, les prémisses et les conclusions utilisent des variables linguistiques qui sont des ensembles flous comme valeurs. L'ensemble de règles prend la place de l'ensemble d'équations classique utilisé pour caractériser un système. L'expression qualitative est établie à partir de la connaissance des experts et des différents mécanismes de raisonnement.

Une règle SI-ALORS s'écrit de la façon suivante :

$$R_i : SI \ x \in A_i \ ALORS \ y \in B_i \quad \forall i \in [1;n] \quad (3 - 20)$$

Où A_i et B_i sont les ensembles flous qui définissent le partitionnement des espaces d'entrée et de sortie.

3.4.3.3 Les modèles flous type Takagi-Sugeno-Kang

Les prémisses des règles sont des variables linguistiques, mais les conclusions sont de type numérique, généralement sous forme de constantes, de polynômes ou de fonctions (linéaires ou non-linéaires) qui dépendent des entrées. Une règle SI-ALORS s'écrit dans ce formalisme comme suit :

$$R_i : SI \ x \in A_i \ ALORS \ y = f_i(x) \quad \forall i \in [1;n] \quad (3 - 21)$$

Où A_i et B_i sont les ensembles flous qui définissent le partitionnement des espaces d'entrée et de sortie.

3.5 LA SYNTHÈSE DES MÉTHODES

Notons tout d'abord, que de nombreuses méthodes sont nées du couplage de plusieurs théories. Ainsi, évoquons en particulier les méthodes de classification floue. Ces méthodes s'appuient sur une description des classes sous forme d'ensembles flous : Une observation est alors décrite par son degré d'appartenance aux différentes classes. Une classe peut ici représenter un état normal ou un état défaillant. Dans ce cas, l'ensemble des degrés d'appartenance est à rapprocher de la distribution de probabilité obtenue lors d'une classification probabiliste (de type bayésien). Citons parmi les méthodes les plus connues celles des K-moyennes (K-means) [MacQueen, 1967 ; Duda et Hart, 1973 ; Babuska, 1998], des C-moyennes floues (Fuzzy-C-means) [Bezdek, 1981 ; Gustafson, 1979], ou encore LAMDA [Aguilar-Martin et Lopez, 1982, Desroches, 1987 ; Piera et Aguilar-Martin, 1989].

Le tableau 3.3 propose une analyse comparative synthétique des approches de détection et diagnostic que nous venons de présenter. Parmi toutes ces méthodes, nous retiendrons l'approche à base de modèle et en particulier nous concentrerons sur les modèles quantitatifs.

La section suivante est consacrée à la formulation du problème et l'utilisation d'une approche à base de modèle quantitatif.

| Modèle | Type | Approches | Méthodes | Référence | Principes | Avantages | Inconvénients |
|--------|-------------------------------------|----------------------------|--|---|--|--|--|
| Aucun | Les méthodes à base de connaissance | Les arbres de défaillances | AMDE - AMDEC Les arbres de défaillances Les arbres de causes | [Recht, 1966 ; Villemeur, 1988 ; Zwingelstein, 1995] | Approches basées sur une connaissance du système, de ses défauts et de leurs effets associés | <ul style="list-style-type: none"> • Aucun traitement nécessaire • Très efficace pour l'analyse de défaillances simples d'éléments conduisant à la défaillance globale du système • Démarche très puissante pour la résolution des problèmes de diagnostic de défaillances des procédés industriels • Très répandue dans de nombreux domaines • Identification des modes communs de défaillances pouvant affecter le système étudié, c'est-à-dire les événements qui provoquent simultanément des états de panne sur plusieurs composants du système. | <ul style="list-style-type: none"> • Recensement préalable non exhaustif des défaillances et de leurs relations éventuelles → longue expérience • Toute modification ou évolution du système nécessite la réécriture du tableau • Cas de défaillances multiples non traités • Mal adaptée aux systèmes dynamiques, dépendants fortement du temps, à cause du nombre élevé de variables, processus, points de fonctionnement à analyser • Développement sensible aux erreurs à différentes étapes : arbre construit dépend de son créateur • Pas de méthode formelle pour vérifier l'exactitude de l'arbre développé • Basées sur un grand nombre d'hypothèses (faites par un expert → subjectif) → à un diagnostic incertain voire erroné |
| | | Les systèmes experts | | [Henley, 1984 ; Chester <i>et al.</i> , 1984 ; Niida, 1985] | <ul style="list-style-type: none"> • Associations empiriques entre effets et causes représentées par des règles basées sur expérience de l'expert plutôt que sur connaissance de structure et comportement du système • Parcours des règles par techniques classiques IA telles que le chaînage avant arrière ou mixte | <ul style="list-style-type: none"> • Efficacité au niveau temps de calcul • Aucun raisonnement compliqué et coûteux en tps de calcul → pas de calcul intermédiaire • Règles = produits d'expert humain donc résultat compréhensible pour opérateur • Implantation très simple car outils de génération de systèmes experts (chronos) | <ul style="list-style-type: none"> • Dépend de l'expertise faite sur le système • Difficultés d'acquisition de l'expertise • Manque de généricité car règles dépendent de l'architecture du système • Problème de l'évolution du système : ajout changement de composant → nouvelle expertise • Robustesse : règles fixées et non robustes face aux situations non reconnues • Données incertaines : difficultés dans analyse d'un ensemble de données non corrélées incomplètes et ambiguës • Manque de connaissances profondes : pas d'explications sur les conclusions adoptées • Incohérence des règles : ajout ou suppression de règle → impact sur autres règles difficile à détecter |

Tableau 3.3 a L'Analyse comparative des méthodes dédiées à la détection et au diagnostic de fautes : tableau récapitulatif

| Modèle | Type | Approches | Méthodes | Référence | Principes | Avantages | Inconvénients |
|----------------|--|------------------------------|--|--|--|---|---|
| Comportemental | Les méthodes basées sur le traitement de données | L'extraction statistique | L'analyse en composante principale Les moindres carrés partiels | [Nomikos et MacGregor, 1994 ; MacGregor et Kourti, 1995 ; Jolliffe, 1986 ; Diamantaras et Kung, 1996 ; Harkat, 2003] | <ul style="list-style-type: none"> • Identification des variables sous-jacentes, facteurs, qui expliquent les corrélations à l'intérieur d'un ensemble de variables observées | <ul style="list-style-type: none"> • Technique descriptive permettant d'étudier les relations qui existent entre les variables sans tenir compte a priori d'une quelconque structure • Outil puissant capable de compresser des données et de réduire leur taille de telle sorte que l'information est conservée • Facilite l'analyse des données • Manipulation du bruit et corrélation pour extraire l'information efficacement | <ul style="list-style-type: none"> • Représentation invariante dans le temps [possible extension au cas dynamique [Qin et al., 1999 ; Verbeek, 2000] • Mise à jour périodique • Aucune propriété de signature ce qui rend difficile l'isolation de faute |
| | | | La classification reconnaissance de forme | [Bishop, 1995 ; Duda et Hart, 1973] | <ul style="list-style-type: none"> • Reconnaissance d'une forme parmi différentes possibilités à partir d'observations bruitées | <ul style="list-style-type: none"> • Simple à appliquer • Temps de calcul faible pour la classification d'une nouvelle observation et indépendant de la dimension de l'ensemble d'apprentissage • Capable de traiter des données qui sont à la fois incertaines et imprécises | <ul style="list-style-type: none"> • Pas de règles générales pour le choix de l'espace de représentation qui est le facteur de réussite de diagnostic • Nombre de classes supposé connu au départ → connaissance supposée exhaustive • Toute observation doit appartenir à une classe définie • Les caractéristiques de certains modes de fonctionnement restent inconnues • Nécessité d'utiliser des techniques qui permettent de prendre en compte la détection de nouvelles classes, d'apprendre leurs caractéristiques et de les inclure dans le système de diagnostic |
| | | L'extraction non statistique | L'analyse fréquentielle | [Flandrin, 1998] | <ul style="list-style-type: none"> • Signaux analysés en état normal de fonctionnement • Hautes fréquences reliées au bruit et Basses fréquences reliées aux évolutions propres de l'état du procédé | <ul style="list-style-type: none"> • Utile pour analyser des signaux qui montrent des oscillations avec des périodes longues (débit, courant électrique, pression, ...) • Relativement simple à mettre en place | <ul style="list-style-type: none"> • Sensibilité aux bruits de mesure quand ceux-ci coïncident avec la zone fréquentielle d'intérêt • Nécessité d'un échantillonnage fréquent pour la reconstitution du signal tout en minimisant les pertes de fréquence |

Tableau 3.3.b L'Analyse comparative des méthodes dédiées à la détection et au diagnostic de fautes : tableau récapitulatif

| Modèle | Type | Approches | Méthodes | Référence | Principes | Avantages | Inconvénients |
|------------------|--|------------------------------|--------------------------------------|--|--|---|--|
| Comportemental | Les méthodes basées sur le traitement de données | L'extraction non statistique | Les réseaux de neurones artificiels | [McCulloch et Pitts, 1943] | <ul style="list-style-type: none"> • Technique non linéaire complexe basée sur le modèle d'un neurone humain • Ensemble d'entités simples, neurones, interconnectées entre elles • Approximation de la relation inconnue reliant les symptômes aux défaillances | <ul style="list-style-type: none"> • Faible sensibilité aux bruits de mesure, • Capacité à résoudre des problèmes non linéaires et multi-variables • Capacité à stocker la connaissance de manière compacte • Capacité à apprendre en ligne et en temps réel | <ul style="list-style-type: none"> • Temps de calcul important pendant la phase d'apprentissage • Définition des paramètres du réseau délicate • Ne fournit pas une représentation explicite des connaissances : modèle implicite, caché dans la structure du réseau et les poids des connexions entre neurones • Ne permet pas de travailler avec des informations de type qualitatif |
| | | | L'analyse qualitative de la tendance | [Cheung et Stephanopoulos, 1990 ; Dash <i>et al.</i> , 2003 ; Sundarraman et Srinivasan, 2003 ; Rengaswamy <i>et al.</i> , 2001 ; Neves et Martin, 2000] | <ul style="list-style-type: none"> • Technique basée sur l'analyse et la représentation qualitative des tendances du système physique • deux étapes principales : • L'identification des tendances dans les mesures, • Et l'interprétation de ces tendances en terme de scénarios. | <ul style="list-style-type: none"> • Méthode adaptée aux incertitudes, incomplétude et hétérogénéité des données • Raisonnement temporel adéquat aux informations instantanées mais aussi au comportement historique • Adaptée aux procédés complexes (grand volume de données) grâce à sa représentation symbolique • Représentation de la connaissance adaptée à l'interprétation de l'opérateur humain | <ul style="list-style-type: none"> • Observation délicate des dérivées • Nécessite la mise en œuvre d'une étape de filtrage de bruit, d'un traitement de discontinuités • Synchronisation du dictionnaire des tendances (signatures) et des observations non triviale → Erreurs d'interprétation et d'inférence au niveau du diagnostic • Nécessite une description exhaustive des modes du procédé et des comportements de défauts • Approche inappropriée dans le cas de dérivées lentes [Dash et al. 2003] |
| Phénoménologique | Quantitatif | L'espace de parité | | [Chow, 1980 ; Chow et Willsky, 1984 ; Patton et Chen, 1991a,b ; Lou, <i>et al.</i> , 1986 ; Potter et Suman, 1977 ; Ragot et Maquin, 1996, 1998] | <ul style="list-style-type: none"> • Reconstruction de la sortie du modèle • Équations du modèle projetées dans un espace particulier : espace de parité • Élimination des inconnues grâce aux relations de redondance analytique | <ul style="list-style-type: none"> • Méthode très générale • Facilite l'analyse des données : isolation de la défaillance • Vérifie les données issues du système • Connaissance du système découplée de la connaissance du diagnostic • Moindre coût de développement • Structure intéressante pour les procédés complexes | <ul style="list-style-type: none"> • Peu développée dans un cadre incertain • Prise en compte d'incertitudes additives • Rejet des incertitudes par découplage • Mal adaptée pour les non linéarités, les variables non observées et les pannes non additives |

Tableau 3.3.c L'Analyse comparative des méthodes dédiées à la détection et au diagnostic de fautes : tableau récapitulatif

| Modèle | Type | Approches | Méthodes | Référence | Principes | Avantages | Inconvénients |
|------------------|-------------|--|--|--|--|---|--|
| Phénoménologique | Quantitatif | L'estimation paramétrique | L'estimation par projection orthogonale L'estimation Bayésienne L'estimation au sens du maximum de vraisemblance L'estimation au sens des moindres carrés | [Isermann, 1984, 1993 ; Isermann et Freyermuth, 1991] | <ul style="list-style-type: none"> • Estimer les paramètres du modèle : constantes physiques du modèle ou agrégation de plusieurs paramètres physiques • Correspondance unique entre paramètres modèle et paramètres système | <ul style="list-style-type: none"> • Apporte de l'information sur l'ampleur des déviations • Méthode bien adaptée aux défauts multiplicatifs (ayant une incidence sur les paramètres) • Application à des procédés à faible nombre de variables où des modèles précis peuvent être définis | <ul style="list-style-type: none"> • Nécessité d'avoir un système physique excité en permanence : Problème dans le cas de procédés dangereux ou fonctionnant en mode stationnaire • Relations entre les paramètres mathématiques et physiques pas toujours inversibles de façon unitaire → tâche de localisation du diagnostic difficile • Utilisation difficile pour des installations complexes due au grand nombre de variables mis en jeu et des nombreux modes de fonctionnement |
| | | L'estimation d'état | Les observateurs Les filtres | [Luenberger, 1971 ; Adjallah, 1993 ; Willsky, 1976 ; Massoumnia, 1986] | <ul style="list-style-type: none"> • Reconstruction des états et des sorties du système à partir des entrées et des sorties : variables et paramètres • Génération de résidus | <ul style="list-style-type: none"> • Applicable aux systèmes linéaires et non linéaires • Répandue car génération de résidus flexibles • Méthodes très répandues • Calcul simple • Possibilité de découpler les réponses des entrées inconnues • Méthode plus robuste aux bruits de mesures | <ul style="list-style-type: none"> • Nécessité d'avoir un modèle précis et complet • Mal adaptée au processus complexe • Erreur de diagnostic due aux perturbations • Pas de garantie de détection si le type de défaut n'a pas été modélisé • Adaptabilité difficile aux changements de processus et Manque de méthode générale dû au caractère local du modèle (appliqué au système étudié) |
| | Qualitatif | Les équations différentielles qualitatives | QSIM QDE | [Kuipers, 1986 ; Venkatasubramanian <i>et al.</i> , 2003] | <ul style="list-style-type: none"> • Technique permettant de produire des descriptions qualitatives de comportements de systèmes imparfaitement connus | <ul style="list-style-type: none"> • Capacité à établir un diagnostic à partir de connaissance incomplète et incertaine du système • Capacité à raisonner sur le comportement dynamique du système | <ul style="list-style-type: none"> • Description structurale du système nécessaire • Peut conduire a un diagnostic ambiguë voire erroné |

Tableau 3.3.d L'Analyse comparative des méthodes dédiées à la détection et au diagnostic de fautes : tableau récapitulatif

| Modèle | Type | Approches | Méthodes | Référence | Principes | Avantages | Inconvénients |
|------------------|------------|---|-----------------------------------|--|---|--|---|
| Phénoménologique | Qualitatif | Les graphes d'influence | Les digraphes Les bond-graphes | [De Kleer, 1986 ; Rahmani <i>et al.</i> , 1997 ; Travé-Massuyès <i>et al.</i> , 1997 ; Vedam et Venkatasubramanian, 1997a] | <ul style="list-style-type: none"> • Description des effets que les variables peuvent avoir les unes sur les autres par la structure causale | <ul style="list-style-type: none"> • Connaissance de comportements (modèles) de défaut non nécessaire pour la localisation • Capacité explicative • Exploitation au maximum de la structure • Langage graphique unique applicable à de nombreux systèmes • Repose uniquement sur le parcours de chemins causaux | <ul style="list-style-type: none"> • Génération d'un grand nombre d'hypothèses pouvant conduire à un diagnostic erroné • Construction de la structure peut entraîner des pertes d'information |
| | Flou | Les modèles linguistiques flous | | [Mamdani, 1977 ; Nakoula, 1997] | <ul style="list-style-type: none"> • Système à partir des règles SI-ALORS • Utilisation de variables linguistiques | <ul style="list-style-type: none"> • Pouvoir explicatif dû à l'utilisation de variables linguistiques → conclusion facilement interprétable • Possibilité d'inclure des connaissances humaines exprimées sous forme de règles linguistiques • Capacité à utiliser des entrées floues • Interpolation contrôlable | <ul style="list-style-type: none"> • Peut conduire à des modèles inappropriés ou non optimaux • Capacité de représentation limitée • Nombre de paramètres à identifier élevé • Nombre de données d'identification important • Certains états non atteints lors de l'acquisition de données |
| | | Les modèles flous type Takagi-Sugeno-Kang | | [Takagi et Sugeno, 1985] | <ul style="list-style-type: none"> • Système à partir des règles SI-ALORS • Les prémisses des règles : des variables linguistiques • Les conclusions : de type numérique | <ul style="list-style-type: none"> • Description de la structure physique connue d'un système • Approximation du comportement d'un système complexe • Capacité de représentation importante • Nécessite moins de données que l'approche de Mamdani • Variable de sortie directement liée aux variables d'entrée | <ul style="list-style-type: none"> • Règle n'ayant pas de signification sémantique claire • Ne fournit pas une meilleure compréhension de la relation entre les variables • Précision globale du modèle réduite par l'interpolation des modèles locaux • Incapacité à intégrer l'expérience humaine formulée par des règles linguistiques • Incapacité à utiliser directement des entrées floues |

Tableau 3.3.e L'Analyse comparative des méthodes dédiées à la détection et au diagnostic de fautes : tableau récapitulatif

3.6 LA FORMULATION DU PROBLÈME

L'analyse comparative décrite dans la section précédente a permis de mettre en évidence les points forts et faibles de chaque approche de surveillance. Cette section vise à présenter de manière plus détaillée la théorie sur laquelle nos travaux s'appuient.

3.6.1 La surveillance à base de modèles

Lorsque la sortie prédite suit les variations de l'entrée, l'utilisation d'un modèle est un outil adapté à la surveillance dynamique, en particulier lors des changements de points de fonctionnement [Boillereaux et Flaus, 2003]. Les paragraphes suivants ont pour objectif de décrire plus en détails la formulation adoptée pour le problème de surveillance.

3.6.1.1 La structure

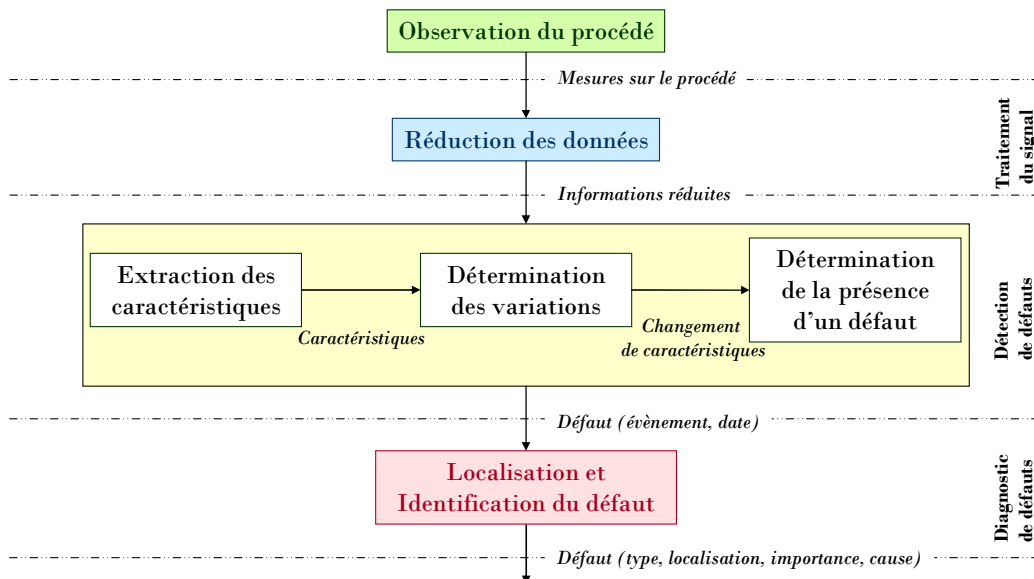


Figure 3.17 La structure de la surveillance

Dans un premier temps, le système de surveillance fait l'acquisition des observations du processus obtenues à partir des capteurs. Ces observations permettent la vérification de la bonne conduite du procédé (vérification de la réaction du système physique face à la commande). Ces données peuvent être tout d'abord prétraitées par des techniques de traitement du signal ou de l'extraction de caractéristiques par exemple. Ici, la surveillance repose sur un modèle du procédé (*modèle de référence*) qui permet de comparer le comportement attendu, obtenu à l'aide du modèle, et le comportement réel, observé. En cas de déviation (*détection*), le diagnostic détermine l'état de fonctionnement dans lequel le système se trouve. Il établit un lien de cause à effet compatible avec les informations disponibles sur le comportement réel du système et sur le comportement du modèle de référence (*diagnostic*) [Zaytoon, 2001].

3.6.1.2 Les tests de détection

Un test de détection vérifie la cohérence de l'ensemble des informations (observations qualitatives et mesures) du système physique avec la connaissance du comportement de référence fourni par le modèle. La nature du test de détection dépend fondamentalement du choix de la formalisation [Touaf, 2005]. Ainsi le tableau suivant présente un panel du couple Formalisation /Technique de détection :

| Formalisation (nature du modèle) | Domaine d'étude |
|---|---|
| Spectres fréquentiels des signaux | Analyse spectrale des signaux |
| Caractéristiques temporelles des signaux | Analyse temporelle des signaux |
| Classe d'appartenance | Méthodes de classification |
| Modèles algèbro-déterministes | Relations de parité |
| Modèles algèbro-stochastiques | Filtre de Kalman |
| Modèles algèbro-ensemblistes | Relations de parité ensemblistes |

Tableau 3.4 Les couples formalisme/technique de détection

Ces tests reposent sur des modèles de comportement ou phénoménologique. Le modèle peut être orienté « bon fonctionnement » (fonctionnement normal) ou « mauvais fonctionnement » (représentation du système physique étudié affecté de défauts). Généralement, l'efficacité des méthodes orientées « bon fonctionnement » est proportionnelle au nombre d'observations du système, tandis que l'efficacité des méthodes orientées « mauvais fonctionnement » est proportionnelle au nombre de modèles de fonctionnement défaillant.

Dans le cas d'un modèle orienté *bon fonctionnement*, on parle de la cohérence du test pour un test de détection négatif : le comportement réel correspond au comportement simulé par le modèle de référence. Inversement, pour un test de détection positif, on parlera de l'incohérence du test de détection. Nous considérerons par la suite un modèle orienté « bon fonctionnement ».

Les tests de détection génèrent des résidus représentatifs de la cohérence du test. Les résidus $r(t)$ doivent être sensibles aux défauts et vérifient le lemme suivant :

Lemme 3-1 :

$\forall t, r(t) \neq 0$, si et seulement si, "il y a au moins un défaut".

En d'autres termes, le test de détection positif implique qu'au moins un composant est dans un état défaillant. Par contre, la contraposée du lemme 3-1 n'est pas évidente : un test de détection négatif n'implique pas forcément l'absence d'un défaut. En effet, il est possible qu'en certains points de fonctionnement un comportement défaillant s'apparente à un comportement normal. Il n'est donc pas possible d'affirmer l'absence de défauts suite à la négation du test de détection. La prise en compte des erreurs de modèle et de mesures permet alors de lever l'ambiguïté en rajoutant une incertitude sur le test de détection.

Par ailleurs, notons que le test de nullité est trop absolu et inadapté à un système soumis à des perturbations. Nous adopterons donc le lemme suivant :

Lemme 3-2 :

$\forall t, |r(t)| > \varepsilon$, si et seulement si, “il y a au moins un défaut”.

[Patton *et al.*, 2000] définissent ε comme une grandeur prédéfinie fonction du taux de détection désiré.

3.6.1.3 Le diagnostic

Le diagnostic détermine les composants défectueux d'un système physique. Durant cette étape, les résidus sont analysés pour décider :

- s'il y a un défaut ou non ;
- de la localisation du défaut (détermination du composant défectueux)
- et de son identification (détermination de la nature du défaut et de sa cause).

Le choix de la méthode de génération de résidus dépend donc :

- du procédé surveillé,
- du modèle disponible et de ses caractéristiques,
- mais également, du type de défauts que l'on souhaite détecter : défauts additifs ou multiplicatifs, multiples ou non...

Nous conseillons aux lecteurs les articles de synthèse d'Isermann pour de plus amples informations sur les avantages et les conditions d'application de ces méthodes [Isermann, 1984, 1993, 1994, 1997]. Nous avons vu précédemment, que grâce à la redondance analytique et à l'évaluation des résidus générés par cette dernière, la détection de défauts est faisable. L'isolation de défauts consiste à déterminer exactement le défaut présent. Afin d'accomplir correctement cette tâche, les résidus doivent réagir de façon différente et préétablie aux différents défauts. Les résidus générés par les différentes méthodes de génération que nous avons exposées dans le point précédent se classent communément en deux groupes [Patton, 1994] : les résidus directionnels [Gertler, 1992] et les résidus structurés [Gertler 1992 ; Chen *et al.*, 1995].

3.6.1.3.1 Les résidus directionnels

L'idée de cette approche est de générer un résidu sous forme d'un vecteur. Ce vecteur résidu est de norme nulle dans le cas du bon fonctionnement du système. En cas de défaillances, ce vecteur se dirige dans une direction, qui est fonction du défaut présent. En effet, la présence d'un défaut amène le vecteur résidu dans une direction fixe (cf. figure 3.18). L'étape d'isolation des défauts consiste alors à déterminer, parmi les différentes directions établies précédemment, laquelle est la plus proche de celle du résidu observé. La conception du générateur de résidus consiste alors à répertorier des directions de défauts, les plus distinctes possibles, afin d'obtenir de bonnes performances d'isolation par la suite.

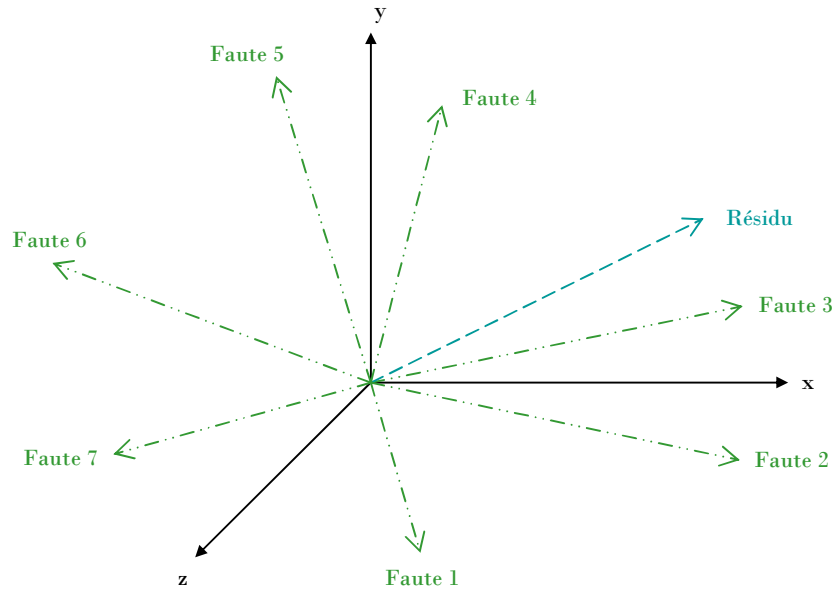


Figure 3.18 Les résidus directionnels

3.6.1.3.2 Les résidus structurés

Cette approche a été développée par l'équipe de Gertler [Gertler et Singer, 1990 ; Gertler, 1991]. Cette méthode est appréciée et amplement utilisée, de par la simplicité de son principe et de sa généralité (elle s'applique à un large panel de systèmes linéaires ou non,...). Elle se base sur un ensemble de résidus : chaque résidu est sensible à un sous-ensemble différent de défauts. La structure obtenue se met sous forme de matrice. On la nomme matrice d'incidence ou structure d'influence. Elle est donc utilisée pour décrire l'influence de chaque défaut sur les résidus.

Ainsi après la détection d'un défaut, l'isolation est effectuée en identifiant la structure d'influence associée, c'est-à-dire celle se rapprochant le plus des valeurs de résidus obtenus. La condition nécessaire pour que tous les défauts inclus dans une structure d'influence soient isolables est que les colonnes de cette structure soient parfaitement distinctes. La structure d'influence satisfaisant cette condition est appelée structure isolante.

C'est cette approche d'isolation basée sur les résidus structurés qui a été retenue pour notre méthodologie de diagnostic. Nous la définissons plus en détail dans le chapitre suivant.

3.7 LE BILAN

L'objectif de ce chapitre est de faire une étude bibliographique générale des différentes techniques habituellement utilisées pour la résolution des problèmes de surveillance. Dans la première partie, la terminologie utilisée dans ce mémoire a été définie. La surveillance est composée de deux fonctions : la détection et le diagnostic. La détection de fautes a pour objectif de déterminer la présence ou non d'une défaillance. En cas de détection, la fonction de diagnostic prend le relais. Elle consiste à localiser le composant défaillant et à identifier les causes ayant provoqué cette défaillance. Les résultats de la détection sont liés à la qualité de l'observation du système.

La classification des méthodes de diagnostic est faite selon la présence ou non d'un modèle. Les méthodes à base de connaissances, basées sur le traitement de données, à base de modèle quantitatif, qualitatif et flou ont été présentées. Le choix d'une de ses approches dépend essentiellement de la connaissance disponible sur le système physique. Les avantages et inconvénients de chacune des approches ont été répertoriés.

Dans le cadre de cette problématique, notre travail porte sur les méthodes à base de modèles, de façon à exploiter les performances de la plate-forme de simulation *PrODHyS*.

En général, seules les variables d'entrée et de sortie sont connues. Il est alors nécessaire de reconstruire l'état du système à partir de ses informations afin de mettre en œuvre les tests de détection. Ce point est traité dans le prochain chapitre.

CHAPITRE 4 LA GÉNÉRATION DE RÉSIDUS À L'AIDE DE RECONSTRUCTEURS D'ÉTAT

Ce chapitre est consacré aux méthodes de génération de résidus à base de reconSTRUCTEURS d'état. Dans un premier temps, quelques notions relatives à la théorie de l'observabilité sont introduites, en particulier les notions d'observateur et d'observabilité. Le but ici est de présenter la problématique de l'observabilité et les principaux travaux dédiés à ce domaine. Ensuite, les difficultés que peut soulever l'observabilité des systèmes dynamiques hybrides sont soulignées. Un aperçu du problème de construction d'observateur pour les systèmes hybrides en découle. Par ailleurs, le filtre de Kalman et ses algorithmes dérivés sont détaillés. Le point suivant traite de l'utilisation de ces observateurs pour des études de détection et de diagnostic de fautes. Les notions de table d'incidence et de signatures de défaut sont introduites.

De nombreux travaux sur les Systèmes Dynamiques Hybrides s'articulent autour des axes modélisation, stabilité et commandabilité [Birouche, 2006]. Ces dernières années, on trouve, dans la littérature, des efforts plus particuliers sur l'observabilité. Bien que la théorie de l'observation d'état a atteint un certain degré de maturité dans les domaines des systèmes continus et à événements discrets, le champ exploratoire reste vaste concernant l'observation des systèmes dynamiques hybrides.

L'observation d'état est particulièrement adaptée aux études de détection et de diagnostic de fautes. En effet, les décisions s'appuient alors sur un plus grand nombre d'informations. Ainsi, 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, puis à utiliser l'erreur d'estimation comme résidu. Clark a été l'un des premiers à utiliser ce concept [Clark *et al.*, 1975 ; Clark, 1979, 1989]. Cette méthode a ensuite été très largement exploitée et notamment a donné lieu à la conception de générateurs de résidus flexibles [Frank, 1987 ; Patton *et al.*, 1989 ; Patton et Chen, 1991a ; Magni et Mouyon, 1991]. Si le problème de conception d'observateurs pour les systèmes linéaires semble bien maîtrisé, il n'en est rien pour les systèmes non linéaires : il n'existe actuellement aucune solution globale complètement satisfaisante.

Ce chapitre aborde le problème de la génération de résidus à partir de l'estimation d'état dynamique des systèmes observés par des mesures en temps discret. Dans un premier temps, les concepts d'observabilité et d'observateur sont définis et présentés dans le cas linéaire et non linéaire. Le filtre de Kalman et ses méthodes dérivées sont traités dans le point suivant. Enfin, l'intérêt des observateurs d'état pour des études de détection et diagnostic de fautes est souligné.

4.1 LES NOTIONS D'OBSERVABILITÉ ET D'OBSERVATEUR

La conduite d'un procédé dynamique (supervision et commande) requiert généralement la connaissance de variables non mesurées. C'est pourquoi, la détection d'un algorithme pour estimer ces variables inconnues à partir des variables mesurées est un atout majeur. Dans une représentation sous forme d'état, on suppose que les variables inconnues sont l'intégralité ou une partie des états du système considéré, et les variables mesurées représentent les entrées et les sorties du système.

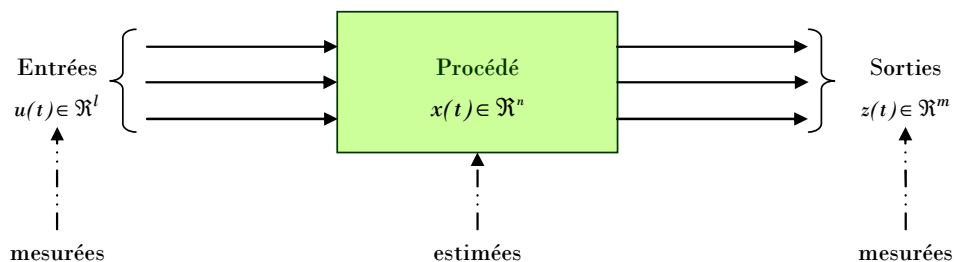


Figure 4.1 Les notations du système

Par convention, on notera $x(t)$ le vecteur d'état du système (cf. figure 4.1). De même, les notations $u(t)$ et $z(t)$ sont utilisées pour représenter respectivement le vecteur des entrées et le vecteur des

sorties du système. En général, le nombre de sorties mesurées $z(t)$ est inférieur au nombre des états du système ($m < n$).

4.1.1 L'observabilité

La notion d'observabilité des systèmes linéaires a été introduite pour la première fois par Kalman [Kalman, 1963]. Considérons donc le système physique suivant

$$\begin{cases} \dot{x}(t) = F \cdot x(t) + G \cdot u(t) \\ z(t) = H \cdot x(t) \end{cases} \quad (4 - 1)$$

- Où $x(t) \in \mathfrak{R}^n$ est le vecteur d'état ;
 $F \in \mathfrak{R}^{n \times n}$ définit le modèle du système,
 $u(t) \in \mathfrak{R}^l$ est le vecteur d'entrée,
 $G \in \mathfrak{R}^{n \times l}$ définit les entrées du système ;
 $z(t) \in \mathfrak{R}^m$ est le vecteur de sortie,
 $H \in \mathfrak{R}^{m \times n}$ définit la sortie.

L'observabilité se traduit comme la capacité à évaluer le vecteur d'état $x(t)$ à partir des mesures $z(t)$ effectuées sur le système physique.

Lemme 4-1 :

Le système (4 - 1) est observable à la date t s'il est possible d'évaluer l'état $x(t)$ du système à partir des connaissances des sorties mesurées $z(t)$ et des entrées $u(t)$.

Le critère d'observabilité de Kalman \emptyset est donné par la formule suivante :

$$\emptyset = \begin{bmatrix} H \\ H \cdot F \\ H \cdot F^2 \\ \vdots \\ H \cdot F^{n-1} \end{bmatrix} \quad (4 - 2)$$

Lemme 4-2 :

Le rang de la matrice d'observabilité est égal à la dimension du sous-espace S d'états observables, où le sous-espace S est inclus dans l'ensemble X des états du système : $S \subset X$.

De cette définition, on déduit que le système (4 - 1) est entièrement observable si le rang de la matrice d'observabilité est égal à la dimension du vecteur d'état n : $\text{rang}(\emptyset) = n$.

On peut alors définir la notion d'indiscernabilité qu'on appelle aussi parfois indistingabilité :

Définition 4-3 : indiscernabilité ou indistingabilité

Soient deux vecteurs d'état x_1 et x_2 avec $x_1, x_2 \in X$. x_1 et x_2 sont dits indiscernables sur l'intervalle de temps $[t_i; t_j]$, si $\forall t \in [t_i; t_j]$ et $\forall u(t) \in U$ (U étant l'espace des entrées), les sorties respectives de x_1 et x_2 , soient z_1 et z_2 , sont identiques sur cet intervalle : $z_1(t) = z_2(t), \forall t \in [t_i; t_j]$.

L'ensemble des états indiscernables du vecteur d'état est noté $\mathbb{I}(x)$.

De la définition 4-3 découle la définition d'observabilité :

Définition 4-4 : observabilité

Un état x est observable sur l'intervalle de temps $[t_i; t_j]$, si son ensemble d'états indiscernables ne contient qu'un seul élément, l'état $x : I(x) = \{x\}$ sur l'intervalle $[t_i; t_j]$.

On en déduit la définition de la discernabilité (nommée aussi distingabilité) :

Définition 4-5 : discernabilité ou distingabilité

Soient deux vecteurs d'état x_1 et x_2 avec $x_1, x_2 \in X$ tel que $x_1 \neq x_2$. x_1 et x_2 sont dits discernables sur l'intervalle de temps $[t_i; t_j]$, si $\forall t \in [t_i; t_j] \forall u(t) \in U$, on a $z_1(t) \neq z_2(t)$.

4.1.2 Les observateurs

Un observateur ou un reSTRUCTEUR d'état est un système qui fournit une estimation de l'état courant non mesuré, à partir des entrées et sorties du système. Habituellement, le système dynamique est représenté par un modèle sous forme d'équations différentielles ou de différences d'état et d'une équation algébrique d'observation. Considérons donc le système dynamique suivant :

$$\begin{cases} \dot{x} = f(x, u) \\ z = h(x) \end{cases} \quad (4 - 3)$$

On notera \hat{x} l'estimation de l'état x . On cherche à concevoir un estimateur de l'état $x(t)$ du système exploitant la connaissance des entrées $u(t)$ et des sorties $z(t)$. Plus précisément, on désire écrire l'observateur θ sous la forme d'un système dynamique :

$$\begin{cases} \dot{\hat{x}} = f(\hat{x}, u) - \theta(g) \cdot (\hat{x} - z) \\ \dot{g} = \phi(\hat{x}, u, z, g) \end{cases} \quad (4 - 4)$$

g est une fonction dépendant des caractéristiques du système. \dot{g} permet de faire évoluer (corriger) l'observateur θ au cours du temps. $\theta(g)$ est le gain d'observation et $\theta(g) \cdot (\hat{x} - z)$ correspond au terme de correction. θ est défini tel que l'erreur d'estimation $\tilde{x}(t) = \hat{x}(t) - x(t)$ converge asymptotiquement (parfois exponentiellement) vers zéro quand t tend vers l'infini. Lorsque cette propriété est satisfaite, on parle d'observateur asymptotique.

4.1.2.1 Les observateurs des systèmes linéaires

Nous n'exposerons ici que les concepts clés qui caractérisent les observateurs des systèmes linéaires. Nous ne détaillerons donc pas leur implémentation. Pour plus d'informations à ce sujet, nous recommandons aux lecteurs les travaux suivants : [Kalman, 1960 ; Kalman et Bucy, 1961 ; Luenberger, 1971]

Comme nous l'avons vu dans les définitions 4-3, 4-4 et 4-5, l'observabilité des systèmes linéaires est indépendante des entrées $u(t)$. En effet, si un tel système est observable pour une entrée nulle ($u(t) = 0, \forall t \geq 0$), alors il est observable pour toute entrée. Pour ces systèmes, des algorithmes ont été précisément développés. Ces derniers portent le nom d'observateurs de type Kalman-Luenberger. Ce type d'estimateurs s'applique aux systèmes linéaires observables entièrement, ou observables partiellement mais

tels que tous les pôles de la partie inobservable sont stables [Xu, 2002]. Le système est alors défini comme étant détectable. On distingue alors le cas déterministe du cas stochastique :

- Dans le cas déterministe, de nombreux gains ont été développés. Le choix s'effectue selon la vitesse de convergence souhaitée. Citons l'algorithme le plus connu pour les systèmes linéaires stationnaires, l'observateur de Luenberger [Luenberger, 1971 ; Landau, 2001]. Cet observateur a été défini dans le point 3.4.1.
- Dans le cas stochastique, on suppose que la dynamique du système et les sorties du système (mesures) sont affectées par des bruits (centrés et statistiquement connus). Dans ce contexte, il existe un gain optimal unique. Ce gain minimise la variance de l'erreur d'estimation. Cet observateur est le filtre de Kalman [Landau, 2001]. Il sera développé dans le point suivant.

4.1.2.2 Les observateurs des systèmes non – linéaires

Les systèmes considérés dans ces travaux (les procédés) ont en général un comportement non-linéaire. Pour ce type de système, il n'existe pas de solution générale pour estimer leur état.

Ainsi, les premières solutions proposées consistaient à se ramener à un cas linéaire (en utilisant par exemple un modèle linéarisé autour d'un point de fonctionnement) afin de pouvoir utiliser les estimateurs de type Kalman-Luenberger [Marino et Tomei, 1995 ; Krener, 1984 ; Bastin et Gevers, 1998 ; De Persis et Isidori, 2001]. Néanmoins, ce type de méthode soulève des problèmes de généralité. En effet, considérons le cas où le générateur de résidus se base sur un modèle linéarisé autour d'un point de fonctionnement. Lorsque l'état du système s'écarte sensiblement de ce point de fonctionnement, des dérives importantes peuvent être observées, à cause du caractère non linéaire du comportement [Maquin *et al.*, 1997]. Le principal inconvénient de ces méthodes est donc qu'elles ne s'appliquent que sous des conditions très restrictives [Xu, 2002]. Par conséquent, ces méthodes ne s'utilisent pas pour la plupart des systèmes non-linéaires.

Il apparaît alors naturel de considérer des modèles non-linéaires, représentant au mieux le comportement [Hengy et Frank, 1986 ; Frank, 1987 ; Adjallah, 1993]. Citons quelques observateurs connus :

- En pratique, le filtre de Kalman étendu et ses variantes sont largement exploités [Jazwinski, 1970 ; Reif et Ubehauen, 1999 ; Einicke et White, 1999]. Ces algorithmes sont le sujet du point suivant.
- Gauthier *et al.* ont défini un observateur à grand gain, algorithme dédié aux systèmes non-linéaires observables pour toute entrée, dont la convergence est prouvée théoriquement [Gauthier *et al.*, 1992].
- Citons également l'observateur adaptatif non linéaire, utilisé quand l'état et les paramètres du système sont inconnus. Cet algorithme estime conjointement l'état et les paramètres du système [Xu, 2002 ; Xu et Zhang, 2004].
- Il existe aussi des observateurs implicites sous forme d'équations algèbro-différentielles [Nikoukchah, 1998].

4.1.3 La synthèse des observateurs de systèmes dynamiques hybrides

Nous avons vu que pour les systèmes linéaires, la conception d'un observateur est assurée par la propriété d'observabilité. En revanche, pour les systèmes non-linéaires, l'observabilité n'est pas suffisante à la réalisation d'un estimateur d'état général. Ainsi, en général, la conception d'observateurs pour les systèmes non-linéaires est effectuée pour des classes spécifiques.

Par ailleurs, le couplage d'une structure continue à une structure discrète induit un caractère mixte aux systèmes dynamiques hybrides. Un observateur de système dynamique hybride peut alors être vu comme l'association d'un observateur de la partie continue et un observateur de la partie discrète. [Balluchi *et al.*, 2002] propose un observateur de ce type. L'algorithme proposé s'appuie :

- sur la théorie des systèmes à événements discrets pour l'observateur « discret », afin d'identifier l'état discret ;
- et sur le formalisme classique des observateurs pour l'observateur « continu », pour estimer l'état continu.

Néanmoins, la majorité des travaux trouvés dans la littérature simplifient le problème d'observation en ne tenant pas compte de la partie discrète. Pour cela, ils s'appuient sur l'hypothèse qu'à chaque instant, le sous-système actif est connu [Birouche, 2006].

4.2 LE FILTRE DE KALMAN

Le filtrage consiste à mettre en forme un signal. Généralement, la fonction de filtrage considère les informations utiles (signal) qui sont polluées par un bruit. Le filtre de Kalman est un des outils les plus utilisés de l'estimation d'état et de paramètres pour les systèmes stochastiques [Kalman, 1960 ; Kalman et Bucy 1961]. Il estime de façon optimale l'état du système linéaire. C'est un estimateur itératif basé sur une relation de récurrence, ce qui signifie que seul l'état précédemment estimé et les mesures actuelles sont nécessaires pour le calcul de l'estimation de l'état courant.

Pendant longtemps, son rôle était d'éliminer les bruits résiduels. Ses applications aujourd'hui sont diverses :

- estimation des conditions initiales inconnues ;
- prédiction des trajectoires de mobiles ;
- localisation d'un engin (navigation, radar...) ;
- utilisation pour implanter des lois de commande fondées sur un estimateur de l'état et un retour d'état (commande linéaire quadratique gaussienne)...

Le problème de l'estimation d'état à la date t d'un système dynamique soumis à des entrées déterministes et aléatoires, à partir des mesures bruitées du système obtenues à la date t_m , est un problème :

- de prédiction, si $t > t_m$;
- de filtrage, si $t = t_m$;
- et de lissage, si $t < t_m$.

4.2.1 Le principe

Parmi les méthodes dites à base de modèles utilisées dans le passé, le filtre de Kalman est clairement l'un des outils les plus utilisés et l'un des plus populaires de l'estimation d'état et de paramètres pour les systèmes stochastiques. Le modèle du système n'est pas nécessairement linéaire mais il doit être différentiable.

Considérons tout d'abord le cas linéaire. Le système d'équation utilisé dans le filtre de Kalman se décompose en la définition de deux modèles : l'un représentant le processus et le second modélisant les mesures. Il représente l'état du système dynamique. Considérons donc le système suivant :

$$\begin{cases} \dot{x}(t) = F(t) \cdot x(t) + G(t) \cdot u(t) + v(t) \\ z(t) = H(t) \cdot x(t) + w(t) \end{cases} \quad (4 - 5)$$

Le filtre de Kalman requiert un système à valeur continue. Une forme discrète simplifiée du modèle de l'espace d'état peut être, par exemple, générée en utilisant l'approximation d'Euler. En appliquant cette approximation au système (4 - 5), on obtient le système suivant :

$$\begin{cases} X_{k+1} = F \cdot X_k + G \cdot U_k + V_k \\ Z_{k+1} = H \cdot X_{k+1} + W_{k+1} \end{cases} \quad (4 - 6)$$

Où : $X_k \in \mathfrak{R}^n$ est le vecteur d'état du système,

$F \in \mathfrak{R}^{n \times n}$ définit la dynamique du système,

$U_k \in \mathfrak{R}^l$ est le vecteur d'entrée (ou vecteur de commande),

$G \in \mathfrak{R}^{n \times l}$ définit les entrées du système ; cette matrice de commande représente le lien entre la distribution de l'entrée dans le vecteur d'état ;

$V_k \in \mathfrak{R}^n$ est le vecteur représentant les erreurs du modèle (bruit d'état),

$Z_k \in \mathfrak{R}^m$ est le vecteur d'observations (mesures),

$H \in \mathfrak{R}^{m \times n}$ définit les mesures, c'est la matrice d'observation ; elle fait le lien entre le vecteur d'état et les mesures ;

et $W_k \in \mathfrak{R}^m$ est le vecteur qui représente le bruit des mesures,

Lors de l'utilisation de ce filtre, l'hypothèse posée consiste à dire que les bruits $\{V_k\}$ et $\{W_k\}$ sont des bruits blancs gaussiens non corrélés de moyenne nulle et de matrices de covariances, respectivement Q_k et R_k :

$$\begin{cases} E[V_k V_k^T] = Q_k \\ E[W_k W_k^T] = R_k \end{cases} \quad (4 - 7)$$

Les matrices Q_k et R_k sont des paramètres de conception qui s'interprètent respectivement comme les variances des bruits de dynamique et d'observation. Généralement la covariance du bruit du système est plus difficile à évaluer que celle de la mesure. En effet, aucune observation directe de l'état du système n'est possible. Le réglage du filtre sera explicité ultérieurement (cf. Chapitre 7).

La figure 4.2 illustre les différentes étapes de l'estimation de l'état du système. On distingue deux grandes étapes : la prédiction qui correspond à l'estimation a priori et la correction qui correspond à l'estimation a posteriori. Dans un premier temps, le vecteur d'état est estimé à partir des valeurs de l'étape précédente. Ensuite, il est corrigé grâce aux observations dans l'étape de correction. Ces deux étapes forment une boucle itérative :

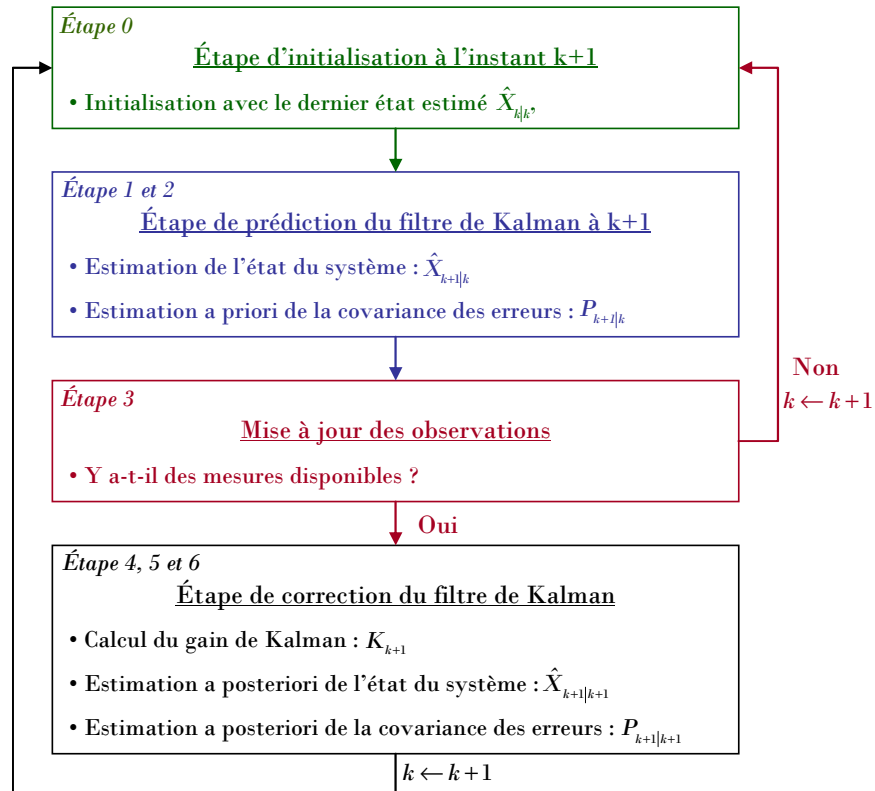


Figure 4.2 L'organigramme de principe du filtre de Kalman

La notation $k+1|k$ signifie que le calcul de la prédiction est fait à l'instant $k+1$ connaissant seulement l'état à l'instant k . Donc la notation $k+1|k+1$ correspond à la correction de l'estimation $k+1|k$ à partir des nouvelles informations de l'instant $k+1$ (mesures).

Les équations du filtre de Kalman sont algébriques et récurrentes. Par conséquent, le calcul d'une estimation nécessite peu de temps CPU, ce qui permet d'exploiter ce filtre pour du contrôle en temps réel. Dans l'algorithme, chaque nouvelle estimation est une combinaison entre la précédente estimation et les mesures actuelles. Les étapes suivantes correspondent à une itération à l'instant $k+1$ de l'algorithme du filtre de Kalman :

Étape 0 : Initialisation avec le précédent état estimé $\hat{X}_{k|k} : \hat{X}_{k|k} \leftarrow \hat{X}_{k+1|k+1}$

Étape 1 : Prédiction de l'état du système autour de $\hat{X}_{k+1|k}$

$$\hat{X}_{k+1|k} = F \cdot \hat{X}_{k|k} + G \cdot U_k \tag{4 - 8}$$

Étape 2 : Calcul de la matrice de covariance de l'erreur autour de $P_{k+1|k}$:

$$P_{k+1|k} = F \cdot P_{k|k} \cdot F^T + Q_k \quad (4 - 9)$$

Étape 3 : Si de nouvelles mesures sont disponibles, alors allez à l'étape 4, sinon à l'étape 0

Étape 4 : Calcul du gain de Kalman K_{k+1} par l'expression :

$$K_{k+1} = P_{k+1|k} \cdot H^T \cdot [H \cdot P_{k+1|k} \cdot H^T + R_{k+1}]^{-1} \quad (4 - 10)$$

Étape 5 : Mise à jour de l'état estimé $\hat{X}_{k+1|k+1}$:

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} \cdot [Z_{k+1} - H \cdot \hat{X}_{k+1|k}] \quad (4 - 11)$$

Étape 6 : Correction de la matrice de covariance de l'erreur :

$$P_{k+1|k+1} = [I - K_{k+1} \cdot H] \cdot P_{k+1|k} \quad (4 - 12)$$

Le gain de Kalman a été calculé de façon à minimiser la trace de la covariance d'erreur a posteriori $P_{k+1|k+1}$. Si les observations sont plus précises que l'erreur du modèle, alors $R_{k+1} \ll P_{k+1|k+1}$ et le gain de Kalman ne dépend que de la matrice caractéristique des mesures : $K_{k+1} = k(H)$. Ainsi, les mesures ont plus de poids dans l'estimation d'état. Dans le cas contraire (l'erreur de modèle est plus précise que celle des mesures), on a $R_{k+1} \gg P_{k+1|k+1}$ et $K_{k+1} = \frac{[P_{k+1|k+1} \cdot H^T]}{R_{k+1}}$. Ainsi la correction sera faible.

4.2.2 La linéarisation des problèmes non linéaires

La formulation présentée dans le point précédent se base sur un modèle linéaire du système physique. Elle n'est donc pas applicable à des modèles non linéaires. Pour le filtrage non-linéaire, il n'y a pas de solution générale. En pratique, les ingénieurs continuent à utiliser des versions dérivées du filtre de Kalman-Bucy. Le problème fondamental dans ce cas là est de linéariser le modèle, afin d'appliquer le filtre de Kalman standard pour estimer l'état du système. Plusieurs méthodes ont été développées dans ce but. Nous présentons ici les deux méthodes principales : le filtre de Kalman linéarisé et le filtre de Kalman étendu.

Pour cela, considérons les équations suivantes du système non linéaire :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) + v(t) \\ z(t) = h(x(t)) + w(t) \end{cases} \quad (4 - 13)$$

Où, $x(t)$ est le vecteur d'état,

f définit le modèle du système,

$u(t)$ est le vecteur d'entrée,

$v(t)$ traduit l'erreur du modèle,

$z(t)$ est le vecteur de sortie,

h définit la sortie,

et $w(t)$ traduit le bruit des mesures.

4.2.2.1 Le filtre de Kalman linéarisé

Le filtre de Kalman linéarisé est basé sur la linéarisation du modèle (4 - 13) autour d'une certaine trajectoire nominale. Cette méthode ne dépend que des mesures. Elle requiert un système à valeur continue sur temps discret. Le système (4 - 13) devient :

$$\begin{cases} x_{k+1} = f(x_k, u_k) + v_k \\ z_{k+1} = h(x_{k+1}) + w_{k+1} \end{cases} \quad (4 - 14)$$

f et g sont des fonctions non-linéaires connues. La non-linéarité peut apparaître soit dans le modèle du système soit dans celui des mesures soit bien sûr dans les deux.

Supposons que nous avons déterminé par un moyen quelconque (par exemple par l'expansion en série de Taylor) la trajectoire approximative $\{\bar{x}_k\}$, solution approchée du problème. Cette trajectoire est appelée trajectoire nominale, et représente la trajectoire de référence (cf. figure 4.3).

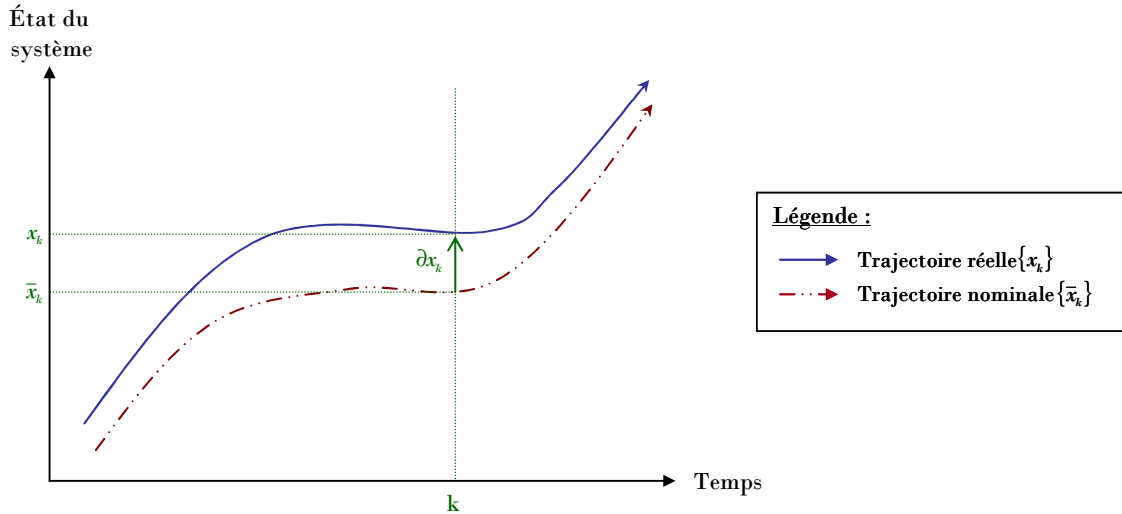


Figure 4.3 La trajectoire réelle et nominale du filtre de Kalman linéarisé

La trajectoire réelle peut donc s'écrire de la façon suivante :

$$\begin{cases} x_{k+1} = \bar{x}_{k+1} + \partial x_{k+1} = f(\bar{x}_k + \partial x_k, u_k) + v_k & \text{car } x_k = \bar{x}_k + \partial x_k \\ z_{k+1} = h(\bar{x}_{k+1} + \partial x_{k+1}) + w_{k+1} \end{cases} \quad (4 - 15)$$

La linéarisation consiste alors à :

- linéariser f autour de \bar{x}_k :

$$f(x, u) \approx f(\bar{x}_k, u_k) + \nabla f(\bar{x}_k, u_k) \cdot (x - \bar{x}_k) \quad (4 - 16)$$

- linéariser h autour de \bar{x}_{k+1} :

$$h(x) \approx h(\bar{x}_{k+1}) + \nabla h(\bar{x}_{k+1}) \cdot (x - \bar{x}_{k+1}) \quad (4 - 17)$$

Le système linéarisé obtenu s'écrit sous la forme :

$$\begin{cases} \mathbf{X}_{k+1} = \mathbf{F}_k \cdot (\mathbf{X}_k - \bar{x}_k) + f_k + V_k \\ \mathbf{Z}_{k+1} = \mathbf{H}_{k+1} \cdot (\mathbf{X}_{k+1} - \bar{x}_{k+1}) + h_{k+1} + W_{k+1} \end{cases} \quad (4 - 18)$$

$$\text{Avec } F_k = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}_k} ; f_k = f(\bar{x}_k, u_k) ; H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\bar{x}_{k+1}} \text{ et } h_{k+1} = h(\bar{x}_{k+1}).$$

On applique le filtre de Kalman-Bulcy à ce nouveau système, et on obtient exactement :

- les équations de prédiction du filtre de Kalman suivantes :

$$\begin{cases} \hat{\mathbf{X}}_{k+1|k} = f_k + F_k \cdot (\hat{\mathbf{X}}_{k|k} - \bar{x}_k) \\ \mathbf{P}_{k+1|k} = F_k \cdot \mathbf{P}_{k|k} \cdot F_k^T + Q_k \end{cases} \quad (4 - 19)$$

$$\text{Avec } F_k = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}_k} \text{ et } f_k = f(\bar{x}_k, u_k)$$

- et les équations de correction :

$$\begin{cases} \mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T \cdot [\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T + R_{k+1}]^{-1} \\ \hat{\mathbf{X}}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} + \mathbf{K}_{k+1} \cdot [\mathbf{Z}_{k+1} - (h_{k+1} + \mathbf{H}_{k+1} \cdot (\hat{\mathbf{X}}_{k+1|k} - \bar{x}_{k+1}))] \\ \mathbf{P}_{k+1|k+1} = [\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}] \cdot \mathbf{P}_{k+1|k} \end{cases} \quad (4 - 20)$$

$$\text{Avec } H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\bar{x}_{k+1}} \text{ et } h_{k+1} = h(\bar{x}_{k+1}).$$

Par ailleurs, les approximations suivantes sont effectuées :

$$\begin{aligned} f(\hat{\mathbf{X}}_{k|k}, u_k) &= f_k + F_k \cdot (\hat{\mathbf{X}}_{k|k} - \bar{x}_k) \\ h(\hat{\mathbf{X}}_{k+1|k}) &= h_{k+1} + H_{k+1} \cdot (\hat{\mathbf{X}}_{k+1|k} - \bar{x}_{k+1}) \end{aligned} \quad (4 - 21)$$

On obtient finalement les équations suivantes pour le filtre de Kalman linéarisé :

$$\begin{cases} \hat{\mathbf{X}}_{k+1|k} = f(\hat{\mathbf{X}}_{k|k}) \\ \mathbf{P}_{k+1|k} = F_k \cdot \mathbf{P}_{k|k} \cdot F_k^T + Q_k \\ \mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T \cdot [\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1|k} \cdot \mathbf{H}_{k+1}^T + R_{k+1}]^{-1} \\ \hat{\mathbf{X}}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} + \mathbf{K}_{k+1} \cdot [\mathbf{Z}_{k+1} - h(\hat{\mathbf{X}}_{k+1|k})] \\ \mathbf{P}_{k+1|k+1} = [\mathbf{I} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}] \cdot \mathbf{P}_{k+1|k} \end{cases} \quad (4 - 22)$$

$$\text{Avec } F_k = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}_k} \text{ et } H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\bar{x}_{k+1}}.$$

Cette méthode présente l'avantage d'être facilement exécutable en temps réel. Néanmoins, un de ses inconvénients majeurs est son inadéquation pour des processus à durée prolongée. En effet, la trajectoire de référence peut diverger en un point où l'hypothèse linéaire n'est plus valide par rapport à la variation du vecteur d'état [Abuhadrous,2005]. Une solution à ce problème est de remplacer la trajectoire nominale par la trajectoire estimée. Cette nouvelle approche constitue le filtre de Kalman étendu. Elle

présente l'avantage d'être plus robuste aux approximations non linéaires des erreurs que le filtre de Kalman linéarisé.

4.2.2.2 Le filtre de Kalman étendu

Ici, la trajectoire nominale est donc remplacée par l'estimateur courant de X_{k+1} compte-tenu des observations disponibles à l'instant k . Le filtre de Kalman étendu a été implémenté pour la première fois par Schmidt [Schmidt, 1966 ; Mohinder *et al.*, 2001]. Cette approche est probablement l'approche la plus commune et la plus populaire pour traiter les systèmes non-linéaires. Parmi les méthodes de diagnostic dites à base de modèles utilisées dans le passé, ce filtre est clairement l'un des plus étudiés. À l'origine, il a été exploité tant pour approximer les états et paramètres des procédés chimiques, que pour identifier les causes du comportement anormal du procédé [Watanabe *et* Himmelblau, 1984].

Notons que bien évidemment, le modèle du système n'est pas nécessairement linéaire mais il doit être différentiable.

Ce filtre est similaire au filtre linéarisé. La linéarisation du système, initialement non linéaire, s'effectue, dans ce cas là, autour de la trajectoire estimée par le filtre, et plus autour de la trajectoire nominale pré-calculée comme c'était le cas pour le filtre de Kalman linéarisé. Les dérivées partielles sont évaluées le long de la trajectoire mise à jour par les estimations du filtre ; elles dépendent donc des observations du système.

La méthode consiste à linéariser la fonction f autour de $\hat{X}_{k|k}$:

$$f(x, u) \approx f(\hat{X}_{k|k}, u_k) + \nabla f(\hat{X}_{k|k}, u_k) \cdot (x - \hat{X}_{k|k}) \quad (4 - 23)$$

et à linéariser la fonction h autour de $\hat{X}_{k+1|k}$:

$$h(x) \approx h(\hat{X}_{k+1|k}) + \nabla h(\hat{X}_{k+1|k}) \cdot (x - \hat{X}_{k+1|k}) \quad (4 - 24)$$

Le système (4 - 18) devient :

$$\begin{cases} X_{k+1} = F_k \cdot (X_k - \hat{X}_{k|k}) + f_k + V_k \\ Z_{k+1} = H_{k+1} \cdot (X_{k+1} - \hat{X}_{k+1|k}) + h_{k+1} + W_{k+1} \end{cases} \quad (4 - 25)$$

$$\text{Avec } F_k = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\hat{X}_{k|k}} ; f_k = f(\hat{X}_{k|k}, u_k) ; H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\hat{X}_{k+1|k}} \text{ et } h_{k+1} = h(\hat{X}_{k+1|k}).$$

Remarque : Notons que les gradients ici sont calculés en intégrant l'estimation de l'état à l'instant, et non pas au point réel comme c'est le cas pour le filtre de Kalman linéarisé.

En appliquant le filtre de Kalman-Bucy à ce nouveau système, on obtient exactement le résultat suivant :

$$\begin{cases} \hat{X}_{k+1|k} = f(\hat{X}_{k|k}, u_k) \\ P_{k+1|k} = F_k \cdot P_{k|k} \cdot F_k^T + Q_k \\ K_{k+1} = P_{k+1|k} \cdot H_{k+1}^T \cdot [H_{k+1} \cdot P_{k+1|k} \cdot H_{k+1}^T + R_{k+1}]^{-1} \\ \hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} \cdot [Z_{k+1} - h(\hat{X}_{k+1|k})] \\ P_{k+1|k+1} = [I - K_{k+1} \cdot H_{k+1}] \cdot P_{k+1|k} \end{cases} \quad (4 - 26)$$

Avec $F_k = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\hat{X}_{k|k}}$ et $H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\hat{X}_{k+1|k}}$.

Le filtre de Kalman étendu peut être visualisé dans une configuration de boucle fermée (cf. figure 4.4) :

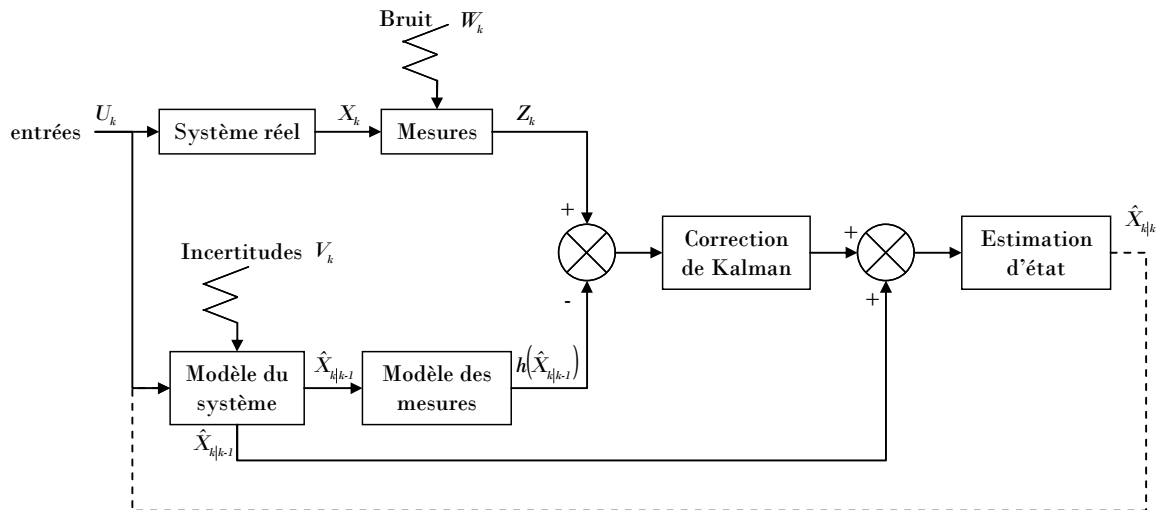


Figure 4.4 Le schéma bloc du filtre de Kalman étendu

Le filtre de Kalman étendu est peu coûteux en temps de calcul et donne de bons résultats pour des systèmes à non-linéarités modérées [Jazwinski, 1970 ; Reif et Unbehauen, 1999 ; Einicke et White 1999]. Il est à noter que dès que les non-linéarités deviennent trop fortes ou s'il est mal initialisé, le filtre de Kalman étendu n'est pas performant.

4.3 L'EXPLOITATION DES RÉSIDUS POUR DES ÉTUDES DE DIAGNOSTIC

4.3.1 Le principe de l'approche

Nous nous intéressons ici à l'approche connue sous le nom de Fault Detection and Isolation (FDI). Cette méthode est composée de trois fonctions de base : la génération de résidus, la détection et le diagnostic (ou localisation). Son principe est illustré sur la figure 4.5. Nous n'aborderons pas, dans ce mémoire, les problématiques soulevées par les étapes de pronostic et de reconfiguration du système (fonction de reprise).

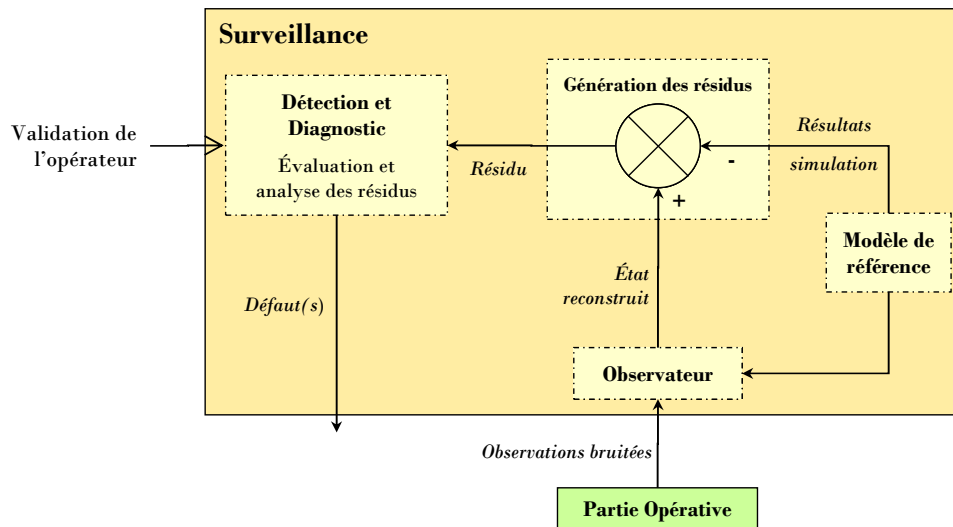


Figure 4.5 Le diagnostic FDI

4.3.2 La génération de résidus

La première étape de l'approche FDI consiste à générer des indicateurs de défauts, appelés également résidus. Ces derniers sont des conteneurs d'informations sur les défaillances du système physique surveillé. Dans le cas des observateurs, le principe est basé sur la mesure de l'écart entre l'état reconstruit du système réel obtenu par l'observateur et la valeur théorique de cet état dans son comportement nominal obtenue par la simulation du modèle de référence.

La génération de résidus est une étape fondamentale pour l'approche FDI. En effet, de la structure du système de résidus engendré dépend la robustesse de la localisation [Ripoll, 1999].

En principe, un résidu représente une relation de redondance analytique. Cette relation est déduite du modèle général du système physique et ne fait intervenir que les variables mesurées, les observations d'entrée et de sortie.

4.3.3 La détection

Deuxième étape du système FDI, la détection consiste en l'évaluation des résidus afin de déterminer la présence ou non d'un défaut dans le système physique.

Nous avons vu précédemment qu'en l'absence de défaut, en d'autres termes en fonctionnement nominal, le résidu doit en théorie être nul. Néanmoins, en raison des incertitudes sur le modèle de référence et des bruits de mesures, le résidu réellement engendré ne peut suivre cette propriété et nous avons donc adopté le lemme 4-2 qui souligne l'existence d'un seuil de détection. La principale difficulté réside donc dans la détermination du seuil optimal des résidus, c'est-à-dire la détermination du seuil ε vérifiant les propriétés suivantes :

- en présence de défaut, $r(t) \geq \varepsilon$
- en fonctionnement normal, $r(t) < \varepsilon$.

4.3.3.1 Un choix difficile

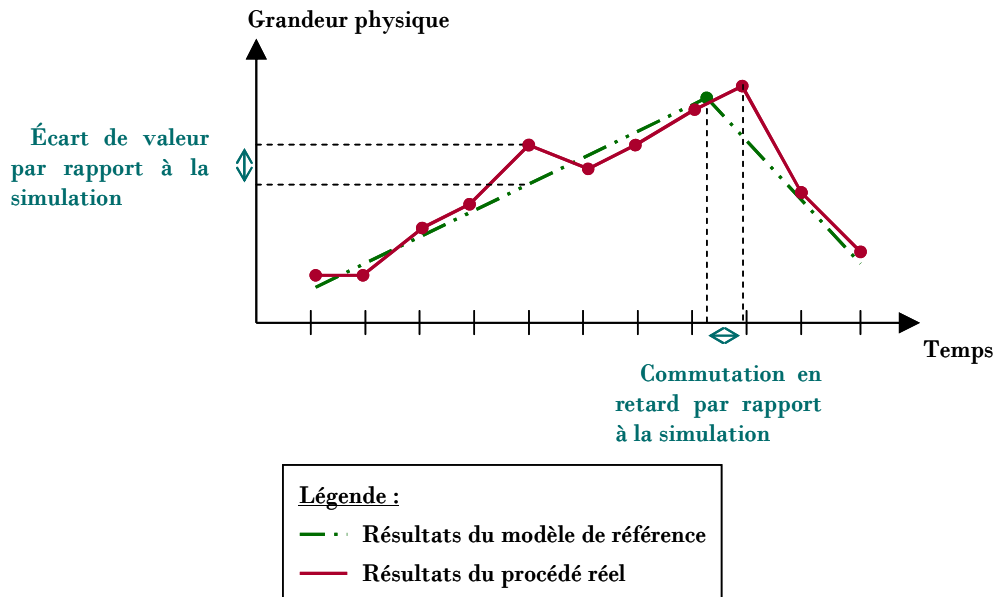


Figure 4.6 Les incertitudes sur la détection

Ce choix est soumis à de nombreuses difficultés :

- Il peut subsister en effet une *incertitude spatiale* (cf. figure 4.6). Cette incertitude réside dans le choix du seuil à partir duquel l'écart entre la valeur réelle et celle obtenue par le modèle de référence est estimé « anormal » :
 - Un seuil trop faible entraîne de fausses alarmes : la situation est supposée anormale alors qu'elle ne l'est pas.
 - Au contraire, un seuil fixé à une valeur trop importante risque d'engendrer une non-détection : la situation est supposée normale alors qu'elle ne l'est pas.

- D'autre part, la simulation du modèle de référence doit évoluer en temps réel et de manière synchrone avec le procédé surveillé. En général, cette caractéristique ne peut pas être garantie à tout instant. En effet, une *incertitude temporelle* (cf. figure 4.6) existe du fait des paramètres intrinsèques des modèles de simulation décrivant la dynamique du système. Ainsi, certains événements temporels ou d'état peuvent apparaître au niveau du modèle de référence simulé soit en avance, soit en retard par rapport aux observations faites sur le procédé surveillé. Ici encore, se pose le problème de déterminer à partir de quel écart ε peut-on considérer qu'une défaillance est potentiellement détectée. Dans tous les cas, une procédure de recalage du modèle de référence sur le procédé réel doit être réalisée pour pouvoir valider un test de détection.

La problématique réside donc dans la détermination du seuil optimal qui est obtenu à partir d'un compromis entre le taux de fausses alarmes et le taux de non-détection. Généralement, elle est établie à partir d'une série de simulations dans lesquelles sa valeur est ajustée.

4.3.3.2 Les seuils fixes et adaptatifs

Deux types de seuils existent : les seuils fixes et adaptatifs.

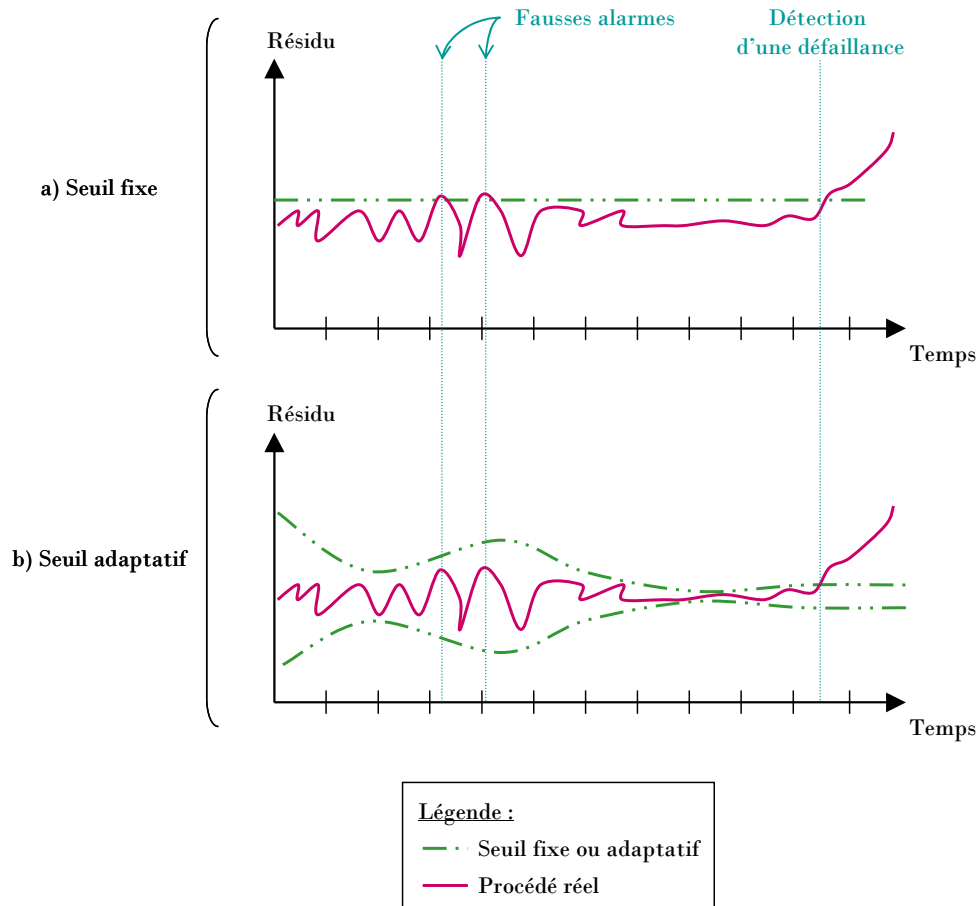


Figure 4.7 Les différents seuils

Les seuils fixes

Cette méthode a été l'objet des premiers travaux sur le problème de choix du seuil. Elle consiste à détecter l'instant pour lequel les grandeurs surveillées sortent d'une zone délimitée par des seuils autour d'une valeur objectif (valeur de la grandeur au sein du modèle de référence). La détection obtenue est indépendante du temps et des entrées du système. Cette technique apparaît inadaptée pour un système soumis à des perturbations car elle peut conduire à de fausses alarmes (cf. figure 4.7.a).

Les seuils adaptatifs

Cette méthode repose sur la définition pour un résidu d'une borne inférieure et d'une borne supérieure. Ces bornes dépendent des incertitudes présentes (incertitudes sur le modèle de référence et bruit des mesures) et sont ainsi plus robustes face à ces dernières. Ils peuvent être différents selon le mode opératoire étudié. La figure 4.7.b montre l'avantage d'utiliser des seuils adaptatifs plutôt que des seuils fixes. On constate que le seuil adaptatif permet d'éviter l'émission de fausses alarmes. De nombreux travaux traitent de cette technique [Weiss, 1988 ; Ding et Frank, 1991 ; Sauter *et al.*, 1996].

4.3.4 Le diagnostic

La localisation est la troisième étape du système FDI et constitue l'étape de diagnostic proprement dit. Le but de cette étape est de trouver l'origine de la défaillance détectée. En général, la localisation s'appuie sur les propriétés structurelles du système physique.

Afin de comprendre le processus de la localisation, nous allons définir, dans un premier temps les notions de table d'incidence et de signature de défaut.

4.3.4.1 La table d'incidence et la signature de défaut

Une table d'incidence, nommée aussi matrice d'incidence, table de signature de pannes ou encore matrice de signature, représente la relation entre les résidus \mathcal{R} et l'ensemble des défauts considérés du système F .

Définition 4-6 : table d'incidence

Soit \mathcal{R} l'ensemble des résidus caractérisant le système, soit F l'ensemble des défauts du système physique,

La table d'incidence \mathcal{T} est définie de la façon suivante :

$$\mathcal{T} : \mathcal{R} \times F \rightarrow [0;1]$$

$$(r; f) \mapsto \mathcal{T}(r, f)$$

$$\text{avec } \mathcal{T}(r, f) = \begin{cases} 1, & \text{si et seulement si } r \text{ est sensible au défaut } f. \\ 0, & \text{si et seulement si } r \text{ est découplé au défaut } f. \end{cases}$$

La table d'incidence est donc une table binaire, dont la ligne i correspond au résidu r_i et la colonne j correspond au défaut f_j . Elle est définie selon l'expérience et dépend du système.

Chaque colonne de cette table représente une signature de défaut particulier, dans des conditions idéales. Une signature de défauts multiples est une combinaison de signatures élémentaires établie par des opérateurs logiques.

4.3.4.2 Le processus

Le processus de diagnostic est un processus en ligne d'appariement (Pattern matching) [Denoeux et Zouhal, 2001]. Le vecteur résidu, obtenu à l'étape précédente de détection est alors comparé aux signatures de défauts, c'est-à-dire aux colonnes de la table d'incidence. Un défaut existe sur le système physique quand le résidu s'apparente à une signature de défaut connue. Ceci définit l'étape de *localisation*.

Afin que chaque défaillance soit localisée, il faut que les colonnes de la matrice d'incidence soient indépendantes deux à deux, et bien sûr chaque colonne doit posséder au moins un « un » (au moins un résidu doit être sensible à ce défaut). Ainsi, la localisation de défaut est unique, si et seulement si, la signature associée à ce défaut est unique.

Selon [Gertler, 1992 ; Ripoll, 1999], la table d'incidence est donc :

- *non localisante*, si au moins deux signatures de pannes sont identiques :

$$\mathcal{T}_1 : \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ r_1 & \left(\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{array} \right) \end{matrix}$$

Figure 4.8 Une table d'incidence non localisante

- *déterministiquement localisante* si toutes les signatures de défauts sont différentes et qu'une signature peut être déduite d'une autre en changeant un « un » par un « zéro » :

$$\mathcal{T}_2 : \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ r_1 & \left(\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{array} \right) \end{matrix}$$

Figure 4.9 Une table d'incidence déterministiquement localisante

- *statistiquement localisante* si toutes les signatures de défauts sont différentes et ne peuvent être déduites d'une autre en modifiant un « un » par un « zéro » :

$$\mathcal{T}_3 : \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ r_1 & \left(\begin{array}{ccccc} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{array} \right) \end{matrix}$$

Figure 4.10 Une table d'incidence statistiquement localisante

Le système est soumis à des perturbations qui peuvent engendrer de fausses alarmes. Le vecteur résidu engendré est alors dégradé : un « un » remplace un « zéro » par exemple. Dans le cas d'une table d'incidence statistiquement localisante, le diagnostic est suffisamment robuste pour localiser le défaut : on qualifie alors la localisation de forte. À l'inverse dans le cas d'une table d'incidence déterministiquement localisante, la localisation est dite faible, puisque face à un résidu dégradé, le diagnostic est faussé.

4.4 LA SYNTHÈSE

Dans ce chapitre, nous nous sommes intéressés aux méthodes de diagnostic à base de modèle, dont le principe repose sur la comparaison des mesures et de leurs estimées analytiques.

Lors des deux précédents chapitres, nous avons évoqué d'une manière générale la problématique des systèmes de diagnostic, en particulier des systèmes à base de modèles quantitatifs. Le diagnostic à base de modèle est largement traité dans la littérature [Chow et Willsky, 1984 ; Gentil, 1997 ; Patton *et al.*, 1989 ; Maquin *et al.*, 1997] et de nombreuses applications industrielles exploitent ce principe [Isermann, 1994, 1997, Venkatasubramanian *et al.*, 2003]. La plupart des méthodes de cette approche sont conçues en trois étapes : la génération de résidus, l'évaluation de résidus et la localisation.

La conception d'une méthodologie générique de diagnostic à base de modèle est encore aujourd'hui un enjeu scientifique important. Le chapitre suivant est destiné à la conception d'une méthode originale de détection et de diagnostic de défauts, suffisamment générale pour permettre son utilisation pour la plupart des systèmes dynamiques hybrides. Nous la nommons *SimAEM*. Son application consiste ici en la surveillance des procédés batch mais son caractère générique lui vaut d'être applicable pour tout type de système continu. Cette proposition est mise en œuvre au cours des chapitres suivants (6, 7 et 8).

PARTIE 3 :

LA SURVEILLANCE :

CONCEPTS ET MISE EN OEUVRE

CHAPITRE 5 LA FORMALISATION DE LA DÉTECTION ET LE DIAGNOSTIC DE FAUTES

Les chapitres 3 et 4 ont dressé une synthèse des différentes approches de diagnostic, en mettant plus particulièrement l'accent sur les approches à base de modèles appelées FDI (Fault Detection and Isolation). Dans le cadre de ces travaux, une approche mixte est proposée et conduit au développement d'une méthodologie originale de diagnostic en ligne, SimAEM. Ce chapitre se propose d'en décrire les concepts.

Pour se faire, il aborde les différentes étapes d'un système de surveillance de FDI. Dans un premier temps, la génération de résidus à partir d'un filtre de Kalman étendu est développée. Dans cette partie, nous nous focalisons sur la mise en équations du problème. Dans un second temps, l'évaluation des résidus et la définition d'une signature non binaire sont introduites. Ensuite, une technique de détermination de la matrice d'incidence est proposée. Enfin la localisation est présentée : elle consiste en l'exploitation de la signature et de la table d'incidence en termes de comparaison de distance. Des distances sont alors définies, en particulier, la distance de Manhattan améliorée qui permet de localiser des défauts multiples (intervenant simultanément).

Un des premiers objectifs de nos travaux est d'élaborer et de mettre en œuvre une méthodologie de détection et diagnostic de fautes suffisamment générale pour être applicable à la plupart des systèmes dynamiques hybrides. Pour notre part, nous visons plus particulièrement le domaine du Génie des Procédés.

5.1 L'ARCHITECTURE GÉNÉRALE DU SYSTÈME DE SURVEILLANCE *FDIPRO*

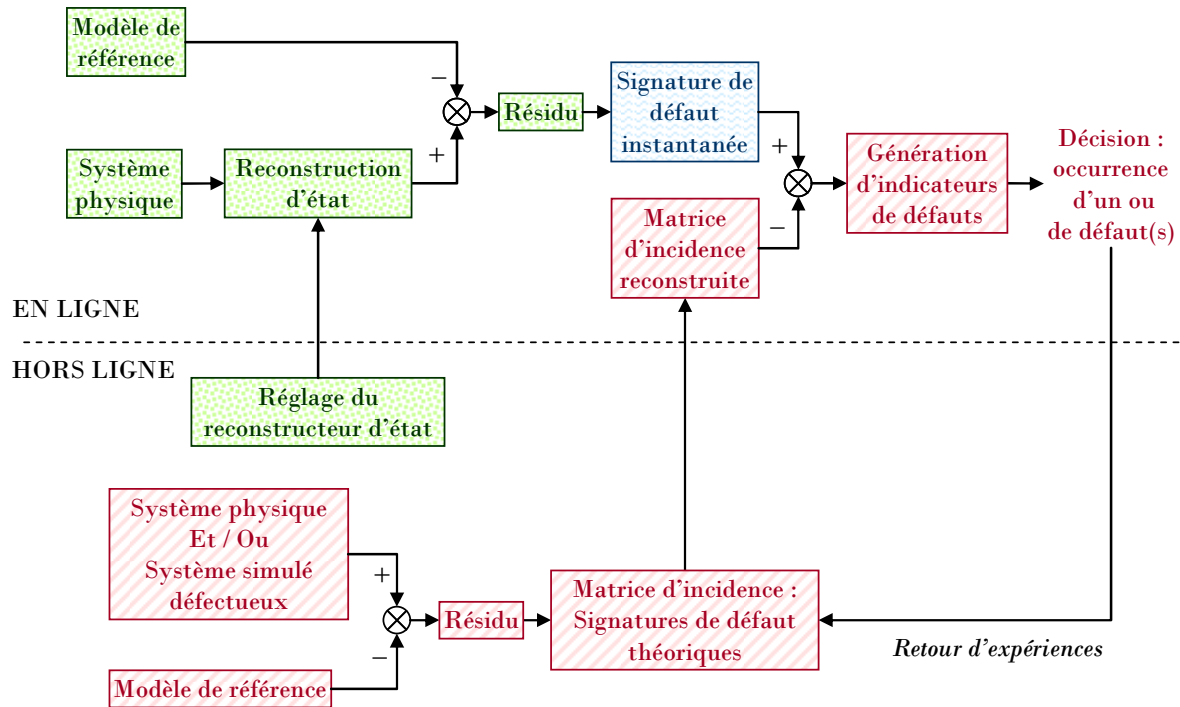


Figure 5.1 L'architecture générale du système de surveillance *FDIPRO*

L'architecture générale du système de surveillance *FDIPRO* (*Fault Detection and Isolation for Process*) est représentée sur la figure 5.1. Ce système s'appuie sur la méthodologie *SimAEM* (*Simulation for Abnormal Event Management*) développé au cours des chapitres 5 et 6. La séquence des différentes opérations conduisant au diagnostic d'une défaillance y est soulignée, ainsi que la distinction entre les opérations faites en ligne et celles faites hors ligne. La méthodologie adoptée se décompose donc en trois étapes :

- La première étape est la *génération de résidus* (en vert et en pointillé sur le schéma). Elle consiste à transformer les mesures du système physique en une structure souhaitée. Pour cela, on s'est intéressé à la conception d'observateurs d'état. En effet, l'utilisation d'un observateur pour des études de diagnostic permet la génération systématique de résidus. Un bref état de l'art sur ce sujet a été établi au chapitre 4. Les méthodes basées sur les reconSTRUCTEURS d'état ont l'avantage d'être plus robustes aux perturbations du système (erreur de modélisation et bruits des mesures). Parmi elles, se trouvent le filtre de Kalman. Dans ce cadre, cette méthode et ses variantes ont été présentées et les avantages du

filtre de Kalman étendu dans un contexte non-linéaire ont été soulignés. Ce dernier a ainsi été mis en œuvre. Notons que son réglage s'effectue hors ligne. La comparaison de l'état reconstruit avec celui obtenu par la simulation du modèle de référence conduit à la génération de résidus.

- La seconde étape, l'évaluation des résidus vise à analyser les résidus (en bleu et en forme de vague sur le schéma). C'est l'étape de détection proprement dite. Elle permet de déterminer la présence ou non d'une défaillance en évaluant les résidus à partir de seuils obtenus préalablement par des études statistiques. Plus particulièrement, elle engendre une structure plus complexe que le résidu, la signature. La notion de signature instantanée de défauts est par ailleurs introduite dans ce chapitre, et son évaluation y est soulignée.
- Enfin, la dernière étape (en rose et en hachuré sur le schéma) consiste en la localisation de défaut ou le diagnostic. Cette étape exploite la structure des signatures engendrées à l'étape précédente afin de déterminer le type de défaut présent. Pour cela, un processus en ligne d'appariement a été mis en place afin d'apparenter la signature de défaut instantanée à une ou plusieurs signatures de défauts théoriques. Pour cela, la signature de défauts instantanée est alors comparée à ces précédentes signatures par le calcul de distances afin d'identifier et localiser le (s) défaut(s). Un défaut existe alors sur le système physique quand le résidu s'apparente à une signature de défaut connue. Ces signatures de défauts théoriques sont listées dans la table d'incidence. Cette dernière est obtenue par expérience ou par la simulation hors-ligne d'un procédé défectueux. La structure de cette matrice doit être la plus générale possible afin qu'elles puissent être appliquées à n'importe quel point de fonctionnement. Notons qu'un travail en ligne sur cette matrice est effectué afin de la reconstruire selon l'état du système.

5.2 LA GÉNÉRATION DE RÉSIDUS AVEC LE FILTRE DE KALMAN ÉTENDU

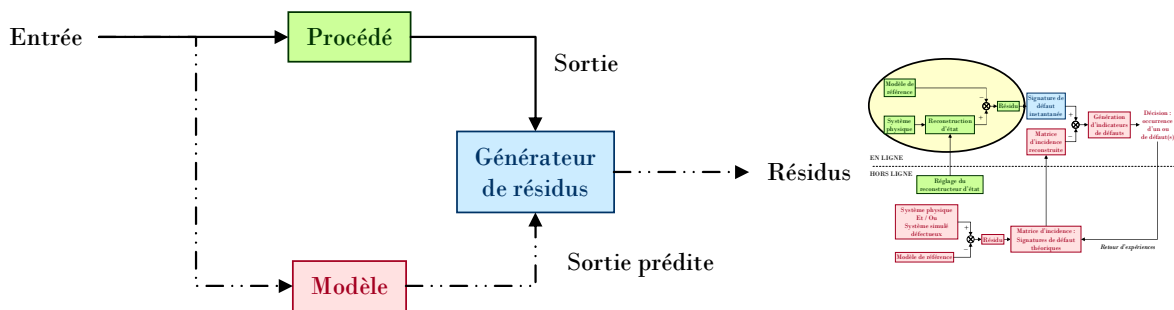


Figure 5.2 La génération de résidus

L'étape initiale d'un système de diagnostic à base de modèle génère des indicateurs de défauts, appelés résidus (cf. figure 5.2). Ces derniers contiennent des informations sur les dégradations ou défaillances du système surveillé. Le principe est de mesurer l'écart entre les mesures du système, et la valeur dite « théorique » obtenue par la simulation en parallèle du modèle de référence en bon

fonctionnement. La génération de résidus est une étape déterminante du succès du diagnostic en raison du conditionnement des résidus.

Dans cette section, nous présentons le problème de génération de résidus à l'aide du filtre de Kalman étendu et sa mise en équation. Nous expliciterons les méthodes de conception de résidus et l'exploitation des résidus générés pour des études de détection et diagnostic de défaut.

5.2.1 Le modèle utilisé

La plate-forme *PrODHyS* permet la modélisation et la simulation dynamique des systèmes dynamiques hybrides, et plus particulièrement les procédés. Ces derniers conduisent souvent à un jeu mixte d'équations différentielles ordinaires ou partielles à un ordre quelconque couplées avec des équations algébriques non linéaires. Ainsi, les équations différentielles ordinaires représentent généralement les lois de bilan de conservation de la matière, de l'énergie et de la quantité de mouvement établies en régime transitoire et les équations algébriques non-linéaires proviennent des relations constitutives ou phénoménologiques traduisant, entre autres, 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 générale du modèle simulé peut donc être représentée sous la forme implicite suivante :

$$\begin{cases} F(x^{(n)}, x, p, t) = 0 \\ x_{t=0} = x_0 \\ x^{(n)}_{t=0} = x_0^{(n)} \end{cases} \quad (5 - 1)$$

Où F est le système d'équations (différentielles et/ou algébriques) ;

x représente le vecteur d'état du système ;

$x^{(n)}$ constitue l'ensemble des dérivées d'ordre n , du vecteur d'état, par rapport à la variable indépendante (ici le temps) ;

p symbolise l'ensemble des paramètres opératoires du système ;

t est la variable indépendante (ici le temps).

Une fois ce modèle établi, il fait l'objet d'une résolution mathématique. Cette résolution est assurée par le module *DISCo*. Le lecteur trouvera dans l'annexe C une présentation plus détaillée du module *DISCo* ainsi qu'une description de la méthode de *Gear* [Gear, 1971] sur laquelle le solveur s'appuie pour la résolution du système. Notons néanmoins que la résolution globale nécessite les deux informations suivantes :

- $\frac{\partial F}{\partial x}$ la matrice des dérivées des équations par rapport au vecteur d'état,
- $\frac{\partial F}{\partial x^{(n)}}$ la matrice des dérivées des équations par rapport à leurs dérivées.

5.2.2 L'observateur : le filtre de Kalman étendu

Le filtre de Kalman étendu est l'observateur utilisé dans cette étude. Son algorithme estime la trajectoire de x_k du système à partir des signaux d'entrée et de sortie u_k et z_k .

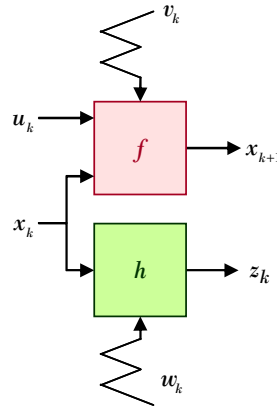


Figure 5.3 Le schéma bloc du système

D'après la théorie développée au chapitre précédent (cf. le point 4.2.2.2), l'observateur que nous avons développé nécessite que le système d'équations puisse s'écrire sous la forme discrète suivante (cf. figure 5.3) :

$$\begin{cases} x_{k+1} = f(x_k, u_k, p) + v_k + \Delta t_k \\ z_k = h(x_k) + w_k \end{cases} \quad (5 - 2)$$

Où $x_k \in \mathfrak{R}^n$ est le vecteur d'état ;

f est une fonction non linéaire qui définit le modèle du système,

$u_k \in \mathfrak{R}^p$ est le vecteur d'entrée,

$v_k \in \mathfrak{R}^n$ représente les erreurs de modèle (bruit d'état),

$z_k \in \mathfrak{R}^m$ est le vecteur de sortie (sortie évaluée à partir du nouvel état),

h définit la sortie, les observations,

$w_k \in \mathfrak{R}^m$ est le vecteur qui représente le bruit des mesures.

p symbolise l'ensemble des paramètres opératoires du système ;

Δt_k est le pas d'intégration.

Afin de pouvoir traiter numériquement le système décrit par le modèle (5 - 1), il est nécessaire de le décrire sous la forme discrète (5 - 2). Pour cela, nous avons envisagé deux solutions : la méthode de Newton-Raphson ou une méthode itérative (par exemple, les méthodes de Jacobi ou de Gauss-Seidel).

La méthode de Newton –Raphson

On suppose qu'on a réussi à localiser une solution x_0 voisine de la solution présumée du système

(5 - 1). Au voisinage de la solution, le système peut alors s'écrire de la façon suivante :

$$F(x^{(n)}, x, p, t) = F(x^{(n)}, x_0 + \Delta x, p, t) = F(x^{(n)} \Big|_{x=x_0}, x_0, p, t) + J(x_0) \cdot \Delta x = 0 \quad (5 - 3)$$

Avec $J(x_0)$ la matrice Jacobienne $J(x)$ du système, évaluée en $x = x_0$, soit :

$$J(x) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \dots & \dots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \dots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

On obtient alors la discrétisation suivante :

$$x_{k+1} = x_k + \Delta x_k \text{ avec } \Delta x_k = -[J(x_k)]^{-1} \cdot F(x^{(n)} \Big|_{x=x_k}, x_k, p) \quad (5 - 4)$$

La méthode itérative

Dans cette méthode, on écrit le système (5 - 1) sous la forme :

$$x = R(x) \text{ en posant } R(x) = x + F(x^{(n)}, x, p, t) \quad (5 - 5)$$

On obtient alors la discrétisation suivante :

$$x_{k+1} = G(x_k) = x_k + F(x^{(n)} \Big|_{x=x_k}, x_k, p, \Delta t_k) \quad (5 - 6)$$

Dans le cadre de cette étude, le résidu généré repose sur l'estimation de variables d'état par un observateur, ici le filtre de Kalman étendu. Son algorithme a été présenté au chapitre précédent au point 4.2.2.2. Comme le système est non-linéaire (cf. le système (4 - 26)), il estime la trajectoire de ce dernier, c'est-à-dire de x_{k+1} , à partir des signaux d'entrée et de sortie u_k et z_{k+1} , et du calcul au point précédent (x_k) de la fonction f représentant le système et de sa dérivée f' .

Dans le cas de la méthode de Newton-Raphson, l'équation (5 - 4) nous permet d'écrire l'expression discrétisée de la fonction f :

$$\begin{aligned} x_{k+1} &= f(x_k) \text{ avec} \\ f : \mathfrak{X}^n &\rightarrow \mathfrak{X}^n \\ x_k &\mapsto x_k - [J(x_k)]^{-1} \cdot F(x^{(n)} \Big|_{x=x_k}, x_k, p, \Delta t_k) \end{aligned} \quad (5 - 7)$$

Dans le cas de la méthode itérative, l'équation (5 - 4) nous permet d'écrire l'expression discrétisée de la fonction f :

$$\begin{aligned} x_{k+1} &= f(x_k) \text{ avec} \\ f : \mathfrak{X}^n &\rightarrow \mathfrak{X}^n \\ x_k &\mapsto x_k + F(x^{(n)} \Big|_{x=x_k}, x_k, p, \Delta t_k) \end{aligned} \quad (5 - 8)$$

L'état théorique x_k n'étant pas connu, il sera remplacé par son estimation \hat{X}_k .

C'est une approche s'apparentant à la méthode de Newton – Raphson qui est utilisée dans le solveur employé (solveur *DISCO*, cf. annexe C). Afin de ne pas alourdir le traitement et allonger le temps de réponse, il semble préférable d'exploiter autant que possible des résultats déjà disponibles. C'est pourquoi nous avons retenu la méthode de Newton – Raphson et l'expression de la fonction f associée, c'est-à-dire l'expression (5 - 7). Le filtre de Kalman étendu implémenté, le système (4 - 26) est donc appliqué à cette fonction.

On obtient alors une estimation de notre état sous la forme simplifiée suivante :

$$\hat{X}_{k+1|k+1} = f(\hat{X}_{k|k}) + K_{k+1} \cdot (Z_{k+1} - h(\hat{X}_{k+1|k})) \quad (5 - 9)$$

Où : $\hat{X}_k \in \mathfrak{R}^n$ est l'estimée du vecteur d'état du système,

$K_{k+1} \in \mathfrak{R}^{n \times m}$ est le gain de Kalman calculé à partir du système (4 - 26),

$Z_{k+1} \in \mathfrak{R}^m$ est le vecteur d'observations (mesures),

Notons que lorsqu'aucune observation du système n'est disponible, l'équation (5 - 9) se résume en :

$$\hat{X}_{k+1|k+1} = f(\hat{X}_{k|k}) \quad (5 - 10)$$

En principe, la conception de l'observateur assure la propriété suivante :

Propriété 5-1 :

Si la représentation obtenue par le modèle (5 - 2) est parfaite ($w_k = \bar{0}$) et n'est soumise à aucun bruit de mesure ($w_k = \bar{0}$), alors \hat{X}_k tend vers le vrai état x_k quand k tend vers l'infini.

Notons également que la prédiction de la mesure \hat{Z}_{k+1} est obtenue par l'équation suivante :

$$\hat{Z}_{k+1} = h(\hat{X}_{k+1|k}) \quad (5 - 11)$$

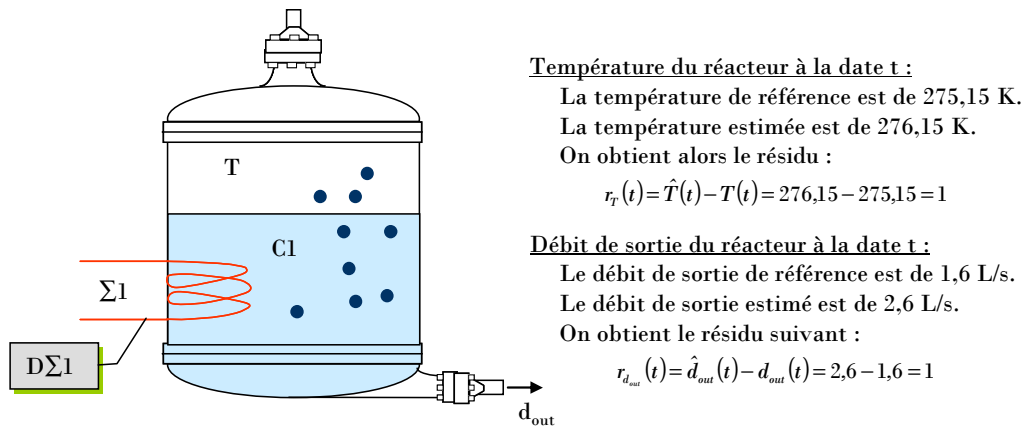
Remarque : il est à noter que les entrées du système n'apparaissent pas explicitement dans les équations du modèle généré par PrODHyS (système (5 - 1)). Néanmoins le système instancié à la date t dépend des entrées du système (commande).

5.2.3 La génération de résidus

À partir de la relation (5 - 9), des indicateurs de défaut sont générés. Nous appellerons ces indicateurs de défauts, *résidus*, et les notons $r(t)$. Ces derniers résultent de la comparaison entre le vecteur d'état reconstruit par l'observateur représentant l'état estimé, $\hat{X}(t)$, et le vecteur d'état obtenu par la résolution du modèle de référence $X(t)$:

$$r(t) = \hat{X}(t) - X(t) \quad (5 - 12)$$

Nous qualifions ce type de résidu d'« absolu ». Illustrons ce concept au travers d'un exemple simple. Considérons un réacteur C1 chauffée par une source d'énergie $\Sigma 1$.



Température du réacteur à la date t :

La température de référence est de 275,15 K.

La température estimée est de 276,15 K.

On obtient alors le résidu :

$$r_T(t) = \hat{T}(t) - T(t) = 276,15 - 275,15 = 1$$

Débit de sortie du réacteur à la date t :

Le débit de sortie de référence est de 1,6 L/s.

Le débit de sortie estimé est de 2,6 L/s.

On obtient le résidu suivant :

$$r_{d_{out}}(t) = \hat{d}_{out}(t) - d_{out}(t) = 2,6 - 1,6 = 1$$

Figure 5.4 Les résidus absolus

Afin de déterminer les variables représentatives du comportement anormal, il est nécessaire de comparer le résidu respectivement de la température T et du débit d_{out} . Or, bien que semblables en valeur, ces derniers ne sont pas adimensionnels : $r_T(t) = 1 \text{ K}$ et $r_{d_{out}}(t) = 1 \text{ L/s}$. Pour pouvoir les comparer, il faut donc les affranchir de toutes grandeurs physiques. Pour cela, nous avons défini un nouveau résidu dit « relatif » :

$$r^r(t) = \frac{\hat{X}(t) - X(t)}{X(t)} \quad (5 - 13)$$

Ainsi, on obtient les résidus relatifs suivants :

$$r_T^r(t) = \frac{\hat{T}(t) - T(t)}{T(t)} = \frac{276,15 - 275,15}{275,15} = 0,36\%$$

$$\text{et } r_{d_{out}}^r(t) = \frac{\hat{d}_{out}(t) - d_{out}(t)}{d_{out}(t)} = \frac{2,6 - 1,6}{1,6} = 62,5\%$$

On peut ainsi constater que l'une des variables représentatives de l'état anormal du système est le débit de sortie tandis que la température est dans des conditions normales.

5.3 L'ÉVALUATION DE RÉSIDUS

5.3.1 Un travail sur fenêtre glissante

Le fonctionnement en temps réel est un facteur important de la détection de défaut. En effet, une détection précoce de dysfonctionnement est un atout si on veut y remédier et ainsi éviter leurs conséquences sur le fonctionnement du processus parfois désastreuses (cf. le point 1.4.1).

Par ailleurs, les informations passées peuvent aider à la compréhension du comportement présent. Au cours du temps, petit à petit, les observations sont collectées, suivant leur disponibilité. Intuitivement,

on se doute que lorsque l'horizon de temps est grand ($t \gg 1$), les données de l'instant initial n'auront pas d'influence sur les résidus de l'instant t .

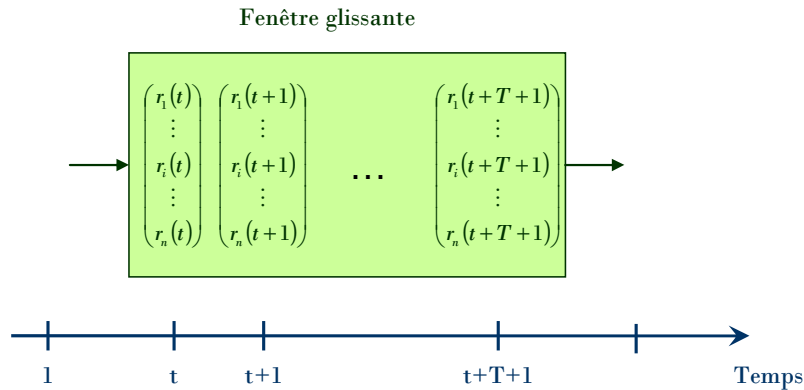


Figure 5.5 La génération de résidus sur fenêtre glissante

Il ne nous est donc pas nécessaire de stocker toutes les données. On définit alors une fenêtre d'observation de taille T à travers laquelle on observe le système. Le stockage de l'information se résume à cette fenêtre. Cette dernière est alors représentative de l'état du système. Sa taille est un paramètre choisi en fonction de la dynamique du système.

Comme le montre la figure 5.5, lors de la génération de nouveaux résidus, la fenêtre d'observation glisse vers la droite.

5.3.2 La génération des signatures

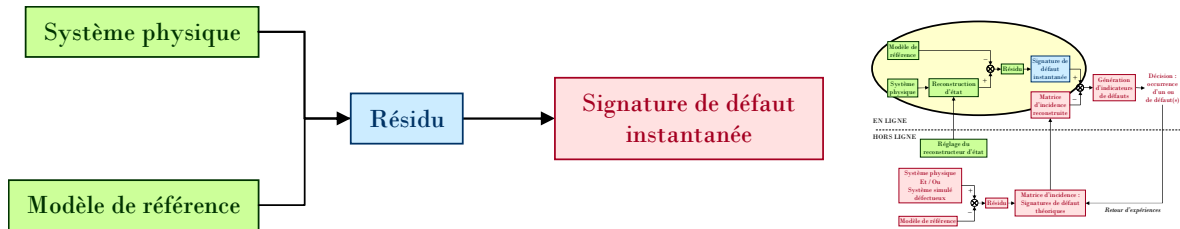


Figure 5.6 L'évaluation des résidus

La détection consiste ensuite à évaluer une signature instantanée à partir du résidu engendré dans la première étape. Nous notons cette signature instantanée \mathcal{S} . Cette signature est un vecteur défini positif de dimension n (la taille du vecteur d'état). Plus particulièrement, chaque composante de ce vecteur est un réel positif dont les éléments sont le résultat d'un test de seuillage. On définit donc un élément de la signature de la façon suivante :

$$\mathcal{S}_i(t) = \begin{cases} 0 & \text{si } |r_i(t)| \leq \varepsilon_i(t) \\ \alpha_i > 0 & \text{si } |r_i(t)| > \varepsilon_i(t) \end{cases} \quad \text{avec } i \in [1; n] \quad (5 - 14)$$

Où α_i est le résultat du test de dépassement de seuil,

$r_i(t)$ est le résidu généré dans la première partie du diagnostic,

$\varepsilon_i(t)$ est le seuil adaptatif de détection.

\mathcal{S} est une signature de défaut instantanée puisqu'une composante non nulle de ce vecteur présume l'occurrence d'un défaut ($\mathcal{S}_i(t) = \alpha_i > 0$ avec $i \in [1; n]$). La nullité de ce vecteur suppose a priori un comportement normal du système surveillé ($\mathcal{S}_i(t) = 0$ pour $i = 1..n$). Le vecteur nul transcrit le fait que le résidu reste dans le domaine normal et donc qu'aucune alarme n'est déclenchée.

Nous aurions très bien pu définir ce vecteur de signature comme étant un vecteur binaire : si le résidu dépasse le seuil alors la signature vaut « 1 ». Néanmoins en définissant de la sorte le vecteur de signature, il y a une perte d'information sur l'ampleur de la défaillance : de combien y a-t-il dépassement ? Ce dépassement n'est-il pas négligeable ? En ne définissant pas le vecteur de signature comme étant un booléen, l'émission de fausses alarmes est ainsi évitée et le cas d'une dégradation et celui d'une défaillance pourront être différenciés.

La signature de défaut instantanée à l'instant t résulte donc de la comparaison du résidu $r(t)$ et du seuil de détection fixé préalablement $\varepsilon(t)$. Nous pouvons maintenant donner la définition formelle de la signature de défaut instantanée.

Définition 5-2 : signature instantanée de défauts

La signature de défauts instantanée $\mathcal{S}(t)$ est un vecteur fonction du résidu $r(t)$ et du seuil de détection $\varepsilon(t)$ dont chaque composante \mathcal{S}_i est définie de la façon suivante :

$$\mathcal{S}_i(t) = \text{Max} \left[\left(|r_i(t)| - \varepsilon_i(t) \right); 0 \right] \quad \text{avec } i \in [1; n]$$

Dans le point précédent, nous avons vu qu'il était intéressant de travailler avec des résidus relatifs afin de s'affranchir de toute dimension. De la même façon, nous définirons donc une signature instantanée de défauts relative :

Définition 5-3 : signature instantanée de défauts relative

La signature de défauts instantanée $\mathcal{S}^r(t)$ est un vecteur fonction du résidu $r^r(t)$ du seuil de détection $\varepsilon(t)$ et du vecteur d'état de référence $X(t)$. Chaque composante \mathcal{S}_i^r est définie de la façon suivante :

$$\mathcal{S}_i^r(t) = \text{Max} \left[\left(|r_i^r(t)| - \varepsilon'_i(t) \right); 0 \right] \quad \text{avec } i \in [1; n]$$

$$\text{avec } r_i^r(t) = \frac{\hat{X}_i(t) - X_i(t)}{X_i(t)} \quad \text{et } \varepsilon'_i(t) = \frac{\varepsilon_i(t)}{X_i(t)}.$$

Enfin, il est intéressant de normaliser ces signatures afin de constater les variations prépondérantes. Ainsi, la signature de défauts relative normalisée s'écrit :

$$\mathcal{S}_i^{rN}(t) = \frac{\mathcal{S}_i^r(t)}{\sum_{k=1}^n \mathcal{S}_k^r(t)} = \frac{\text{Max} \left[\left(|r_i^r(t)| - \varepsilon'_i(t) \right); 0 \right]}{\sum_{k=1}^n \text{Max} \left[\left(|r_k^r(t)| - \varepsilon'_k(t) \right); 0 \right]} \quad \text{avec } i \in [1; n] \quad (5 - 15)$$

$$\text{avec } r_i^r(t) = \frac{\hat{X}_i(t) - X_i(t)}{X_i(t)} \quad \text{et } \varepsilon'_i(t) = \frac{\varepsilon_i(t)}{X_i(t)}.$$

Par conséquent, la somme de toutes les composantes de la signature de défauts relative normalisée vaut 1. Ceci traduit l'heuristique suivante : plus un résidu r_i^f est sensible à un défaut, moins les autres résidus r_k^f (avec $k \neq i$) le seront.

5.4 LA DÉTERMINATION DE LA MATRICE D'INCIDENCE

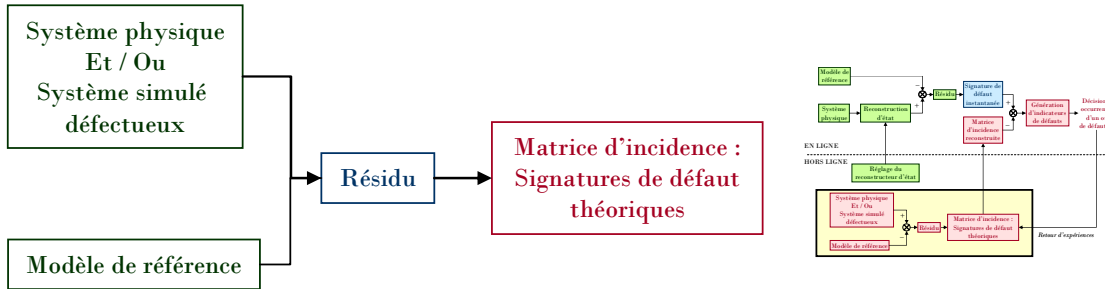


Figure 5.7 La génération de la matrice d'incidence

Beaucoup de travaux ont été élaborés concernant la distance aux signatures de défauts ou sur les propriétés structurelles de la matrice d'incidence afin de définir des critères de robustesse pour l'étape de localisation [Gertler et Singer, 1990 ; Fang et Ge, 1988 ; Chin et Danai, 1991]. La matrice d'incidence est construite à partir des connaissances sur le système physique. Dans le domaine industriel, l'acquisition de ces connaissances repose généralement sur une démarche expérimentale. L'obtention de la matrice d'incidence est communément faite hors-ligne.

Une signature de défaut est caractéristique d'un résidu et d'un défaut particuliers. Cette signature est obtenue par expérience (ou simulation dans notre cas). La démarche consiste à évaluer une signature à partir du résidu généré par la comparaison du modèle de référence à l'expérience ou la simulation du procédé défectueux (cf. figure 5.7). Plus particulièrement, chaque composante de ce vecteur est un réel positif dont les éléments sont le résultat d'un test de seuillage (cf. équation (5 - 14)).

Dans notre cas, nous nous basons sur un certain nombre de simulations du même défaut à différents instants. Nous générons les signatures caractéristiques de ce défaut pour les p simulations du procédé défectueux. Le but est alors de globaliser ces p signatures en une seule et unique signature caractéristique du défaut. Deux cas se présentent alors :

- Les signatures caractérisent le même vecteur d'état. On suppose dans ce cas, que les p simulations ont la même importance, c'est-à-dire que leurs occurrences sont équiprobables. La signature caractéristique correspond au centre de gravité des p signatures obtenues par « simulation ». Soulignons que pour des systèmes complexes, il est intéressant d'analyser les données et de déterminer leurs composantes principales. On obtient alors une représentation approchée du nuage des p simulations dans un sous-espace de dimension faible.
- Les signatures ne caractérisent pas le même vecteur d'état (pas le même nombre de variables d'état). Il est alors nécessaire d'effectuer une analyse canonique. Considérons deux groupes de simulations caractérisant le même défaut mais l'une le vecteur d'état 1 et l'autre le vecteur d'état 2. Cette analyse

consiste en l'examen des liens existants entre ces deux ensembles afin de savoir s'ils mesurent ou non les mêmes propriétés. Elle est basée sur une décomposition de type *Analyse par Composantes Principales*. Nous n'exposerons pas ici cette théorie mais nous recommandons au lecteur de se reporter au chapitre 9 du livre de Saporta G. « Probabilités des données et statistiques » [Saporta, 1990]. Notons toutefois que si ces deux espaces sont confondus, cela prouve qu'un seul des deux ensembles suffit, puisqu'ils ont le même pouvoir de description. Inversement, si ces deux ensembles sont orthogonaux, c'est que les deux ensembles ne mesurent pas les mêmes propriétés. Il faudra alors considérer deux signatures de défauts différentes caractérisant néanmoins le même défaut.

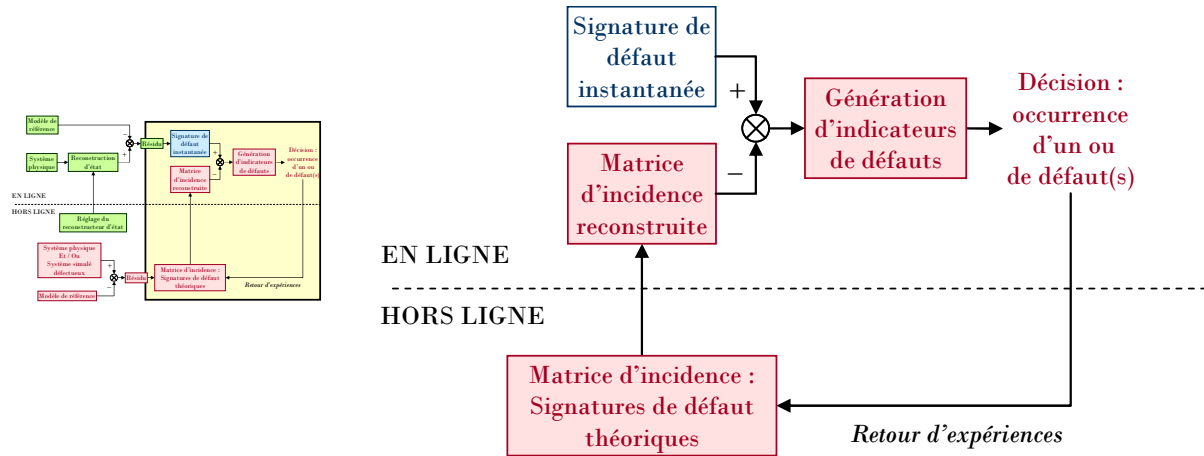


Figure 5.8 La matrice d'incidence reconstruite

Une fois la matrice d'incidence globale obtenue, il est important de reconstruire en ligne une matrice d'incidence en ligne adaptée à l'état du système (cf. figure 5.8). Pour cela, on réduit la matrice d'incidence en ne faisant intervenir que les résidus présents dans la signature de défauts instantanée. Enfin, on normalise chaque signature de défaut théorique. Effectuons cette démarche sur un exemple simple :

On suppose que la signature obtenue est la suivante :

$$S^{rN} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \end{pmatrix} \begin{matrix} r_1 \\ r_3 \\ r_4 \end{matrix}$$

De même, on considère la matrice d'incidence suivante :

$$\mathcal{T} = \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ \begin{matrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/2 & 1 & 0 & 1/3 \\ 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 1/2 & 0 & 1/3 & 0 \end{pmatrix} \end{matrix}$$

On constate ici qu'elle n'est pas adaptée à la signature instantanée car elle fait intervenir le résidu r_2 qui n'apparaît pas dans la signature de défaut instantanée. Un travail sur cette matrice est donc nécessaire. On supprime donc la 2^{ème} ligne de la matrice d'incidence et on normalise les signatures de défaut obtenues.

$$\mathcal{T} = \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ r_1 & 1 & 0 & 0 & 1/3 & 1/3 \\ r_3 & 0 & 0 & 0 & 1/3 & 1/3 \\ r_4 & 0 & 1/2 & 0 & 1/3 & 0 \end{matrix} \Rightarrow \mathcal{T}' = \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ r_1 & 1 & 0 & 0 & 1/3 & 1/2 \\ r_3 & 0 & 0 & 0 & 1/3 & 1/2 \\ r_4 & 0 & 1 & 0 & 1/3 & 0 \end{matrix}$$

Soulignons que le défaut f_3 n'est pas ici localisable.

5.5 LA LOCALISATION DE DÉFAUTS

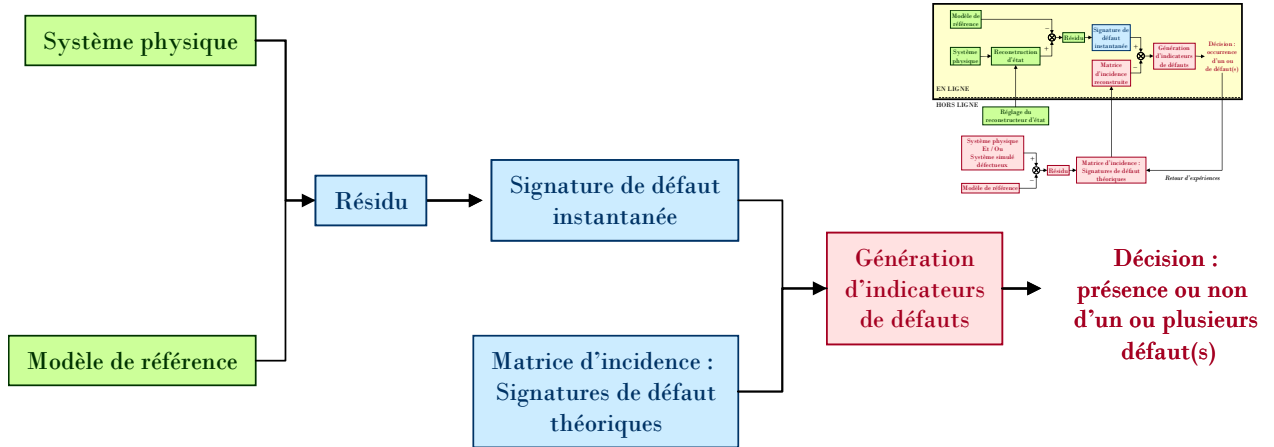


Figure 5.9 Le système de localisation

Le système de localisation est illustré par la figure 5.9. Il consiste à établir le diagnostic à partir de faits mesurés sur le processus (*Signature de défaut instantanée*) et de faits modélisés ou obtenus par expérience (*Signatures de défaut théoriques*).

5.5.1 Le principe

Nous avons défini dans le chapitre précédent la notion de table d'incidence (cf. définition 4-6). Nous l'avons défini comme une table binaire \mathcal{T} où chaque élément \mathcal{T}_{ij} est associée à un résidu r_i et au défaut f_j (i désignant la ligne et j la colonne avec $i=1..n$, n étant la taille du vecteur d'état et $j=1..m$, m le nombre de défauts considérés). La valeur « 1 » désigne que le résidu est sensible au défaut alors que la valeur « 0 » désigne que le résidu est découplé du défaut.

Les colonnes de la matrice d'incidence \mathcal{T} représente les signatures de défauts. La notation adoptée pour les colonnes de la matrice d'incidence est la suivante : $\mathcal{T}_{\cdot,j}$ ($j = 1..m$). $\mathcal{T}_{\cdot,j}$ correspond à la signature associée au $j^{ème}$ défaut f_j . De même, chaque ligne de la matrice d'incidence, notée de la même façon $\mathcal{T}_{i,\cdot}$, représente une signature du $i^{ème}$ résidu. La figure 5.10 présente un exemple de signatures de défauts théoriques et de signatures de résidus d'une matrice d'incidence.

Notre approche s'apparente à un problème de reconnaissance de formes. La forme qu'on cherche à classifier est la signature de défauts instantanée relative normée \mathcal{S}^{rN} générée dans l'étape précédente. Il

convient alors de déterminer le degré d'affectation de cette forme aux classes déjà formées ; dans notre cas, chaque classe est caractérisée par une signature de défauts théoriques $\tau_{\cdot,j}$ ($j = 1..m$).

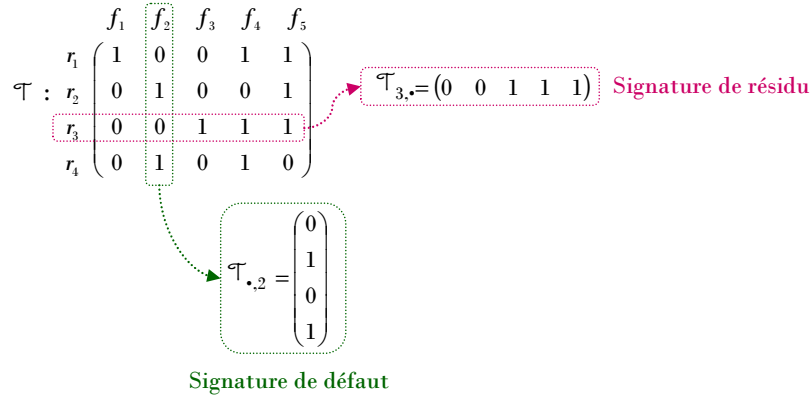


Figure 5.10 Une signature de défaut et une signature de résidu

Dans le cas de la détection et du diagnostic de défaut, la signature de défauts instantanée relative normée S^{rN} est donc comparée aux m signatures de défauts théoriques $\tau_{\cdot,j}$ ($j = 1..m$). La signature S^{rN} générée à l'étape précédente transcrit les symptômes du système physique tandis que le vecteur $\tau_{\cdot,j}$ représente le $j^{ème}$ défaut. On définit alors la propriété suivante :

Propriété 5-4 :

La finesse de la corrélation entre la signature de défauts instantanée relative normée S^{rN} et la signature de défaut théorique $\tau_{\cdot,j}$ ($j=1..m$) est directement proportionnelle à la probabilité de l'occurrence du défaut f_j ($j=1..m$).

On en déduit le lemme suivant :

Lemme 5-5 :

S'il existe $j \in [1;m]$ tel que $S^{rN} \cong \tau_{\cdot,j}$, alors le diagnostic conclut à l'occurrence du défaut f_j .

Pour comparer une signature instantanée $S^{rN}(t)$ et une signature de défaut particulière $\tau_{\cdot,j}$, on peut utiliser comme opérateur de composition une fonction de similitude ou une distance. Dans notre cas, la classification est traduite en termes de distance dans l'espace de signatures (cf. le point suivant 5.5.2) :

Définition 5-6 : distance

Soient S l'espace des signatures instantanées relatives normées et T l'espace borné des signatures de défaut théoriques ($\text{card}(T)=m$, m étant le nombre de défauts considérés). Une distance définit la corrélation symptômes – défauts. La distance entre une signature instantanée $S^{rN}(t)$ et une signature de défaut particulière $\tau_{\cdot,j}$ est définie par la fonction suivante :

$$D : S \times T \rightarrow [0;1]$$

$$(S^{rN}(t); \tau_{\cdot,j}) \mapsto D_j(t) = D(S^{rN}(t), \tau_{\cdot,j})$$

Propriété 5-7 :

La distance D vérifie les propriétés suivantes :

Soient $X \in S, Y \in T$,

1. $D(X, Y) = 0 \Rightarrow X = Y$

2. $\mathcal{D}(X, Y) = \mathcal{D}(Y, X)$
3. Soit $Z \in \mathcal{S}, \mathcal{D}(X, Z) \leq \mathcal{D}(X, Y) + \mathcal{D}(Y, Z)$

On définit alors un indicateur de défaut :

Définition 5-8 : indicateur de défaut

Un indicateur de défaut $\mathbb{I}_j \in [0;1]$ est spécifique au défaut f_j avec $j = 1..m$. Il représente la probabilité de l'occurrence du défaut. Il est défini par la relation suivante :

$$\mathbb{I}_j(t) = 1 - \mathcal{D}_i(t) = 1 - \mathcal{D}(S^{rN}(t), \mathcal{T}_{\cdot,j})$$

D'après la propriété 5-7-1, $\mathbb{I}_j(t) = 0$ signifie que le défaut f_j n'est pas occurred et inversement, $\mathbb{I}_j(t) = 1$ traduit le fait que le défaut f_j est détecté et localisé.

En général, nous n'avons pas ces égalités strictes, mais plutôt la relation d'ordre suivante :

$$0 < \mathbb{I}_j(t) < 1$$

Cette relation entraîne le déclenchement d'une alarme sur le défaut f_j . Si l'indicateur de défaut $\mathbb{I}_j(t)$ est proche de zéro, l'occurrence du défaut n'est pas prouvée et la présomption d'une fausse alarme est probable. En revanche, si $\mathbb{I}_j(t)$ est proche de un, alors l'occurrence du défaut f_j est démontrée.

5.5.2 Les distances

Généralement, la distance employée est la distance de Hamming [Kaufmann, 1977 ; Cassar, 1994 ; Vilain, 1996]. Cette dernière est une distance au sens mathématique du terme. Elle permet de comparer deux vecteurs B_1 et B_2 de binaires de même taille. Elle est égale à la somme des valeurs absolues des différences, composante par composante des deux vecteurs B_1 et B_2 , c'est-à-dire au nombre de bits différents, ou en encore au cardinal de l'ensemble $\{i | B_{1i} \neq B_{2i}\}$:

$$\mathcal{D}^H = \sum_{i=1}^n |B_{1i} - B_{2i}| \tag{5 - 16}$$

Par exemple considérons deux vecteurs binaires B_1 et B_2 , et calculons la distance de Hamming associée :

$$B_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} ; B_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow \mathcal{D}^H = \sum_{i=1}^4 |B_{1i} - B_{2i}| = |0-1| + |1-1| + |0-0| + |1-1| = 1$$

Figure 5.11 Un exemple de calcul de distance de Hamming

La distance de Hamming est égale à 1 ; ceci signifie qu'une seule composante diffère entre les deux vecteurs (la première composante).

Cette notion de distance peut être généralisée à des vecteurs non binaires : on parle alors de la distance de Manhattan.

Nous avons fait le choix dans le point précédent de ne pas travailler en binaire mais dans l'ensemble continu [0;1].

Afin d'uniformiser cette distance pour toutes les signatures, la distance de Hamming relative a été définie [Theillol *et al.*, 1995]. Cette distance est calculée par la relation suivante [Ripoll, 1999] :

Définition 5-9 : distance de Hamming relative

Soient deux vecteurs binaires B_1 et B_2

La distance de Hamming relative entre B_1 et B_2 est définie par la relation suivante :

$$D^{Hr}(t) = \frac{\sum_{i=1}^n |B_{1i} - B_{2i}|}{n}$$

Nous généralisons cette définition au cas non binaire et nous définissons ainsi une nouvelle distance que nous nous avons appelées : la *distance de Manhattan relative*.

Définition 5-10 : distance de Manhattan relative

Soient S l'espace des signatures instantanées relatives normées et T l'espace borné des signatures de défaut théoriques ($\text{card}(T)=m$, m étant le nombre de défauts considérés).

La distance de Manhattan relative entre une signature instantanée $S^{rN}(t)$ et une signature de défaut particulière $\tau_{\cdot,j}$ (toutes deux de dimension n) est définie par la relation suivante :

$$D_j^{Mr}(t) = \frac{\sum_{i=1}^n |S_i^{rN}(t) - \tau_{ij}|}{n}$$

La démonstration de cette définition se trouve en annexe D.

On en déduit alors le lemme suivant :

Lemme 5-11 :

Soient S l'espace des signatures instantanées relatives normées et T l'espace borné des signatures de défaut théoriques ($\text{card}(T)=m$, m étant le nombre de défauts considérés).

Soient $S^{rN}(t) \in S$ et $\tau_{\cdot,j} \in T$.

Si la distance de Manhattan relative entre $S^{rN}(t)$ et $\tau_{\cdot,j}$ est nulle alors on a l'égalité :

$$S^{rN}(t) = \tau_{\cdot,j}$$

La contraposée du lemme 5-11 est vraie :

Lemme 5-12 :

Soient S l'espace des signatures instantanées relatives normées et T l'espace borné des signatures de défaut théoriques ($\text{card}(T)=m$, m étant le nombre de défauts considérés).

Soient $S^{rN}(t) \in S$ et $\tau_{\cdot,j} \in T$.

Si $S^{rN}(t) \neq \tau_{\cdot,j}$ alors la distance de Manhattan relative entre $S^{rN}(t)$ et $\tau_{\cdot,j}$ est non nulle.

Un des problèmes majeurs des systèmes FDI réside en leur capacité à détecter l'occurrence de défauts multiples et à les localiser. En effet, les signatures théoriques caractérisent un défaut particulier. Or, l'occurrence de défauts multiples est représentée par une nouvelle signature de défaut [Koscielny, 1993].

Cette signature est obtenue par superposition des signatures de défauts théoriques [Ripoll, 1999]. Ceci est illustré sur la figure 5.12.

$$\mathcal{T} : \begin{matrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{matrix} \begin{pmatrix} f_1 & f_2 & f_3 & f_4 & f_5 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$f_{23} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Signature de l'occurrence des défauts f_2 et f_3

Figure 5.12 La signature de défauts multiples

La prise en compte de toutes les combinaisons linéaires des signatures théoriques n'est pas une solution satisfaisante en raison de l'explosion combinatoire engendrée. Il faut donc utiliser une méthode qui s'affranchit des tests de combinaison. Ainsi, Theillol *et al.* [Theillol *et al.*, 1995] ont défini un indicateur de Hamming modifié, qui ne prend en compte, lors de la comparaison, que les éléments non nuls de la signature de défaut théorique :

Définition 5-13 : distance de Hamming améliorée

La distance de Hamming améliorée est définie par la relation suivante :

$$D_j^{Ha}(t) = \frac{\sum_{i=1}^n |S_i^{rN}(t) - \mathcal{T}_{ij}| \cdot \mathcal{T}_{ij}}{n'}$$

avec n' le nombre d'éléments non nuls de la signature de défauts théoriques $\mathcal{T}_{\cdot,j}$.

Nous généralisons cette distance au cas non binaire, en définissant la signature de Manhattan améliorée \mathcal{D}^{Ma} :

Définition 5-14 : distance de Manhattan améliorée

La distance de Manhattan améliorée est définie par la relation suivante :

$$D_j^{Ma}(t) = \frac{\sum_{i=1}^n |S_i^{rN}(t) \times m' - \mathcal{T}_{ij} \times n'| \cdot \mathcal{T}_{ij}}{n'}$$

Avec n' le nombre d'éléments non nuls de la signature de défauts théoriques $\mathcal{T}_{\cdot,j}$,

m' le nombre d'éléments non nuls de la signature de défauts instantanée S^{rN} .

Remarque : Les distances de Hamming améliorée et de Manhattan améliorée ne sont pas des distances mathématiquement parlant (cf. annexe D). Par abus de langage, nous les avons nommées « distance », puisque ces deux indicateurs permettent d'effectuer une comparaison entre la signature instantanée $S^{rN}(t)$ et une signature de défaut particulière $\mathcal{T}_{\cdot,j}$ en termes de similitude de symptômes du comportement anormal.

Appliquons les signatures de Manhattan relative et améliorée à un exemple concret. Considérons le cas où les défauts f_1 et f_2 ont lieu simultanément. Le vecteur de signature instantanée et la matrice d'incidence sont présentés sur la figure ci-dessous. Calculons ces distances et l'indicateur de défaut correspondant.

Considérons la signature de défauts instantanée et la matrice d'incidence suivantes :

$$S^{rN} = \begin{pmatrix} 1/3 \\ 1/3 \\ 0 \\ 1/3 \end{pmatrix} \quad \mathcal{T} = \begin{matrix} & f_1 & f_2 & f_3 & f_4 & f_5 \\ r_1 & \begin{pmatrix} 1 & 0 & 0 & 1/3 & 1/3 \end{pmatrix} \\ r_2 & \begin{pmatrix} 0 & 1/2 & 0 & 0 & 1/3 \end{pmatrix} \\ r_3 & \begin{pmatrix} 0 & 0 & 1 & 1/3 & 1/3 \end{pmatrix} \\ r_4 & \begin{pmatrix} 0 & 1/2 & 0 & 1/3 & 0 \end{pmatrix} \end{matrix}$$

Distance de Manhattan relative : $\mathcal{D}_i^{Mr} = \frac{\sum_{k=1}^4 |S_k^{rN} - \mathcal{T}_{ki}|}{4}$

| | f_1 | f_2 | f_3 | f_4 | f_5 |
|----------------------|-------|-------|-------|-------|-------|
| $\mathcal{D}^{Mr} =$ | 0,33 | 0,17 | 0,5 | 0,17 | 0,17 |
| $\mathcal{I}^{Mr} =$ | 0,67 | 0,83 | 0,5 | 0,83 | 0,83 |

← Impossibilité de conclure

Distance de Manhattan améliorée : $\mathcal{D}_i^{Ma} = \frac{\sum_{k=1}^4 |S_k^{rN} \times 3 - \mathcal{T}_{ki} \times n'|}{n'}$

| | f_1 | f_2 | f_3 | f_4 | f_5 |
|----------------------|-------|-------|-------|-------|-------|
| $\mathcal{D}^{Ma} =$ | 0 | 0 | 1 | 0,11 | 0,11 |
| $\mathcal{I}^{Ma} =$ | 1 | 1 | 0 | 0,89 | 0,89 |

← Occurrences des défauts f_1 et f_2 détectées et localisées

Figure 5.13 Les tables des distances (Manhattan relative et améliorée) et d'indicateurs de défauts correspondants

Nous voyons ici que le calcul des indicateurs de défauts à partir de la distance de Manhattan relative ne permet pas de conclure. La signature de défauts instantanée ne correspond en effet à aucune des signatures de défauts théoriques. La comparaison composante par composante est limitée. La distance de Manhattan améliorée repose sur l'idée de retrouver dans la signature de défauts instantanée uniquement les symptômes significatifs de défauts, c'est-à-dire les éléments non nuls. Avec l'indicateur de Manhattan amélioré, les deux défauts f_1 et f_2 ont été détectés et localisés.

Ces deux distances sont employées au cours des différents exemples d'application de cette thèse.

5.5.3 La prise de décision

Les indicateurs de défauts, ainsi générés, sont transmis à l'étape de décision qui consiste en la discrimination du défaut le plus vraisemblable. Afin de prendre cette décision, nous avons formulé deux règles :

- La première hypothèse porte sur la répartition des probabilités des défauts. Cette dernière est supposée suivre une loi normale centrée réduite $\mathfrak{N}(moy, \sigma)$. Cette répartition est représentée sur la figure 5.14. Ceci nous permet de définir un seuil limite correspondant à la probabilité à l'écart – type, soit inférieur à 0,68. Ainsi la présence d'un défaut n'est pas validée si son indicateur est inférieur à 0,68.
- Ensuite, afin de limiter les choix de défauts possibles, on pose l'hypothèse suivante : le nombre de défauts pouvant avoir lieu simultanément est limité à trois.

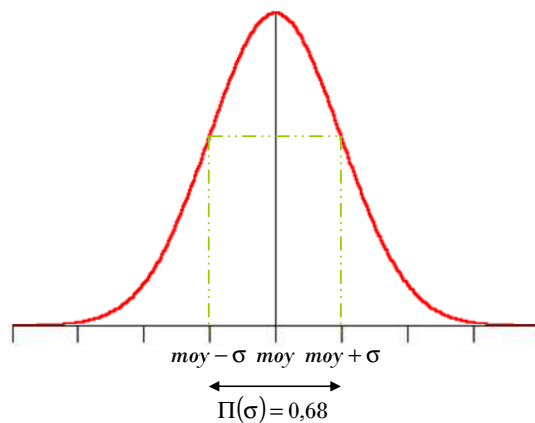


Figure 5.14 La loi normale centrée réduite $\mathfrak{N}(moy, \sigma)$

5.6 LA SYNTHÈSE

Cette section a décrit de manière détaillée la méthodologie *SimAEM* mise en œuvre dans nos travaux. Cette approche se décompose en trois étapes :

- La première étape est la *génération de résidus*. Elle consiste en la comparaison de l'état obtenu par le modèle de référence et celui obtenu par reconstruction du filtre de Kalman étendu (estimation de l'état réel). La reconstruction d'état par le filtre de Kalman étendu consiste à rendre l'erreur d'estimation indépendante des incertitudes du système. Par conséquent, si cette dernière est découplée des incertitudes, il en va de même pour la signature générée à l'étape suivante.
- La seconde étape, l'*évaluation des résidus* analyse les résidus. Elle génère une structure particulière que nous avons définie : la *signature de défauts relative normée*. Cette signature sera appelée par la suite *signature de défauts instantanée*. Contrairement aux signatures que l'on trouve généralement dans la littérature, notre structure n'est pas binaire. Ceci nous permet de quantifier l'importance de la déviation d'une variable en la présence d'un défaut. De plus, en nous affranchissant des notions de grandeurs physiques, nous pouvons comparer les résidus entre eux, et ainsi souligner la déviation observée la plus importante.
- Enfin, la dernière étape consiste en la *localisation de défaut*. Cette étape débute par une recherche hors-ligne de la matrice d'incidence qui regroupe les signatures de défauts théoriques. La structure de cette matrice est générale et applicable à tout point de fonctionnement. Une reconstruction en ligne est alors nécessaire afin de comparer les signatures de défauts théoriques à la signature de défauts instantanée. Pour effectuer cette comparaison, nous avons défini de nouvelles distances : la *distance de Manhattan relative* et la *distance de Manhattan améliorée*. Un défaut existe alors sur le système physique quand la signature de défaut instantanée s'apparente à une signature de défaut théorique. Des indicateurs de défauts sont alors générés et sont transmis à la prochaine étape : l'étape de décision qui consiste en l'identification et la validation de l'occurrence et de la localisation d'un défaut. Notons que l'indicateur de Manhattan améliorée permet de localiser des défauts multiples.

CHAPITRE 6 L'INTÉGRATION DE LA SURVEILLANCE SOUS *PRODHYS*

Après avoir présenté dans le chapitre 5, les concepts généraux de notre approche de surveillance SimAEM et abouti à sa définition, ce chapitre traite de sa mise en œuvre au sein de ProDHYS. Dans ce cadre, le nouveau module ProDHYSaEM a été développé. Ce dernier fournit un ensemble de classes offrant d'une part, une description rigoureuse du cœur de la méthodologie de surveillance adoptée et d'autre part, une description de la simulation d'un procédé avec une défaillance potentielle.

Dans ce contexte, la première section est consacrée à la description des éléments constitutifs de la méthodologie. La deuxième section s'attache plus particulièrement à la simulation d'un système défectueux et à son fonctionnement. Enfin, la dernière partie concerne la modélisation des défauts et décrit en particulier les appareils défectueux.

Une nouvelle couche logicielle a été ajoutée à la structure de la plate-forme *PrODHyS*. Cette couche correspond à un ensemble de classes dédiées à la supervision des procédés. La couche « supervision » s'appuie sur les deux couches « modélisation » et « simulation », et fournit un ensemble d'entités permettant de réaliser des études orientées sur la conduite de procédés. Cette couche comprend :

- Un module dédié à l'ordonnancement, qui couple le module de simulation avec des méthodes d'optimisation stochastique (Thèse de Florian Fabre),
- Le module *PrODHySAEM* (*Process Object Dynamic Hybrid Simulator for Abnormal Event Management*), qui contient un ensemble de classes, chargées de la gestion des études de surveillance des procédés.

Le schéma suivant rappelle la structure générale de la plate-forme et positionne, en particulier, le module *PrODHySAEM* au sein de cette structure :

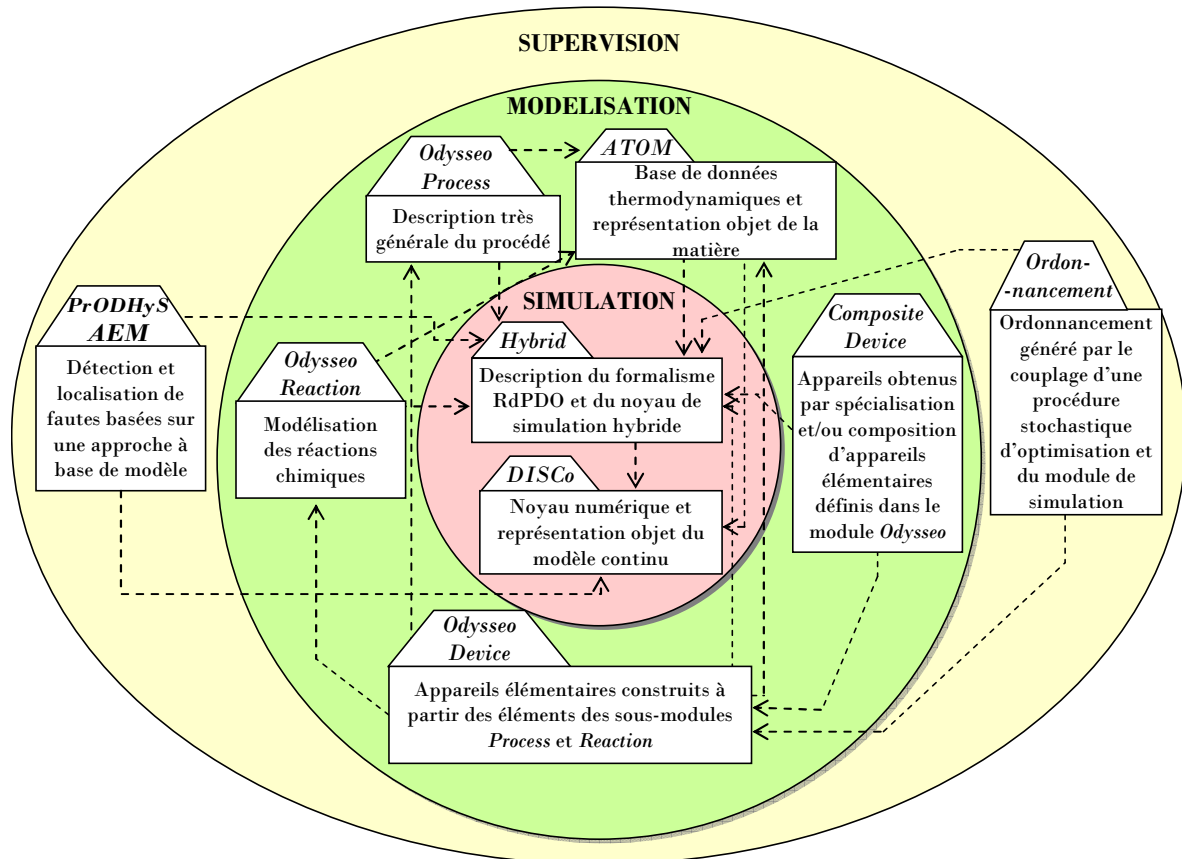


Figure 6.1 La structure de la plate-forme *PrODHyS*

6.1 LES ÉLÉMENTS CONSTITUTIFS DU MODULE *PRODHYSAEM*

Les éléments fondamentaux qui interviennent dans la définition de notre méthode *SimAEM* sont le filtrage, la génération des résidus, la détection, l'élaboration de la table de signature, le diagnostic et la surveillance. Dans le module *PrODHySAEM*, chacun d'eux est décrit par des classes particulières (cf. figure 6.2)

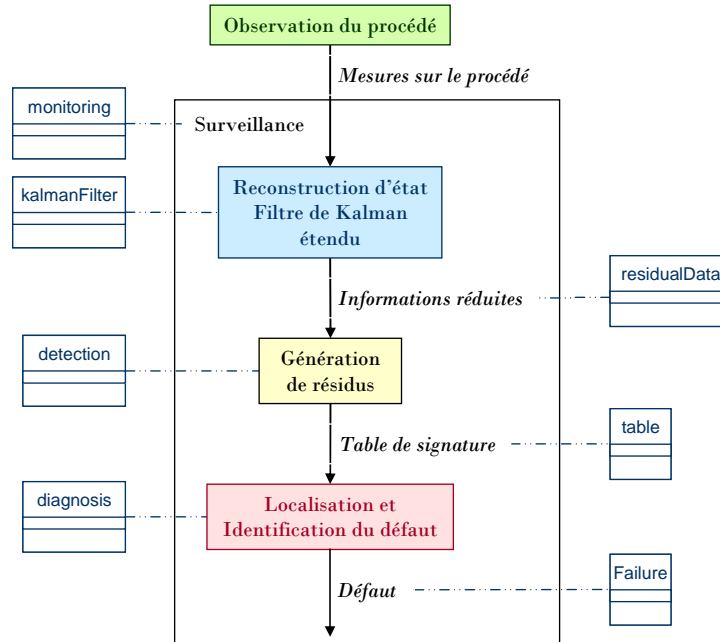


Figure 6.2 Les classes associées aux éléments constitutifs de l'approche *SimAEM*

Dans ce contexte, cette section propose une description détaillée de ces éléments, au moyen de diagrammes UML (cf. annexe A), ainsi que des commentaires sur leur mise en œuvre. Elle met, par ailleurs, l'accent sur les différentes interactions qui les caractérisent.

6.1.1 La reconstruction d'état

La première étape de notre approche de diagnostic consiste en la reconstruction de l'état par l'utilisation du filtre de Kalman étendu (cf. le point 5.2.2). Le diagramme de classes de la figure 6.3 en donne la structure générale.

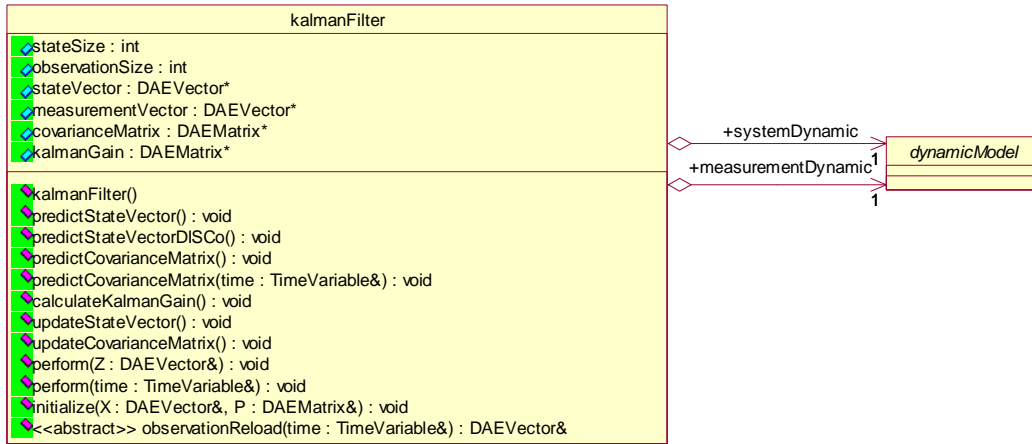


Figure 6.3 Le diagramme de classe du filtre de Kalman

6.1.1.1 Le filtre de Kalman étendu

Un filtre de Kalman étendu est décrit par la classe kalmanFilter. L'instanciation d'un objet kalmanFilter ne nécessite aucun passage en argument. Son initialisation est effectuée par l'appel de la méthode initialize qui requiert en argument deux objets préalablement créés de type DAEVector et DAEMatrix (cf. annexe E).

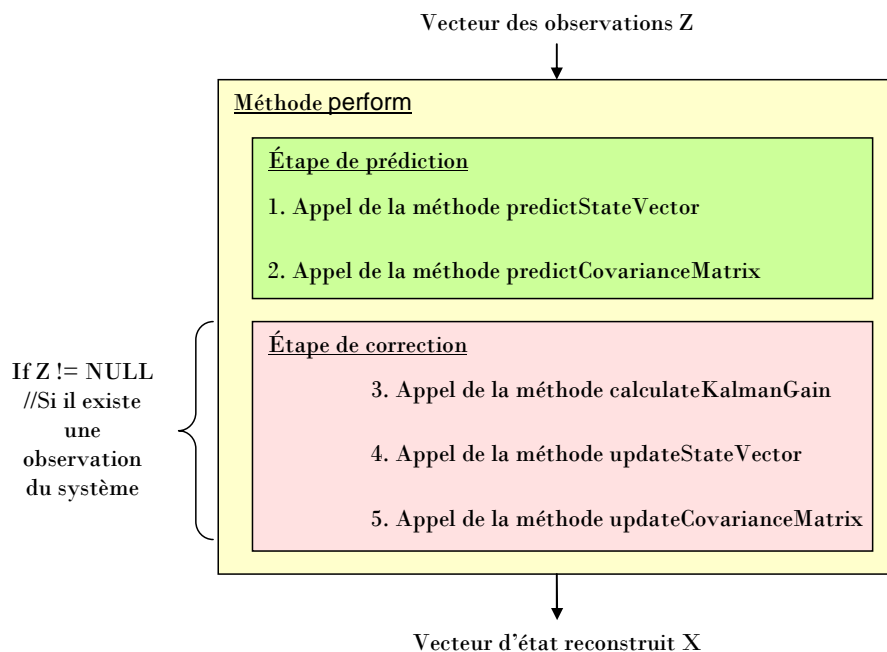


Figure 6.4 Le corps de la méthode perform du filtre de Kalman

La classe kalmanFilter possède deux attributs _systemDynamic et _measurementDynamic de type dynamicModel (classe décrite dans le point suivant). Le premier attribut représente la dynamique du système et le second, la dynamique des observations. La classe dynamicModel permet de prendre en compte tous les types de système : système linéaire ou non. Ainsi, la procédure de calcul établie dans la

classe `kalmanFilter` (procédure présentée dans le point 5.2) est indépendante du système étudié. Son exécution est accomplie au travers de l'appel de l'une des versions de la méthode `perform`. L'une demande en entrée le vecteur des observations, de type `DAEVector`, tandis que l'autre requiert en entrée le temps de type `TimeVariable` (cf. figure 6.4). Cette méthode regroupe les appels des méthodes composant les deux étapes du filtre de Kalman :

- Étape de prédiction :
 - appel de l'une des versions de la méthode `predictStateVector` qui met en œuvre l'équation (4 - 8) ;
 - appel de l'une des versions de la de la méthode `predictCovarianceMatrix` qui met en œuvre l'équation (4 - 9) ;
- Étape de correction :
 - appel de la méthode `calculateKalmanGain` qui met en œuvre l'équation (4 - 10) ;
 - appel de la méthode `updateStateVector` qui met en œuvre l'équation (4 - 11) ;
 - appel de la méthode `updateCovarianceMatrix` qui met en œuvre l'équation (4 - 12).

Le modèle dynamique étant découplé du calcul du filtre de Kalman, la procédure s'applique à tout type de système : modèle du système physique linéaire ou non, modèle caractérisant la dynamique des observations linéaire ou non.

6.1.1.2 Le modèle dynamique

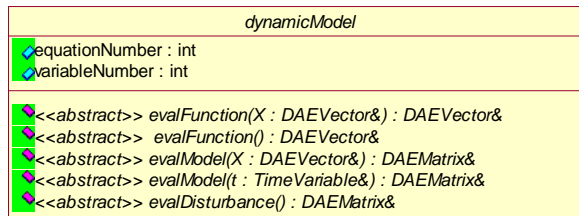


Figure 6.5 Le diagramme de classe du modèle

De façon générale, un modèle dynamique est représenté par la classe `dynamicModel`. Celle-ci décrit un modèle dynamique de la forme simplifiée :

$$\hat{x}_{k+1} = g(\hat{x}_k) + v_k \tag{6 - 1}$$

Avec g linéaire ou non

Où $\hat{x}_k \in \mathfrak{R}^n$ est le vecteur d'état, n correspond au nombre de variables (`_variableNumber`) ;

g est une fonction linéaire ou non qui définit le modèle du système,

$v_k \in \mathfrak{R}^n$ représente les erreurs de modèle.

Un modèle dynamique est donc caractérisé par un nombre d'équations (`_equationNumber`) et un nombre de variables (`_variableNumber`).

Trois méthodes sont définies pour représenter un modèle :

- La méthode `evalModel` retourne un objet de type `DAEMatrix` G , qui dans le cas linéaire correspond à la fonction de transfert telle que $g(x_k) = G \cdot x_k$ et dans le cas non linéaire : $G = \left. \frac{\partial g(x)}{\partial x} \right|_{x=x_k}$.
- La méthode `evalFunction` évalue le modèle au point x_k de type `DAEVector` (donc retourne $g(x_k)$) ;
- La méthode `evalDisturbance` permet d'estimer l'erreur du modèle v_k , c'est-à-dire dans le cas du modèle du système physique, le bruit d'état et dans le cas du modèle de la dynamique des observations, le bruit des mesures.

Cette classe est abstraite et des classes spécialisées seront dérivées pour représenter les modèles du système physique étudié. L'appel au modèle construit par le module *DISCO* est effectué dans les classes spécialisées.

6.1.2 La détection

L'étape suivante consiste en la génération des résidus et des signatures : c'est la *détection*.

6.1.2.1 La génération de résidus et de signatures

La détection est représentée par la classe `detection` (cf. figure 6.6). Elle est dotée de trois tables `_signature`, `_residual` et `_epsilon` qui représentent respectivement la fenêtre d'observation des signatures, des résidus et des seuils de détection. Ces trois tables sont construites grâce aux méthodes respectives, `signatureGeneration`, `residualGeneration` et `epsilonUpdate`. La méthode `perform` regroupe les trois appels des méthodes précédentes. C'est par l'appel de la méthode `perform` qu'on lance la détection.

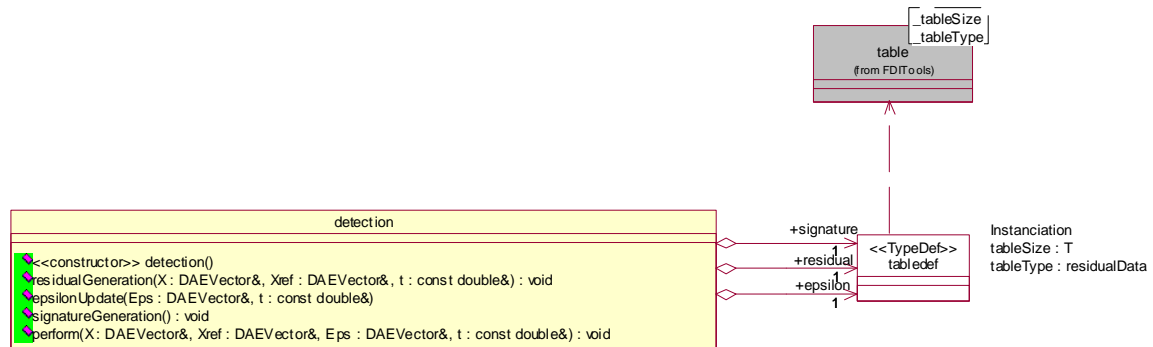


Figure 6.6 Le diagramme de classe de la détection

6.1.2.2 La structure de données

6.1.2.2.1 Les résidus

Dans la méthodologie *SimAEM*, un résidu, un seuil de détection ou encore une signature sont toutes des entités particulières porteuses d'information. De façon générale, ces dernières sont représentées par la même classe `residualData`. Cette classe hérite de la classe `residual` et la spécialisation consiste en l'ajout d'un nouvel attribut `time` de type `double` qui représente le temps en cours. Ainsi, la classe

residualData est composée de trois attributs : le temps et deux attributs hérités, la taille du vecteur d'état `_stateSize` et les données proprement dites qui sont stockées dans un vecteur de type `DAEVector`. L'instanciation d'un objet de type `residualData` nécessite le passage en argument du constructeur de deux objets préalablement créés de type `DAEVector` et `double`.

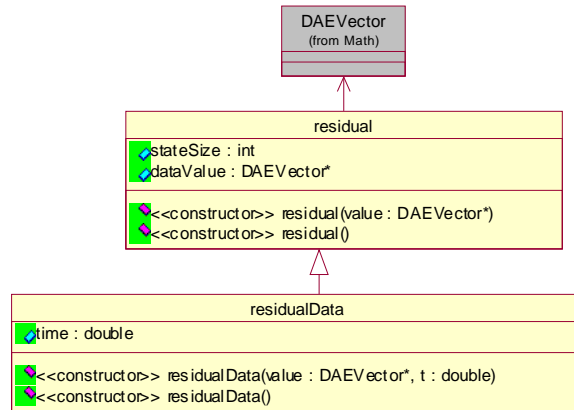


Figure 6.7 Le diagramme de classe des résidus

6.1.2.2.2 Les tables

La classe `table` a été définie pour décrire une structure de type matrice. C'est une classe générique. Les éléments caractérisant complètement la table sont sa dimension `_tableSize` et le type de données `_tableType` dont elle est composée. Ces deux informations sont fournies à l'objet `table` sous forme de paramètres génériques. Le but est de définir une classe indépendante du type d'objet manipulé qui implémente des mécanismes de base réutilisables par un grand nombre. Lorsqu'un objet est instancié, le type qu'il faut utiliser pour cette instance est indiqué. Ceci permet de créer différents types de table. Le diagramme de la classe générique `table` est donné par la figure 6.8.

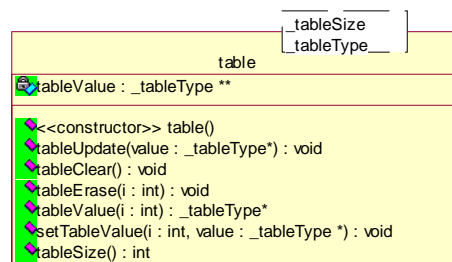


Figure 6.8 Le diagramme de classe de la table

Outre les deux paramètres cités précédemment, la classe `table` possède un attribut supplémentaire qui pointe sur la structure de données contenant les objets de type `_tableType` et de dimension `_tableSize`. Toutefois, le type de résidu n'est pas encore connu au niveau de cette classe.

Dans ce contexte, la structure de données dynamique adoptée est présentée sur la figure suivante :

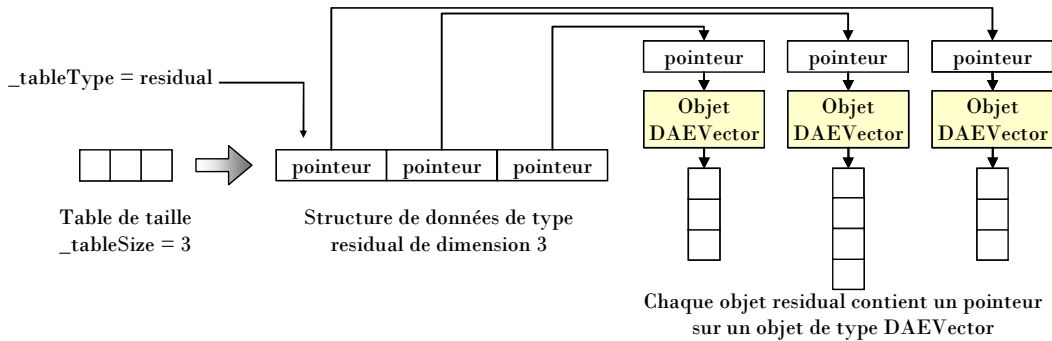


Figure 6.9 La structure de données

La méthode nommée `tableUpdate` permet d'ajouter un élément à la table en décalant la fenêtre d'observation. Son principe est illustré sur la figure 6.10 où T est égal à `_tableSize` :

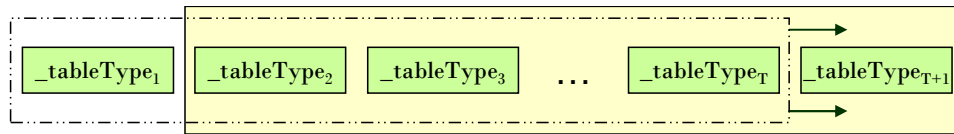


Figure 6.10 Le principe de la méthode `tableUpdate`

La méthode `tableErase` permet d'effacer un élément de la table, tandis que la méthode `tableClear` efface tous les éléments de la table.

6.1.3 Le diagnostic

Le diagnostic va fournir les causes de la défaillance détectée à partir de la signature instantanée générée par l'étape de détection. Il s'effectue à partir des faits mesurés sur le processus (*Signature de défaut instantanée*) et des faits modélisés ou obtenus par expérience (*Signatures de défaut théoriques*).

6.1.3.1 La signature de défaut théorique

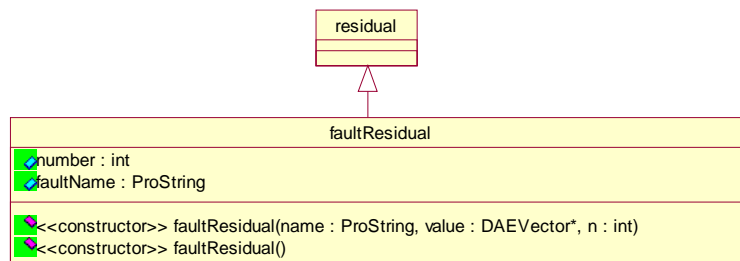


Figure 6.11 Le diagramme de classe de la signature de défaut

La modélisation d'une signature de défaut se fait au travers de la classe `faultResidual`, qui hérite de la classe `residual`. La spécialisation consiste en l'ajout de deux nouveaux attributs : un entier `_number`

qui représente la taille de la population et une chaîne de caractère `_faultName` qui fournit le nom du défaut caractérisant la signature de défaut théorique. Cette classe n'est qu'un conteneur d'informations.

6.1.3.2 La localisation et l'identification de défauts

La dernière étape de l'approche de surveillance *SimAEM* consiste en la localisation et l'identification de défauts. Le diagramme de classe du module de diagnostic est fourni par la figure 6.12.

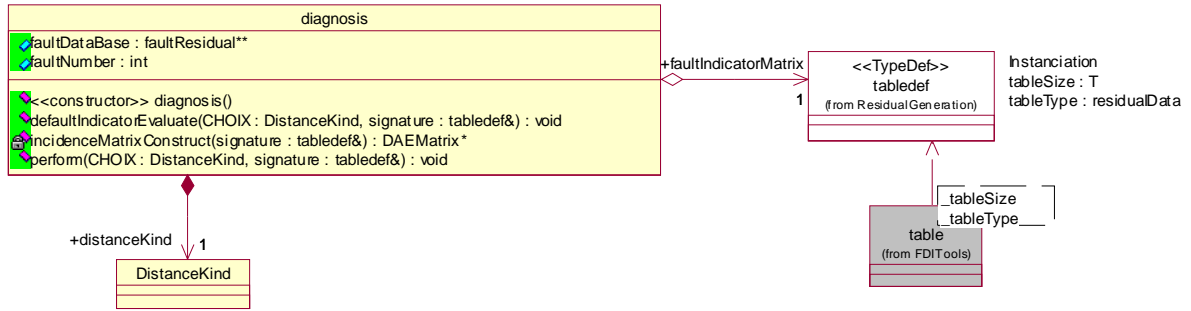


Figure 6.12 Le diagramme de classe du diagnostic

Le diagnostic se base sur la matrice d'incidence. C'est au travers de l'appel de la méthode `incidenceMatrixConstruct` et à partir de la base de données des défauts (`_faultDataBase`) que cette matrice est construite.

La méthode `defaultIndicatorEvaluate` permet de générer les indicateurs de défauts et de mettre à jour la table `_faultIndicatorMatrix`. Le calcul des indicateurs consiste en l'évaluation de la distance entre les signatures de défauts théoriques (matrice d'incidence) et la signature instantanée obtenue durant l'étape de détection.

La méthode `perform` regroupe les appels des deux méthodes `incidenceMatrixConstruct` et `defaultIndicatorEvaluate`. Cette méthode est celle qui assure l'exécution de l'étape de diagnostic.

6.1.3.3 La distance

La classe utilitaire `distance` (cf. figure 6.13) regroupe toutes les méthodes de calcul de distance. Le choix d'une de ces méthodes se fait au travers de l'argument `CHOIX` de type `DistanceKind` dans l'appel de la méthode `perform` de la classe `diagnosis`. Actuellement, seules sont développées les distances de Manhattan relative (cf. définition 5-10) et de Manhattan améliorée (cf. définition 5-14).

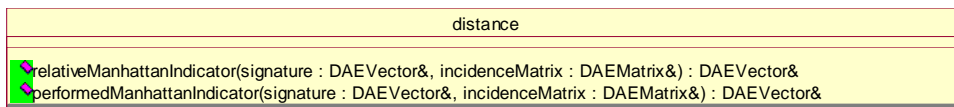


Figure 6.13 Le diagramme de classe de la distance

6.1.4 La surveillance

Le module de surveillance est représenté par la classe `monitoring` (cf. figure 6.14). Celle-ci est dotée d'un ensemble d'attributs et de méthodes qui permettent de mettre en œuvre cette fonction. La

surveillance se déroule en trois étapes principales : la création du module de surveillance, son initialisation et son exécution. Ceci est réalisé par les appels respectivement du constructeur, de la méthode initialize et de la méthode perform.

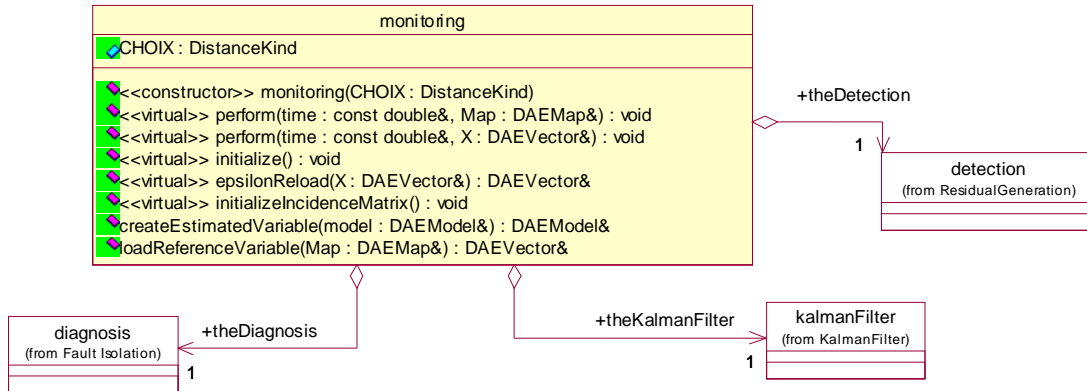


Figure 6.14 Le diagramme de classe de la surveillance

6.1.4.1 La création du module de surveillance

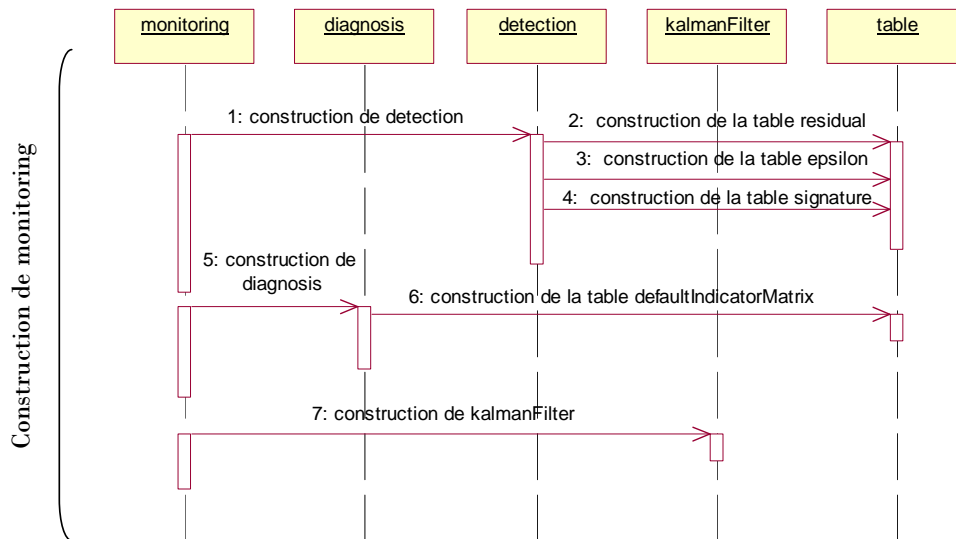


Figure 6.15 Le diagramme de séquence de la construction de monitoring

L'appel du constructeur de la classe monitoring induit l'appel du constructeur des classes detection, diagnosis et kalmanFilter. La figure 6.15 représente le diagramme de séquence de la création d'un objet de type monitoring.

6.1.4.2 L'initialisation du module de surveillance

L'initialisation du module de surveillance se fait par appel de la méthode initialize. Son principe est illustré sur la figure 6.16.

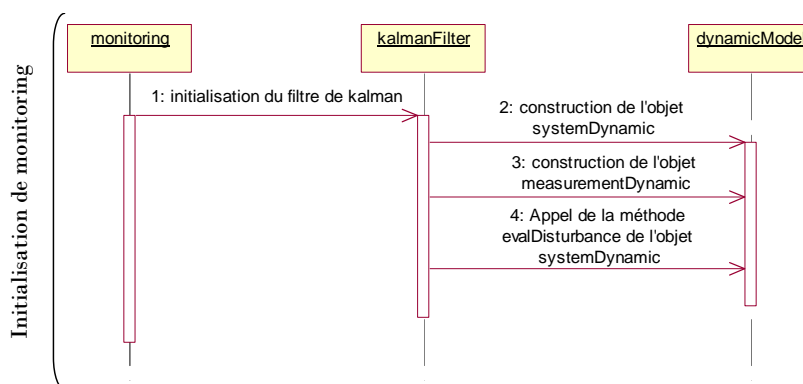


Figure 6.16 Le diagramme de séquence de l'initialisation de monitoring

6.1.4.3 L'exécution du module de surveillance

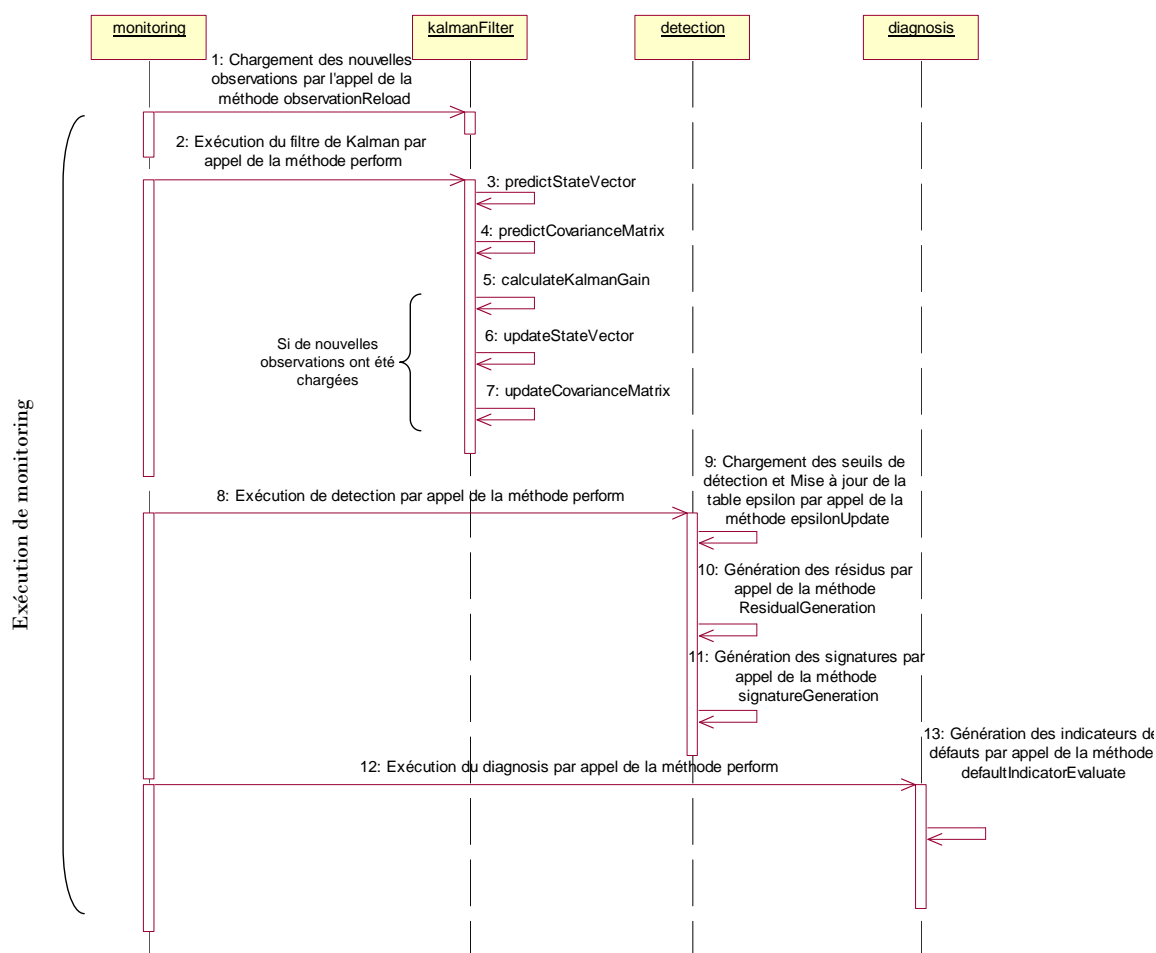


Figure 6.17 Le diagramme de séquence de l'exécution de monitoring

La figure 6.17 représente le séquençement d'appels nécessaires à l'exécution du module de surveillance. La méthode perform de la classe monitoring regroupe l'appel des méthodes perform de ces objets : `_kalmanFilter`, `_detection` et `_diagnosis`.

À ce niveau très général, les éléments constitutifs du module de surveillance (`_detection`, `_diagnosis` et `_kalmanFilter`) sont décrits par des classes de haut niveau qui peuvent être amenées à être spécialisées dans les classes de modules de surveillance héritières.

6.2 LE MODÈLE DE SIMULATION DU PROCÉDÉ DÉFECTUEUX

6.2.1 Les anomalies prises en compte

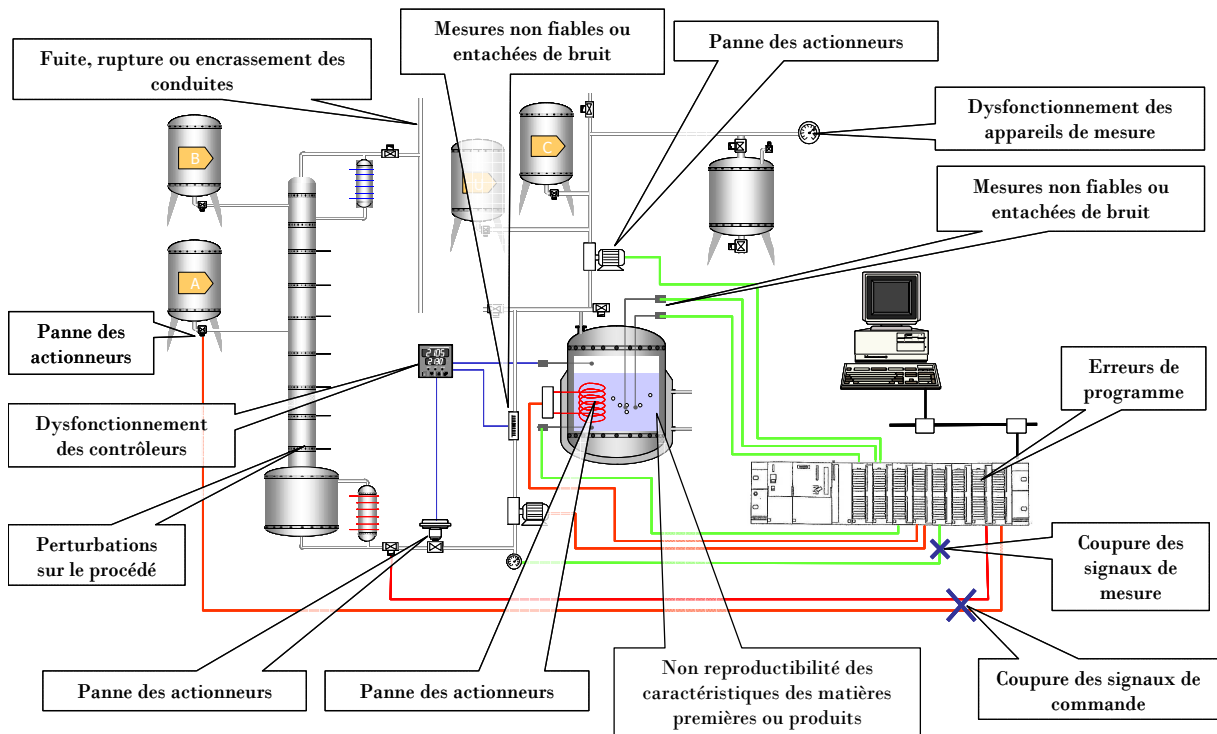


Figure 6.18 Les sources de défaillances

Le module de surveillance doit être capable de traiter aussi bien les défauts du système physique (fuites, pertes thermiques, etc.) que ceux des appareils de contrôle/commande (actionneurs, capteurs, etc.). Dans une approche à base de modèles, l'un des premiers problèmes est de pouvoir différencier les déviations dues à une défaillance de celles liées aux perturbations inhérentes au fonctionnement du procédé. Dans ce cadre, nous avons considéré les perturbations inhérentes suivantes (cf. figure 6.18) :

- les bruits : les bruits de mesure...
- les variations des conditions opératoires du système : perturbations sur le procédé, température initiale faussée...
- les incertitudes paramétriques : incertitudes sur les coefficients d'échange

Nous avons souligné que cet ensemble d’anomalie était inhérent au fonctionnement du processus. Ce ne sont donc pas ces défauts que l’on cherche à diagnostiquer, et afin d’éviter la levée de fausses alarmes, nous avons mis en place des outils permettant de masquer ces anomalies. Dans notre travail, nous considérons ainsi les défaillances que l’on souhaite détecter suivantes (cf. figure 6.18) :

- les défaillances des capteurs et actionneurs : dérive d’un capteur, ou d’un actionneur, défaillance de la valeur seuil de sortie...
- les variations structurelles : usures mécaniques des pièces, défaillance d’un régulateur, fuite d’une conduite...

6.2.2 Le principe

L’outil étant en phase de prototypage, il n’est évidemment pas testé sur site avec un procédé réel. Pour cette raison, le procédé surveillé est actuellement simulé au moyen de *PrODHyS*. Le modèle de simulation associé au procédé « réel » (avec potentiellement des défauts) est donc simulé en parallèle de celui du procédé de « référence » (sans défaut).

Le procédé idéal (sans défauts) et le procédé surveillé (défauts probables) ont la même recette : la partie commande ne varie pas. Néanmoins, des erreurs de paramètres entre les modèles des procédés idéal et réel ont été introduites afin de tester la robustesse. La défaillance, quant à elle, apparaît dans la modélisation des appareils.

6.2.2.1 La modélisation d’un défaut

Un défaut est modélisé par le diagramme de classe représenté sur la figure 6.19. Chaque défaut est défini par sa date d’occurrence `_Time` et sa durée `_Duration`. De même, son état (s’il est actif ou pas) est retranscrit par l’attribut `_Active`. Cet attribut n’est pas figé et l’utilisateur peut le modifier en utilisant la méthode `active` pour activer le défaut et la méthode `desactive` pour désactiver le défaut. Enfin, chaque défaut possède une référence sur l’appareil défaillant `_FaultyDevice`.

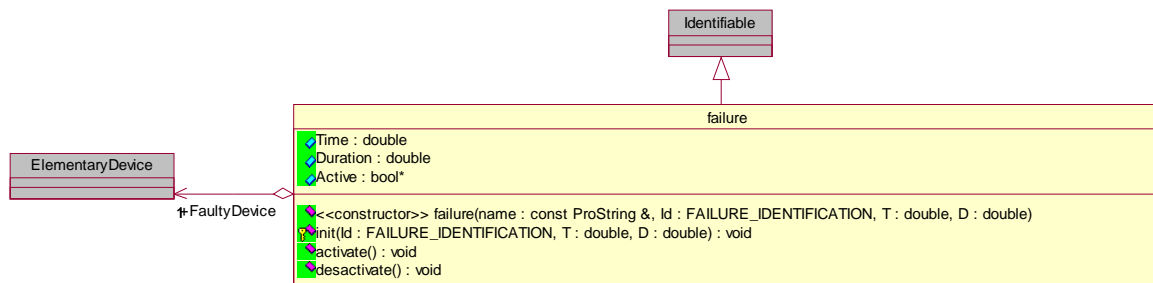


Figure 6.19 Le diagramme de classe d’un défaut

6.2.2.2 La modélisation des appareils

Afin de modéliser un appareil défaillant, le modèle de simulation est construit à partir d’objets `ElementaryDevice` spécialisés dans lesquels les états de défaillance sont définis de manière intrinsèque et

explicite. Ceci conduit alors à ajouter des places et des transitions dans le modèle de l'appareil « idéal » correspondant. Dans le point suivant, la modélisation d'une partie des appareils défectueux est présentée.

6.2.3 La simulation du procédé défectueux

La simulation du procédé « réel » nécessite ainsi la gestion des défaillances potentielles de chaque appareil le constituant.

6.2.3.1 La génération de défaut

En phase de test, les défauts sont évidemment générés de manière aléatoire. Par contre, en phase d'analyse du système, ils sont générés selon un calendrier fixé. Rappelons que chaque défaut est défini par un identificateur, sa date d'occurrence et sa durée. Un réseau de Petri est ensuite construit de manière automatique à partir des informations définissant chaque instance de défaut.

La figure 6.20 montre le RdP correspondant à la génération de trois défauts D1, D2 et D3, les places différentielles modélisant ici des places temporisées.

Lors du franchissement des transitions, les défauts sont activés ou désactivés via une action (appel des méthodes *active* et *desactive* de la classe *failure*). La figure 6.20 montre par exemple la gestion des défauts D1, D2 et D3.

Par ailleurs, cette représentation rend possible l'apparition de plusieurs défauts simultanément. L'apparition de défauts simultanés fera l'objet d'une étude dans le chapitre 8.

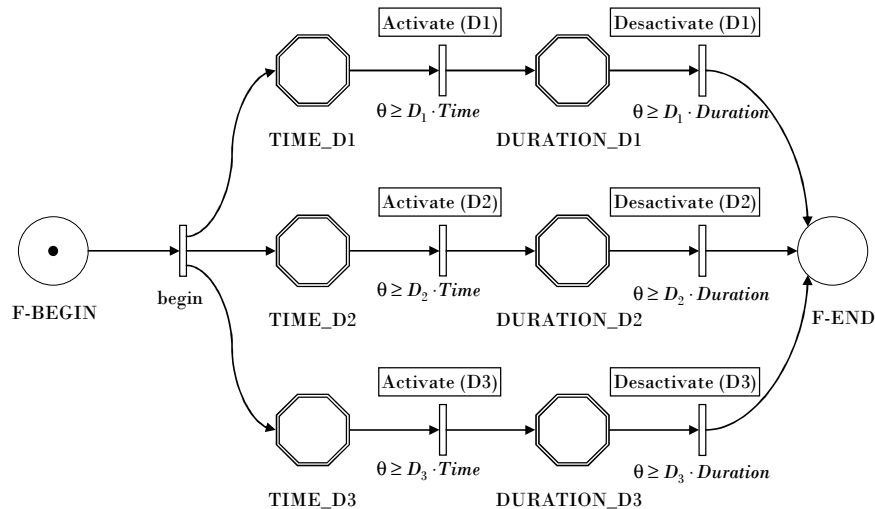


Figure 6.20 Le réseau de Petri associé au générateur de défaut

6.2.3.2 La simulation en parallèle

Précisons que le procédé « idéal » et le procédé surveillé sont pilotés grâce à la même recette. Toutefois, deux instances de cette recette sont créées de manière à pouvoir piloter de manière indépendante ces deux procédés. De même, le modèle sur lequel s'appuie le filtre de Kalman étendu est obtenu par le jeu du réseau de Petri de cette même recette. Ce modèle est identique à celui du procédé de référence mais bien

sûr différent de celui du procédé réel. Ainsi, nous effectuons également une copie de cette recette pour l'exécution du filtrage.

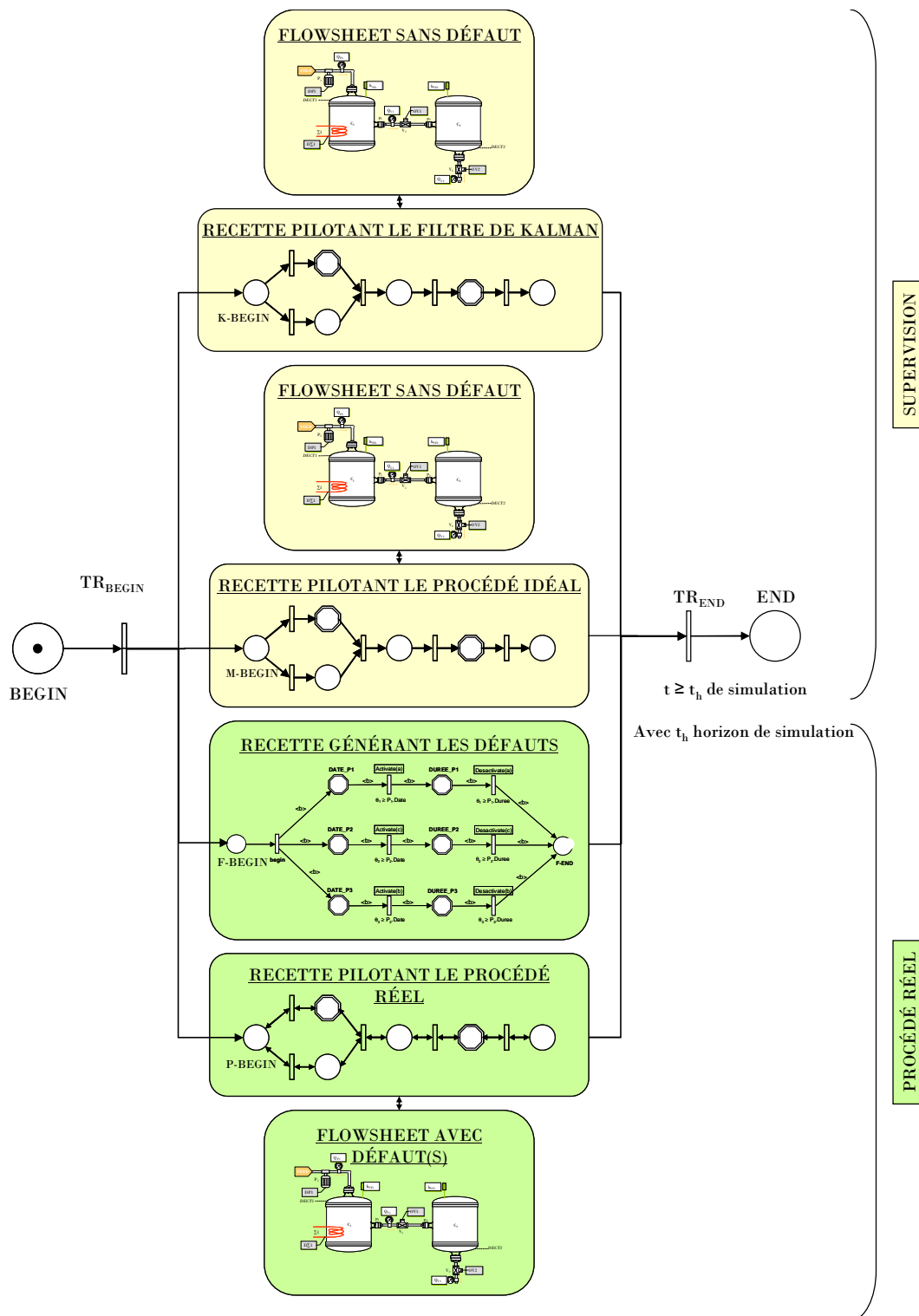


Figure 6.21 Le réseau de Petri de la recette générale

Une recette générale est alors créée et regroupe les trois instances de recette (recette du procédé idéal, recette du procédé réel, et recette du filtre de Kalman) et la recette du générateur de défauts. Afin de différencier les différentes places et transitions de ces réseaux de Petri, un préfixe est ajouté à leurs noms :

- un « M- » pour la recette du procédé de référence,
- un « K- » pour la recette utilisée par le filtre de Kalman,
- un « F- » pour la recette du générateur de défaut,
- un « P - » pour la recette du procédé réel.

La figure 6.21 illustre le concept général. Le tir de la transition TR_{BEGIN} entraîne le marquage de la place $BEGIN$ des trois réseaux de Petri suivants : le réseau de Petri de la recette pilotant le procédé réel, celui pilotant le procédé de référence, celui utilisé par le filtre de Kalman pour la reconstruction d'état et enfin la recette générant les défauts. Rappelons que la recette du procédé réel, celle du procédé de référence et celle du filtre de Kalman sont identiques, mais que les modèles associés à ces deux procédés sont différents.

Le filtre de Kalman utilise le même flowsheet que le procédé de référence. De même, le procédé réel possède le même flowsheet que le procédé de référence mais est constitué d'appareils potentiellement défectueux.

La place de fin de la recette générale est marquée, lorsque le temps est supérieur ou égal au temps de l'horizon de simulation, en d'autres termes lorsque la simulation est terminée.

Notons que lorsque le module sera utilisé sur un cas réel, seule la recette du procédé réel restera de la partie PROCÉDÉ RÉEL de figure 6.21.

6.3 LA MODÉLISATION DES APPAREILS

Le modèle de simulation de l'appareil défectueux est construit à partir d'objets `ElementaryDevice` spécialisés dans lesquels les états de défaillance sont définis de manière intrinsèque et explicite. Ceci conduit alors à ajouter des places et des transitions dans le modèle de l'appareil « idéal » correspondant.

La section 2.2 a montré que l'objectif de *PrODHyS* est de fournir des éléments généraux qui permettent de construire des entités spécifiques plus complexes en exploitant les mécanismes de dérivation propres à l'approche objet. Ces principes sont largement utilisés dans la description de la topologie et de la configuration des appareils défaillants. La section suivante propose une description de l'intégration de ces concepts au sein du formalisme RdPDO.

Nous ne présentons ici que la modélisation des appareils suivants : les capteurs, les détecteurs, les alimentations matières et les vannes.

6.3.1 Les capteurs

Un *capteur* est un appareil qui permet de suivre l'évolution d'une grandeur physique. Plusieurs types de capteurs ont été introduits : capteur de Débit (FlowCaptor), de Température (TempCaptor), de Niveau (LevelCaptor), de Composition (CompCaptor) et de Réention (HoldupCaptor). Le diagramme de classes de la figure 6.22 propose une synthèse des différents capteurs mis en œuvre dans *PRODHyS*. La liste établie n'est pas restrictive et peut être complétée.

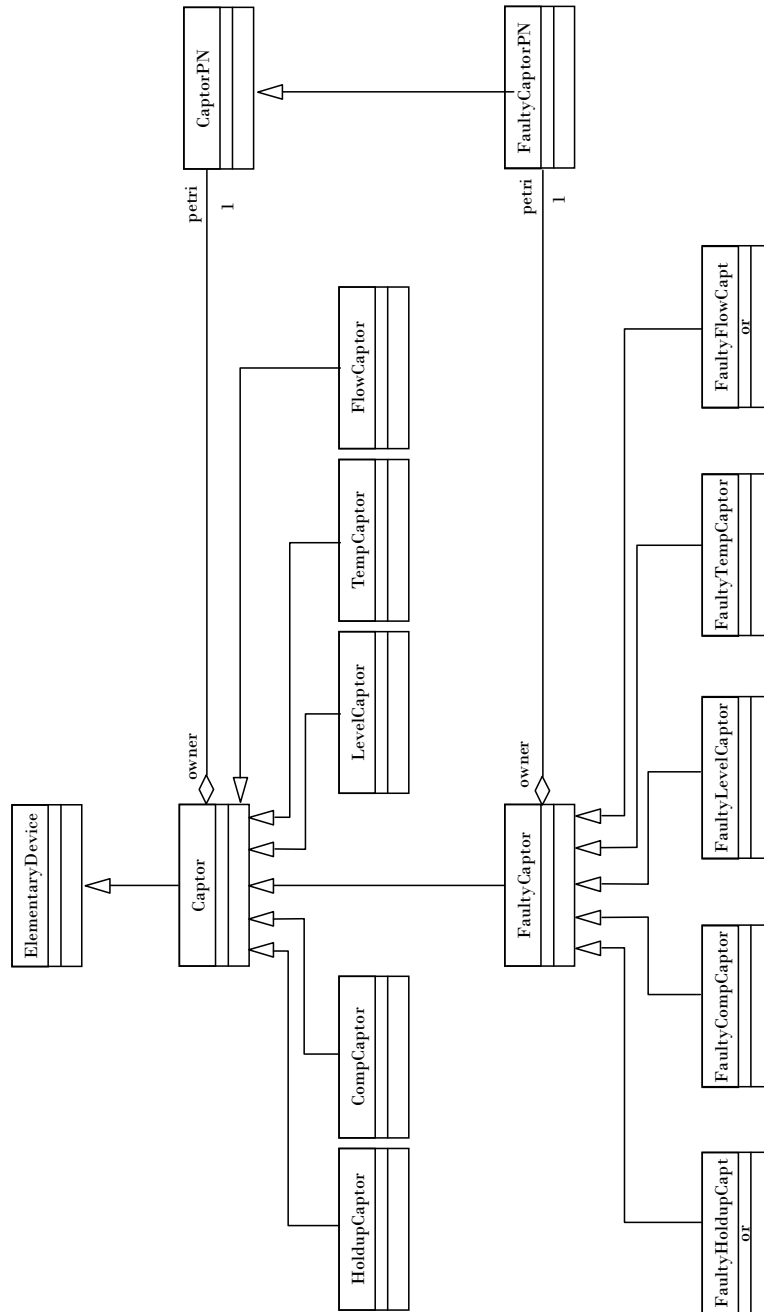


Figure 6.22 Les capteurs spécialisés

6.3.1.1 Le capteur sans défaut

Le comportement d'un détecteur est décrit par la classe `CaptorPN` (cf. figure 6.23) qui hérite de celui de l'appareil élémentaire.

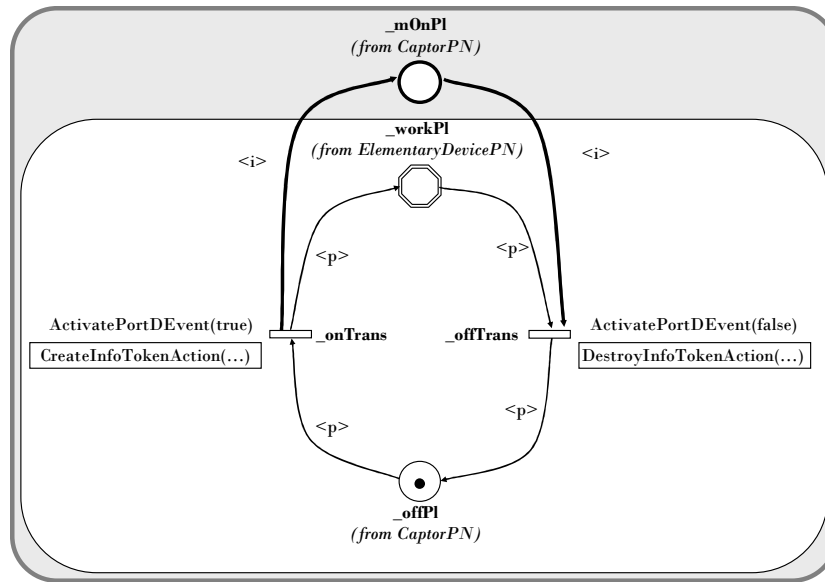


Figure 6.23 Le réseau de Petri du capteur non défaillant

Les conditions et actions associées au réseau de Petri de l'alimentation sont résumées dans le tableau suivant :

| Transition | Événement | Action |
|------------------|--|---|
| _onTrans | Teste si le port d'entrée de communication du capteur est actif. Celui-ci est actif si le port de sortie auquel il est connecté est actif. Ce dernier est actif si la grandeur captée existe. Si l'événement apparaît, alors le capteur passe dans l'état actif (place <code>_workPI</code>) | Création du jeton d'information portant un pointeur sur la grandeur mesurée via le port de communication |
| _offTrans | Teste si le port d'entrée de communication du capteur est inactif. Celui-ci est inactif si le port de sortie auquel il est connecté est inactif. Ce dernier est inactif si la grandeur captée n'existe pas ou plus. Si l'événement apparaît, alors le capteur passe dans l'état inactif (place <code>_offPI</code>) | Destruction du jeton d'information portant le pointeur sur la grandeur mesurée via le port de communication |

Tableau 6.1 Les conditions et actions associées aux transitions du réseau de Petri du capteur sans défaut

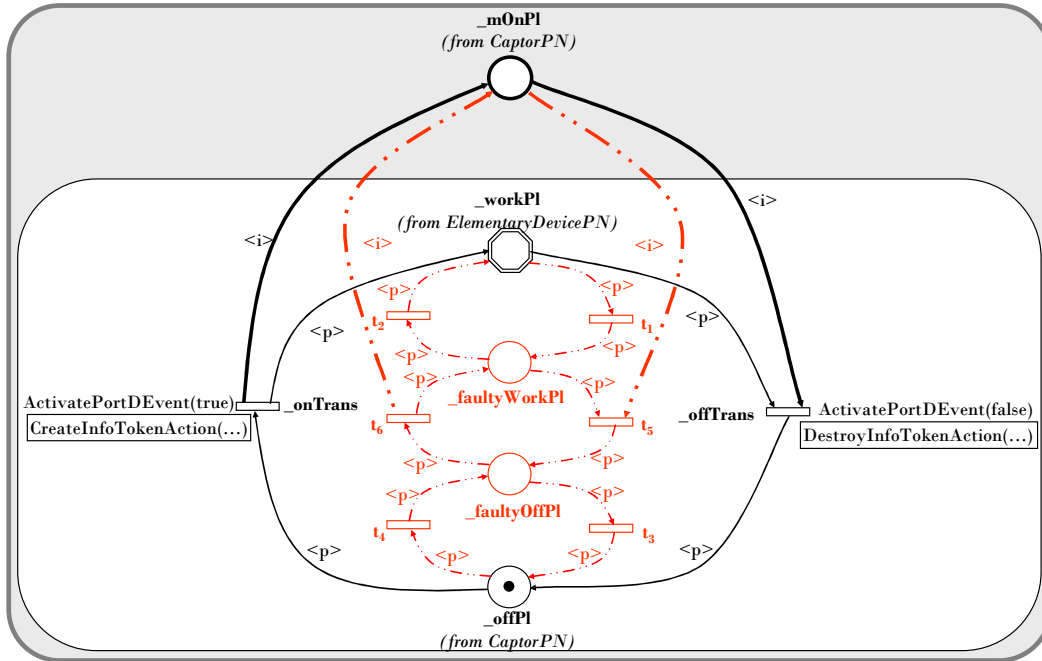


Figure 6.24 Le réseau de Petri du capteur défaillant

| Transition | Evénement | Action |
|---|--|---|
| _onTrans | Teste si le port d'entrée de communication du capteur est actif. Celui-ci est actif si le port de sortie auquel il est connecté est actif. Ce dernier est actif si la grandeur captée existe. Si l'événement apparaît, alors le capteur passe dans l'état actif (place _workPI) | Création du jeton d'information portant un pointeur sur la grandeur mesurée via le port de communication |
| _offTrans | Teste si le port d'entrée de communication du capteur est inactif. Celui-ci est inactif si le port de sortie auquel il est connecté est inactif. Ce dernier est inactif si la grandeur captée n'existe pas ou plus. Si l'événement apparaît, alors le capteur passe dans l'état inactif (place _offPI) | Destruction du jeton d'information portant le pointeur sur la grandeur mesurée via le port de communication |
| t₁ ↔ _workToFaultyWorkTrans | SetToFaultyBehaviourFailure == true Apparition du défaut. | |
| t₂ ↔ _faultyWorkToWorkTrans | SetToFaultyBehaviourFailure == false Disparition du défaut et retour dans l'état de marche normal _workPI | |
| t₃ ↔ _faultyOffToOffTrans | SetToFaultyBehaviourFailure == false Disparition du défaut et retour dans l'état normal inactif _offPI | |
| t₄ ↔ _offToFaultyOffTrans | SetToFaultyBehaviourFailure == true Apparition du défaut. | |
| t₅ ↔ _faultyWorkToFaultyOffTrans | Teste si le port d'entrée de communication du capteur est inactif. | Destruction du jeton d'information portant le pointeur sur la grandeur mesurée via le port de communication |
| t₆ ↔ _faultyOffToFaultyWorkTrans | Teste si le port d'entrée de communication du capteur est actif. | Création du jeton d'information portant un pointeur sur la grandeur mesurée via le port de communication |

Tableau 6.2 Les conditions et actions associées aux transitions du réseau de Petri du capteur avec défaut

6.3.1.2 Le capteur avec défaut

Les défauts des capteurs modélisés sont les suivants :

- Quel que soit son état, le capteur fournit une mesure bruitée. Le défaut est alors un défaut additif mais que nous considérons assimilable.
- Le capteur fournit une mesure erronée :
 - Défaut 1 : le capteur est actif (la grandeur mesurée existe). Le défaut est alors multiplicatif (on considère un taux de dispersion) et est considéré comme caractéristique, voire critique.
 - Défaut 2 : le capteur est inactif (la grandeur mesurée n'existe pas). Une valeur aléatoire est transmise. Le défaut est alors caractéristique, voire critique.

La figure 6.24 représente la modélisation du capteur défectueux. Les éléments en trait continu correspondent au modèle de l'appareil « idéal », tandis que les éléments en trait discontinu sont les éléments qui sont ajoutés pour la modélisation des appareils avec défauts. Les conditions et actions associées au réseau de Petri du capteur sont résumées dans le tableau 6.2.

Par défaut, le signal transmis par le capteur est bruité (défaut additif). De ce fait, lorsque le capteur est dans l'état `_workPI`, le signal transmis au réseau de Petri recette est :

$$\text{Signal} = \text{Grandeur} + \left(1 - \frac{2 * \text{rand}()}{\text{RANDMAX}}\right) \cdot \text{Tolerance} \quad (6 - 2)$$

Avec *Grandeur* : grandeur mesurée (niveau, température, etc.),

Tolerance : écart en absolue fournie dans le fichier d'initialisation,

RANDMAX : le plus grand nombre réel codable positif,

Rand() : fonction qui tire aléatoirement une valeur entre $[0 ; \text{RANDMAX}]$

La figure 6.25 illustre la notion de tolérance et de Grandeur mesurée et bruitée :

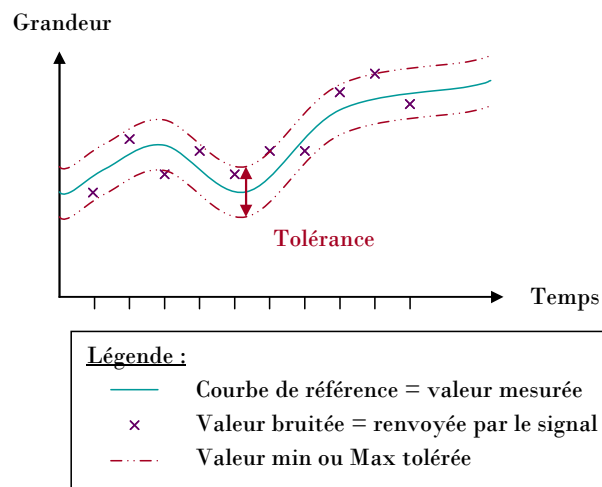


Figure 6.25 Les notions de tolérance et valeur bruitée

Considérons le défaut 1 (état actif du capteur) et son interprétation en termes de modélisation. Quand la transition t_1 est franchie, le signal transmis par le capteur le signal transmis est entaché d'un écart permanent avec la grandeur physique (problème d'étalonnage, par exemple). Cet écart est tiré aléatoirement sur un intervalle caractérisé par un taux de dispersion *TauxErreur* (donné en % dans le fichier initialisation). La valeur du signal est alors :

$$\text{Signal} = \text{Grandeur} \cdot (1 - \text{TauxErreur}) + \frac{2 \cdot \text{rand}() \cdot \text{TauxErreur} \cdot \text{Grandeur}}{\text{RANDMAX}} \quad (6 - 3)$$

Notons que le signal reste bruité.

Considérons maintenant le défaut 2 (état inactif du capteur) et sa représentation. Quand la transition t_4 est franchie, le signal transmis est entaché d'un écart permanent avec la grandeur physique (problème d'étalonnage, par exemple). Cet écart est tiré aléatoirement dans un intervalle $[V_{\min Off}; V_{\max Off}]$ (donné dans le fichier initialisation). En effet, cela correspond à une perturbation du capteur qui transmet un signal sur une grandeur mesurée inexistante (par exemple, une composition vapeur pour une charge liquide).

La valeur du signal est alors :

$$\text{Signal} = \frac{\text{rand}() \cdot (V_{\max Off} - V_{\min Off})}{\text{RANDMAX}} + V_{\min Off} \quad (6 - 4)$$

6.3.2 Les détecteurs

Tout comme le capteur, le détecteur a été conçu pour suivre une grandeur physique du procédé. Cependant, cette grandeur n'est pas transmise en tant que telle à la recette. Elle est évaluée par rapport à une valeur seuil spécifiée au préalable par le niveau commande. Le détecteur peut ainsi transmettre le résultat de cette évaluation (qui se traduit par une expression logique) en émettant un signal binaire sur sa place signal. La communication entre la recette et le procédé se fait alors uniquement en termes de messages binaires. La figure 6.26 illustre le principe du détecteur :

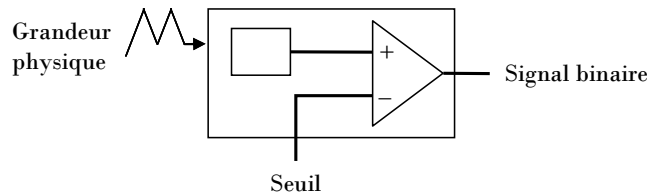


Figure 6.26 Le principe d'un détecteur

Le diagramme de classes de la figure 6.27 propose une synthèse des différents détecteurs mis en œuvre dans *PrODHyS*. Ainsi, le détecteur est défini comme une spécialisation du capteur et son comportement est décrit par la classe *DetectorPN* qui hérite du réseau *CaptorPN*.

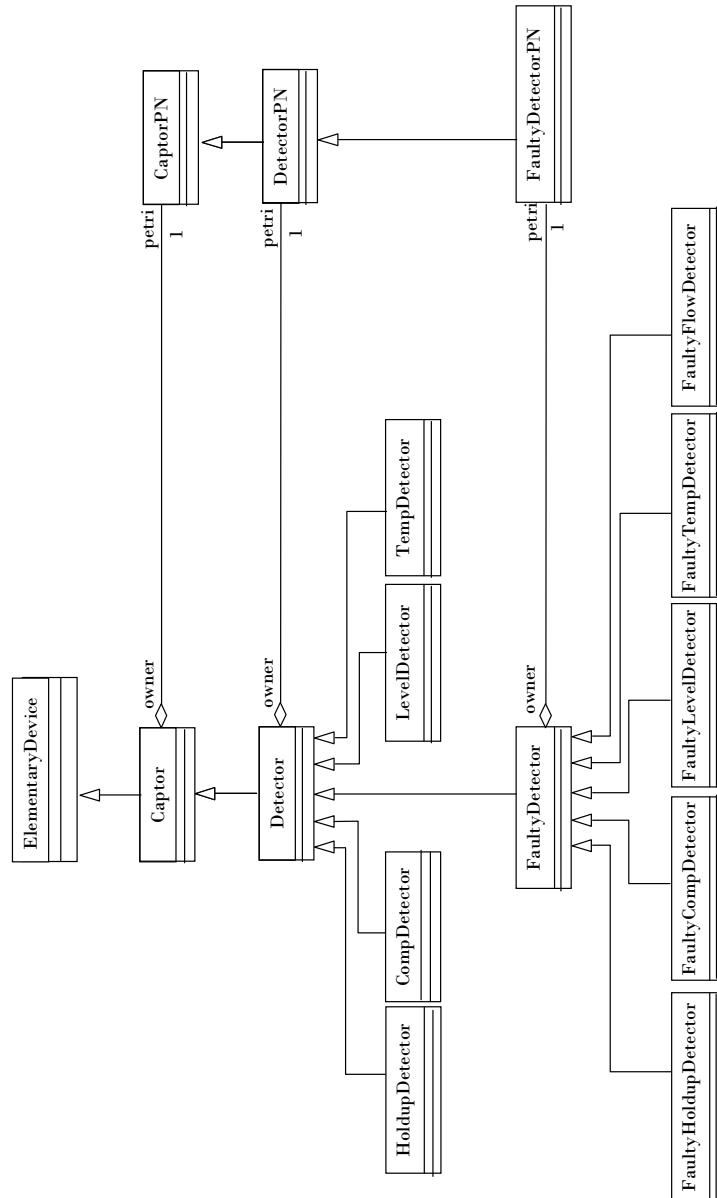


Figure 6.27 Les détecteurs spécialisés

6.3.2.1 Le détecteur sans défaut

Le détecteur sans défaut a été présenté au chapitre 2 dans le point 2.6.2.2.2.

6.3.2.2 Le détecteur avec défaut

La modélisation des défaillances du détecteur est plus simple étant donné la nature booléenne du signal qu'il renvoie. Les défauts du détecteur modélisés sont donc les suivants :

- Le détecteur renvoie la valeur TRUE alors qu'il aurait dû renvoyer la valeur FALSE.
- Et inversement, le détecteur renvoie la valeur FALSE alors qu'il aurait dû renvoyer la valeur TRUE.

Notons également que comme pour le capteur, le signal capté par le détecteur est, par défaut, supposé bruité. De ce fait, lorsque le détecteur est dans l'état `_offPI` ou `_workPI`, le signal utilisé pour tester la condition associée au détecteur est donné par l'équation (6 - 2).

La figure 6.28 représente la modélisation du détecteur défectueux. Les éléments en trait continu correspondent au modèle de l'appareil « idéal », tandis que les éléments en trait discontinu sont les éléments qui sont ajoutés pour la modélisation des appareils avec défauts.

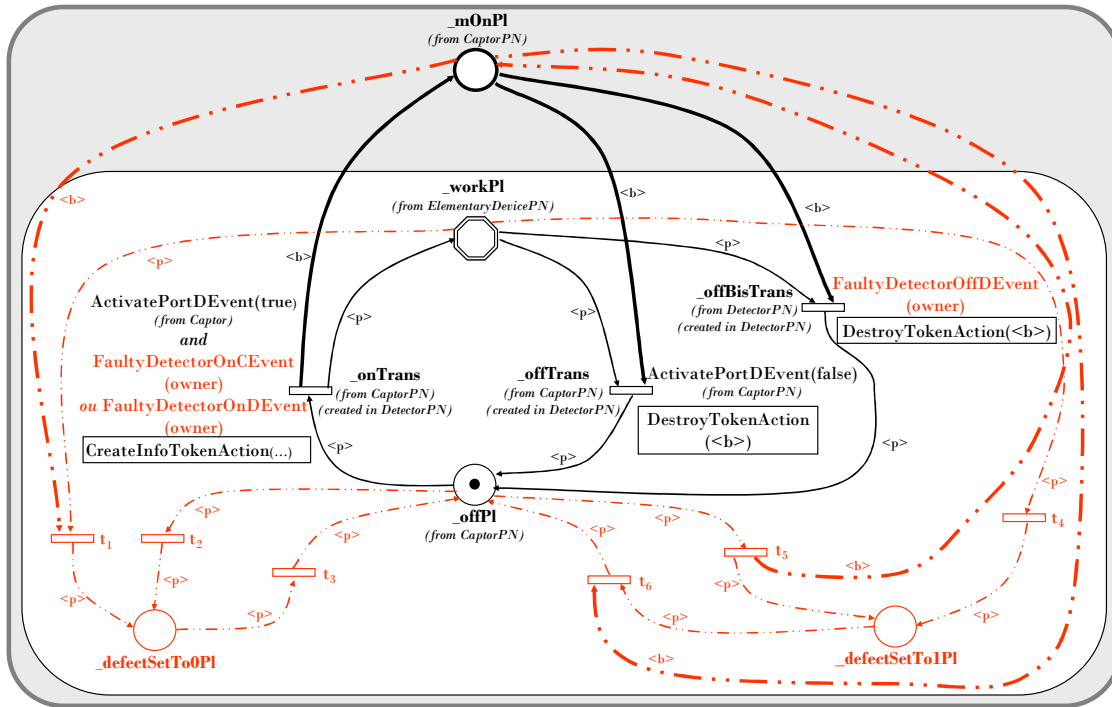


Figure 6.28 Le réseau de Petri du détecteur défaillant

Aux conditions et actions associées aux transitions du détecteur sans défaut (cf. tableau 2.3), s'ajoutent les conditions et actions associées au réseau de Petri du détecteur défectueux. Celles-ci sont résumées dans le tableau 6.3.

Nous considérons que, deux défaillances amènent au défaut qui consiste en l'envoi du message FALSE (ou 0) au lieu du message TRUE (ou 1) :

- Le détecteur passe de l'état ON à l'état OFF sans raison. Cette défaillance est modélisée par le franchissement de la transition t_1 .
- Le détecteur reste bloqué sur l'état OFF. Cette défaillance est modélisée par le franchissement de la transition t_2 .

De même, nous envisageons que, deux défaillances conduisent au défaut qui consiste en l'envoi du message TRUE (ou 1) au lieu du message FALSE (ou 0) :

- Le détecteur passe de l'état OFF à l'état ON sans raison. Cette défaillance est modélisée par le franchissement de la transition t_5 .

- Le détecteur reste bloqué sur l'état ON. Cette défaillance est modélisée par le franchissement de la transition t_4 .

Les transitions t_3 et t_6 permettent de revenir dans un état normal de fonctionnement.

| Transition | Evénement | Action |
|--|---|---|
| t_1 \Leftrightarrow _WorkToDef0Trans | SetTo0Failure == true Apparition du défaut forçant le détecteur à être dans l'état 0 alors qu'il était dans l'état 1 (commutation forcée du détecteur). Ceci se traduit par le non-marquage de la place mOnPl . | Destruction du jeton binaire permettant de marquer la place _mOnPl |
| t_2 \Leftrightarrow _OffToDef0Trans | SetTo0Failure == true Apparition du défaut forçant le détecteur à rester dans l'état 0. Ceci se traduit par le non-marquage de la place mOnPl . | |
| t_3 \Leftrightarrow _Def0ToOffTrans | SetTo0Failure == false Disparition du défaut forçant le détecteur à être dans l'état 0 et retour à l'état 0 en fonctionnement normal (place _offPl). | |
| t_4 \Leftrightarrow _WorkToDef1Trans | SetTo1Failure == true Apparition du défaut forçant le détecteur à rester dans l'état 1. Ceci se traduit par le maintien du marquage de la place mOnPl . | |
| t_5 \Leftrightarrow _OffToDef1Trans | SetTo1Failure == true Apparition du défaut forçant le détecteur à être dans l'état 1 alors qu'il était dans l'état 0 (commutation forcée du détecteur). Ceci se traduit par le marquage de la place mOnPl | Création du jeton binaire permettant de marquer la place _mOnPl |
| t_6 \Leftrightarrow _Def1ToOffTrans | SetTo1Failure == false Disparition du défaut forçant le détecteur à être dans l'état 1 et retour par défaut dans l'état 0 en fonctionnement normal (place _offPl) | Destruction du jeton binaire permettant de marquer la place _mOnPl |

Tableau 6.3 Les conditions et actions associées aux transitions du réseau de Petri du détecteur avec défaut

6.3.3 Les alimentations

Une alimentation représente une source infinie de matière ou d'énergie pour laquelle, la notion de « quantité finie » n'existe pas. Elle possède un seul port de sortie et alimente ainsi un seul appareil à la fois.

L'alimentation appartient à la catégorie des appareils dits actifs. En effet, sa fonctionnalité est de fournir une quantité de matière ou d'énergie en réponse à une commande spécifiée par le niveau commande. Celle-ci se traduit, au niveau du modèle hybride, par le marquage d'une place de commande qui pilote l'arrêt et le démarrage de l'alimentation. Le diagramme de classes de la figure 6.29 présente les différents types d'alimentation mis en œuvre dans *PrODHyS*.

Prenons l'exemple de l'alimentation de matière instantanée. Les alimentations d'énergie sont définies de façon similaire aux alimentations de matière. Elles ne sont par conséquent pas détaillées dans le cadre de ce manuscrit.

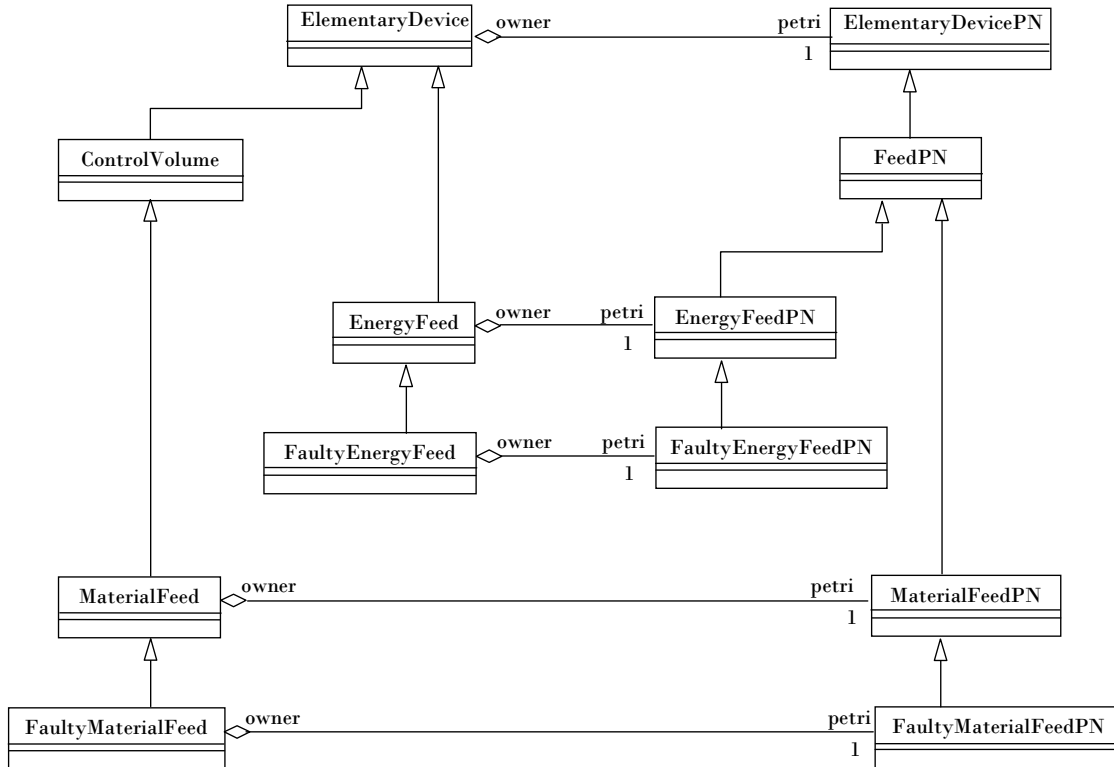


Figure 6.29 Les alimentations spécialisées

6.3.3.1 L'alimentation matière sans défaut

Une *alimentation de matière instantanée* désigne ainsi un appareil délivrant une quantité de matière de façon instantanée. Son comportement est décrit par la classe `MaterialFeedPN` (cf. figure 6.29) qui hérite de la classe `ElementaryDevicePN`. Il est caractérisé par trois états distincts : la place `_closePI` qui correspond à une alimentation fermée, la place `_workPI` qui correspond à une alimentation ouverte et la place `mOpen` qui est la place de commande pilotant l'arrêt et le démarrage de l'alimentation (cf. figure 2.16). Les conditions et actions associées au réseau de Petri de l'alimentation sont résumées dans le tableau 2.2.

6.3.3.2 L'alimentation matière avec défaut

Les défauts de l'alimentation matière modélisés sont les suivants :

- L'alimentation matière fournit un flux alors qu'elle ne devrait pas ;
- L'alimentation matière ne fournit aucun flux alors qu'elle devrait ;
- L'alimentation matière fournit un flux mais à un débit dégradé (soit plus fort soit plus faible).

La figure 6.30 représente la modélisation de l'alimentation matière défectueuse. Les éléments en trait continu correspondent au modèle de l'appareil « idéal », tandis que les éléments en trait discontinu sont les éléments qui sont ajoutés pour la modélisation des appareils avec défauts.

Aux conditions et actions associées aux transitions du détecteur sans défaut (cf. tableau 2.2), s'ajoutent les conditions et actions associées au réseau de Petri de l'alimentation. Celles-ci sont résumées dans le tableau 6.4.

Nous envisageons que deux défaillances conduisent au défaut qui consiste en la fermeture de l'alimentation matière au lieu de son ouverture :

- L'alimentation matière ne fournit aucun débit sans raison. Cette défaillance est modélisée par le franchissement de la transition t_5 .
- L'alimentation matière reste bloquée dans l'état OFF. Cette défaillance est modélisée par le franchissement de la transition t_2 .

De même, nous considérons que deux défaillances amènent au défaut qui consiste à l'ouverture de l'alimentation matière au lieu de sa fermeture :

- L'alimentation matière reste bloquée dans l'état ON. Cette défaillance est modélisée par le franchissement de la transition t_4 .
- L'alimentation matière fournit un flux sans raison. Cette défaillance est modélisée par le franchissement de la transition t_5 .

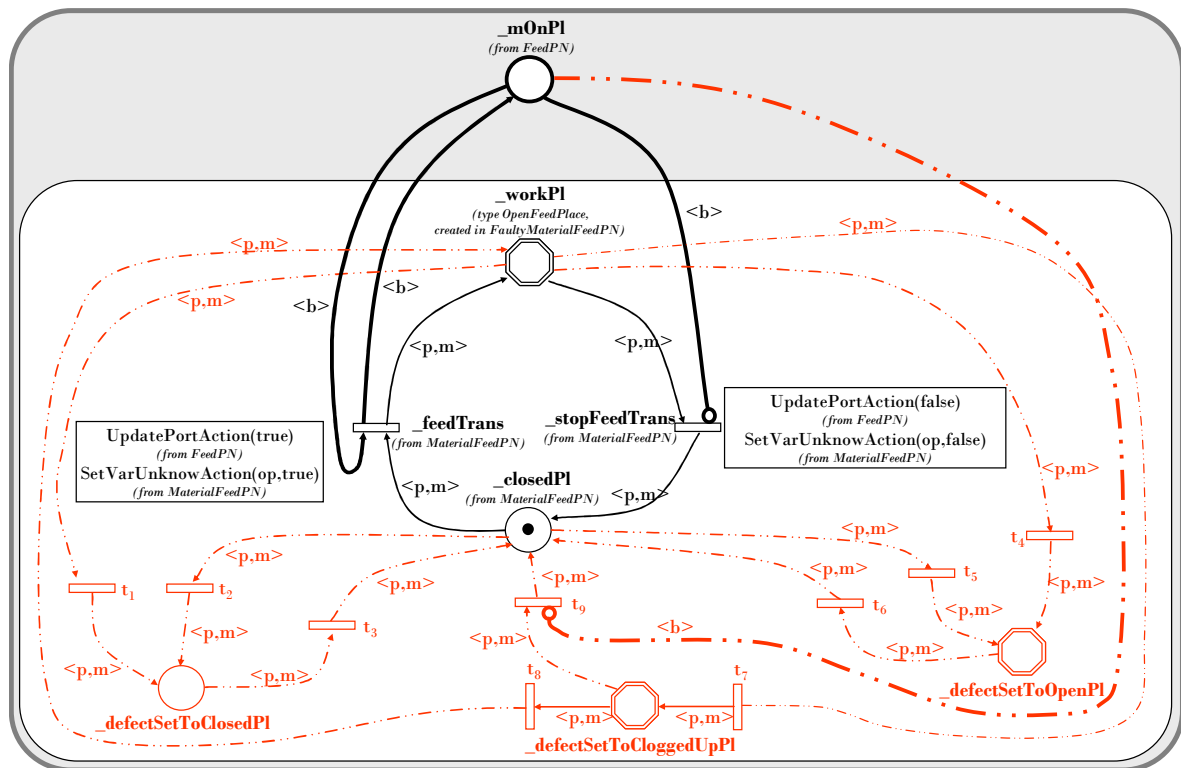


Figure 6.30 Le réseau de l'alimentation matière défaillante

| Transition | Événement | Action |
|--|---|--|
| t_1 ↔ _WorkToDefClosedTrans | SetToClosedFailure == true Apparition du défaut forçant l'alimentation à être dans l'état <i>bloqué fermé</i> alors qu'elle était dans l'état <i>ouvert</i> (commutation forcée de l'alimentation). La place de commande mOpenPI reste cependant marquée. | UpdatePortAction(false) ⇔ l'action permet la désactivation du port de sortie associé à l'alimentation matière, entraînant la désactivation du port d'entrée auquel il est connecté. L'alimentation passe alors dans l'état <i>bloqué fermé</i> (place _defectSetToClosedPI) SetVarUnknownAction(Op,false) ⇔ l'action donne le statut de paramètre à la grandeur <i>Op</i> |
| t_2 ↔ _ClosedToDefClosedTrans | SetToClosedFailure == true Apparition du défaut forçant l'alimentation à rester dans l'état <i>bloqué fermé</i> . | |
| t_3 ↔ _DefClosedToClosedTrans | SetToClosedFailure == false Disparition du défaut forçant l'alimentation à être dans l'état <i>bloqué fermé</i> et retour à l'état <i>fermé</i> en fonctionnement normal (place _closedPI) | |
| t_4 ↔ _WorkToDefOpenTrans | SetToOpenFailure == true Apparition du défaut forçant l'alimentation à être dans l'état <i>bloqué ouvert</i> alors qu'elle était dans l'état <i>ouvert</i> . La place de commande mOpenPI reste marquée. | |
| t_5 ↔ _ClosedToDefOpenTrans | SetToOpenFailure == true Apparition du défaut forçant l'alimentation à être dans l'état <i>bloqué ouvert</i> alors qu'elle était dans l'état <i>fermé</i> . La place de commande mOpenPI n'est évidemment pas marquée. | UpdatePortAction(true) ⇔ l'action permet l'activation du port de sortie associé à l'alimentation matière, entraînant l'activation du port d'entrée auquel il est connecté. L'alimentation passe alors dans l'état <i>bloqué ouvert</i> (place _defectSetToOpenPI) SetVarUnknownAction(Op,true) ⇔ l'action donne le statut de variable du modèle à la grandeur <i>Op</i> |
| t_6 ↔ _DefOpenToClosedTrans | SetToOpenFailure == false Disparition du défaut forçant l'alimentation à être dans l'état <i>bloqué ouvert</i> et retour dans l'état <i>fermé</i> en fonctionnement normal (place _closedPI) | UpdatePortAction(false) ⇔ l'action permet la désactivation du port de sortie associé à l'alimentation matière, entraînant la désactivation du port d'entrée auquel il est connecté. L'alimentation repasse alors dans l'état <i>fermé</i> . SetVarUnknownAction(Op,false) ⇔ l'action donne le statut de paramètre à la grandeur <i>Op</i> |
| t_7 ↔ _WorkToDefCloggedUpTrans | SetToCloggedUpFailure == true Apparition du défaut plaçant l'alimentation dans l'état <i>ouvert dégradé</i> (débit inférieur à la valeur nominale maxFlow) | |
| t_8 ↔ _DefCloggedUpToWorkTrans | SetToCloggedUpFailure == false Disparition du défaut plaçant l'alimentation dans l'état <i>ouvert dégradé</i> et retour dans l'état <i>ouvert</i> en fonctionnement normal (place _workPI) | |
| t_9 ↔ _DefCloggedToClosedTrans | | UpdatePortAction(false) ⇔ l'action permet la désactivation du port de sortie associé à l'alimentation matière, entraînant la désactivation du port d'entrée auquel il est connecté. L'alimentation repasse alors dans l'état <i>fermé</i> . SetVarUnknownAction(Op,false) ⇔ l'action donne le statut de paramètre à la grandeur <i>Op</i> |

Tableau 6.4 Les conditions et actions associées aux transitions du réseau de Petri de l'alimentation matière avec défauts

Considérons le défaut où l'alimentation matière fournit un débit dégradé. Quand la transition t_7 est franchie, le débit fourni par l'alimentation matière dépend d'un taux d'encrassement *TauxEncrassement* (donné en % dans le fichier initialisation). La valeur du débit est alors :

$$\text{Débit} = (1 - \text{TauxEncrassement}) * \text{max Flow} * \text{Op} \quad (6 - 5)$$

Avec *Débit* : débit fourni par l'alimentation,

TauxEncrassement : taux d'encrassement de l'alimentation,

maxFlow : le flux maximal de l'alimentation,

Op : coefficient d'ouverture de l'alimentation.

6.3.4 Les vannes

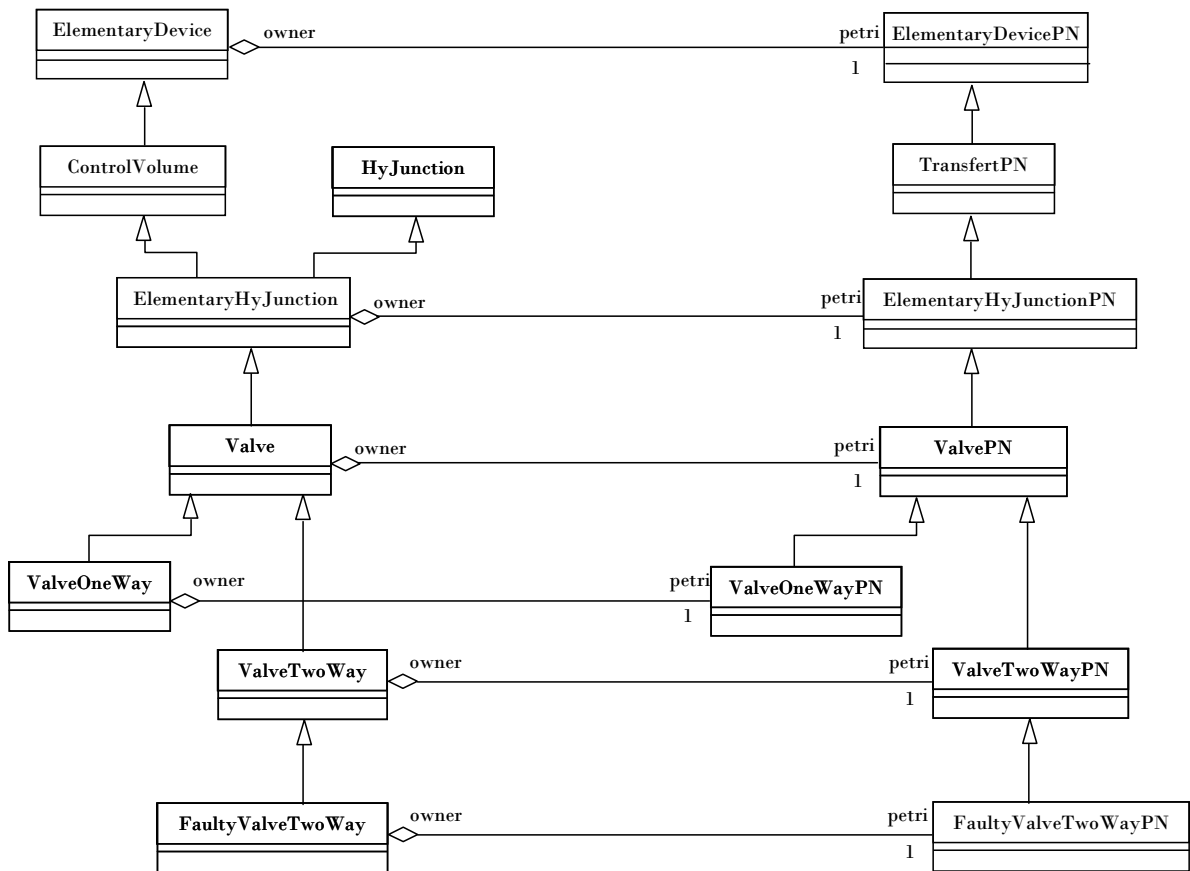


Figure 6.31 Les vannes spécialisées

Dans *PrODHyS*, un seul type de jonction hydraulique spécifique a été développé ; il s'agit de la vanne.

Une vanne, décrite par la classe *Valve* (cf. figure 6.29), est caractérisée par un coefficient d'ouverture op qui vaut « zéro » lorsqu'elle est fermée et « un » lorsqu'elle est complètement ouverte. De plus, plusieurs paramètres dépendant de la géométrie ont été introduits au niveau de sa configuration

ValveCfg. La fonction d'une vanne est d'agir sur un débit en le réglant manuellement ou par l'intermédiaire d'un système de régulation. La vanne entre ainsi dans la catégorie des appareils *actifs*.

6.3.4.1 La vanne sans défaut

6.3.4.1.1 La vanne simple

Le modèle RdPDO qui lui est associé a été présenté au chapitre 2 au point 2.6.2.2.4.

6.3.4.1.2 La vanne à inversion de flux

De même, le modèle RdPDO qui lui est associé a été présenté au chapitre 2 au point 2.6.2.2.4.

6.3.4.2 La vanne avec défaut

Les défauts modélisés de la vanne sont semblables aux défauts de l'alimentation matière. Voici la liste :

- Un flux traverse la vanne alors qu'il ne devrait pas y en avoir ;
- Aucun flux ne traverse la vanne alors qu'il devrait y en avoir ;
- Le débit traversant la vanne a une valeur dégradée (soit plus forte, soit plus faible).

La figure 6.32 représente la modélisation de la vanne à inversion de flux défaillante. Les éléments en trait continu correspondent au modèle de l'appareil « idéal », tandis que les éléments en trait discontinu sont les éléments qui sont ajoutés pour la modélisation des appareils avec défauts.

Aux conditions et actions associées aux transitions de la vanne à inversion de flux sans défaut (cf. tableau 2.6), s'ajoutent les conditions et actions associées au réseau de Petri de la vanne à inversion de flux défaillante. Celles-ci sont résumées dans le tableau 6.5.

Nous envisageons que, deux défaillances conduisent au défaut qui consiste en la fermeture de la vanne au lieu de son ouverture :

- La vanne est fermée sans raison. Cette défaillance est modélisée par le franchissement de la transition t_{14} .
- La vanne reste bloquée dans l'état OFF. Cette défaillance est modélisée par le franchissement de la transition t_{15} .

De même, nous considérons que, deux défaillances amènent au défaut qui consiste en l'ouverture de la vanne au lieu de sa fermeture :

- La vanne reste bloquée dans l'état ON. Cette défaillance est modélisée par le franchissement de la transition t_{12} .
- La vanne est en position ouverte sans raison. Cette défaillance est modélisée également par le franchissement de la transition t_{12} .

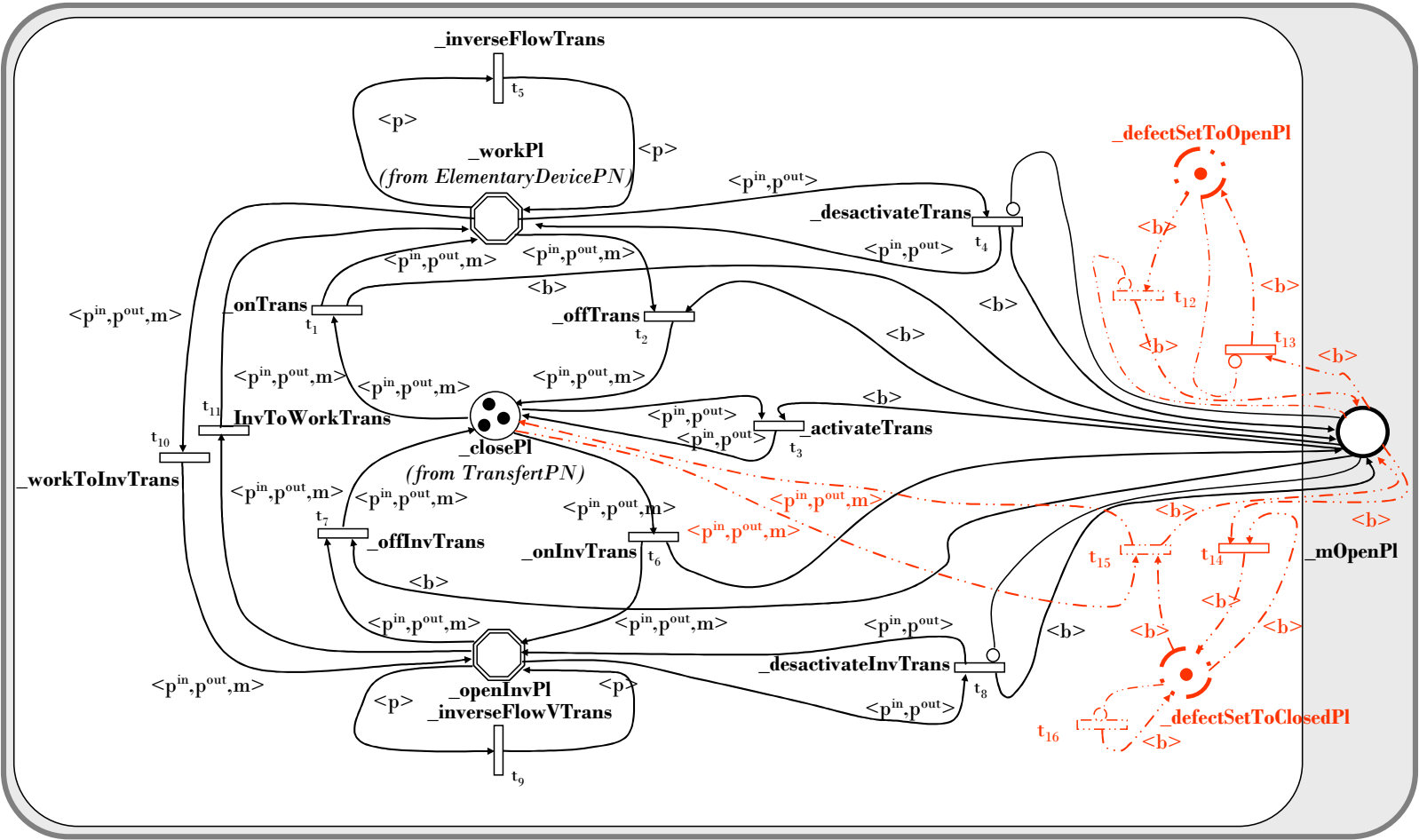


Figure 6.32 Le réseau de Petri de la vane à inversion de flux détaillée

| Transition | Événement | Action |
|---|--|--------|
| $t_{12} \Leftrightarrow$ _SetToOpenOnTrans | SetToOpenFailure == true Apparition du défaut forçant la vanne dans l'état <i>bloqué ouvert</i> alors qu'elle était à l'état <i>fermé</i> ou apparition du défaut forçant la vanne dans l'état <i>bloqué ouvert</i> alors qu'elle était à l'état <i>ouvert</i> . La place de commande mOpenPI devient alors marquée. | |
| $t_{13} \Leftrightarrow$ _SetToOpenOffTrans | SetToOpenFailure == false Disparition du défaut forçant la vanne dans l'état <i>bloqué ouvert</i> | |
| $t_{14} \Leftrightarrow$ _SetToCloseOn1Trans | SetToClosedFailure == true Apparition du défaut forçant la vanne dans l'état <i>bloqué fermé</i> alors qu'elle était à l'état <i>ouvert</i> . | |
| $t_{15} \Leftrightarrow$ _SetToCloseOn2Trans | SetToClosedFailure == true Apparition du défaut forçant la vanne dans l'état <i>bloqué fermé</i> alors qu'elle était à l'état <i>fermé</i> . | |
| $t_{16} \Leftrightarrow$ _SetToCloseOffTrans | SetToClosedFailure == false Disparition du défaut forçant la vanne dans l'état <i>bloqué fermé</i> alors qu'elle était à l'état <i>fermé</i> . | |
| $t_3 \Leftrightarrow$ _activateTrans | SetToClosedFailure == false | |

Tableau 6.5 Les conditions et actions associées aux transitions du réseau de Petri de la vanne à inversion de flux avec défaut

Les transitions t_3 et t_6 permettent de revenir dans un état normal de fonctionnement.

Considérons maintenant le défaut où la vanne est traversée par un flux de débit dégradé. Pour modéliser ce défaut, il faut tout d'abord que la vanne soit dans l'état ouvert (marquage de la place **_mOpenPI**) et que nous avons l'occurrence du défaut validé (**SetToCloggedUpFailure == true**). Le débit traversant la vanne est alors entaché d'un écart permanent avec son coefficient d'ouverture. Cet écart est tiré aléatoirement dans un intervalle $[V_{minOff}; V_{maxOff}]$ (donné dans le fichier initialisation). La valeur du débit est alors fournie par l'équation suivante :

$$Débit = max\ Flow * Op$$

$$avec \ Op = \frac{rand().(V_{maxOff} - V_{minOff})}{RANDMAX} + V_{minOff} \quad (6 - 6)$$

Avec *Débit* : débit fourni par l'alimentation,

maxFlow : le flux maximal de l'alimentation,

Op : coefficient d'ouverture de l'alimentation.

6.4 LA CONCLUSION

Ce chapitre a permis d'illustrer la mise en œuvre de l'approche *SimAEM* au sein de la plateforme *PrODHyS*. Dans ce cadre, il a proposé une description détaillée des éléments le composant (détection, diagnostic, etc.) et de son fonctionnement.

Il a également permis de définir les éléments fondamentaux entrant dans la modélisation d'un procédé défectueux. La modélisation d'un défaut a été présentée. Ensuite, la notion de génération de défauts et le principe de la simulation en parallèle ont été exposés.

Enfin, la dernière section s'est focalisée sur la modélisation des appareils défectueux. Ainsi, la modélisation des capteurs, détecteurs, alimentations et vannes défectueux a été présentée. La liste n'est pas exhaustive, il reste maintenant à étendre ces concepts aux autres appareils, afin d'élaborer une bibliothèque d'appareils défaillants directement disponibles au sein de la plate-forme *PrODHyS*.

PARTIE 4 :

LA MISE EN OEUVRE DE LA

MÉTHODOLOGIE DE

SURVEILLANCE

CHAPITRE 7 L'IMPLÉMENTATION DU SYSTÈME DE SURVEILLANCE DE PROCÉDÉ

Les deux précédents chapitres (5 et 6) se sont consacrés à présenter les concepts généraux sur lesquels se base la méthode SimAEM. Ceci a permis d'aboutir, d'une part, à sa formalisation et d'autre part, à son intégration au sein de la plateforme ProDHYS, à travers le module ProDHYS AEM.

Ce chapitre se propose de décrire maintenant la mise en place du système de surveillance des procédés FDIPro (Fault Detection and Isolation for Process) en exploitant la librairie ProDHYS AEM. Dans ce cadre, nous appliquons la méthodologie SimAEM à l'exemple didactique exposé dans le point 2.6.1. Les procédures pour régler le filtre de Kalman étendu ainsi que pour obtenir la matrice d'incidence sont précisées.

Pour des raisons de sécurité, il n'est évidemment pas envisageable de tester le système de surveillance *FDIPro* directement sur un procédé réel. Afin de réaliser une première évaluation de la méthodologie *SimAEM*, le système de surveillance est testé sur un procédé potentiellement défectueux simulé avec *PrODHyS*. Évidemment toutes les étapes liées au « procédé réel » seront supprimées lorsque le système sera en exploitation sur site. Les sections suivantes décrivent dans ce contexte (procédé simulé) le processus de modélisation pour la mise en place d'un test de surveillance.

7.1 LE PROCESSUS GÉNÉRAL DE MODÉLISATION

La mise en œuvre d'un test unitaire du système de surveillance nécessite plusieurs opérations de la part de l'utilisateur. En effet, il doit :

- décrire le procédé de référence en choisissant les appareils qui le composent et en précisant leur configuration et leur topologie ainsi que les connexions qui les relient. Comme indiqué plus haut, le système est en phase de test et le flowsheet doit donc être reproduit en ne faisant intervenir que des appareils potentiellement défectueux pour simuler le procédé réel ;
- définir la recette générale qui permet la conduite de la simulation. Pour cela l'utilisateur doit :
 - définir la recette de contrôle pilotant le procédé, afin de spécifier la séquence opératoire à suivre. Cette recette est, rappelons le, instanciée deux fois, afin de piloter aussi bien le procédé réel que celui de référence ;
 - établir la recette du générateur de défauts, afin de contrôler l'occurrence des défauts simulés. Dans un premier temps, les dates d'apparition des défauts sont fixées et connues afin de valider notre démarche de surveillance. Dans un second temps, l'occurrence et le type de défaut sont tirés aléatoirement ;
- établir les modèles du système qui sont utilisés, lors de l'exécution du filtre de Kalman étendu et effectuer son réglage ;
- exécuter des séries de simulations de procédés défectueux, afin d'obtenir la matrice d'incidence utilisée pour l'identification et la localisation de défaut ;

Les sections suivantes se proposent de décrire cette démarche de manière détaillée en la mettant en œuvre sur un exemple simple de procédé. La figure 7.1 schématise le procédé pris comme exemple, pour illustrer les concepts présentés dans les chapitres précédents. Celui-ci correspond à l'application traitée dans le chapitre 2. Le lecteur peut se reporter au point 2.6.1 pour de plus amples détails sur ses caractéristiques.

Le procédé étudié se base sur le principe des vases communicants. Il comporte deux bacs reliés entre eux par une vanne. Dans cette application, le bac C1 est connecté à une alimentation et un soutirage peut être effectué sur le bac C₂. De plus, le procédé est instrumenté avec deux détecteurs :

- Le premier contrôle le niveau de liquide dans la cuve C1 et émet une alarme quand le niveau de liquide atteint le niveau haut h_{1max} ;

- Le second surveille le niveau de liquide dans la cuve C_2 et renvoie une alarme lorsque le niveau de liquide atteint le niveau bas h_{2min} .

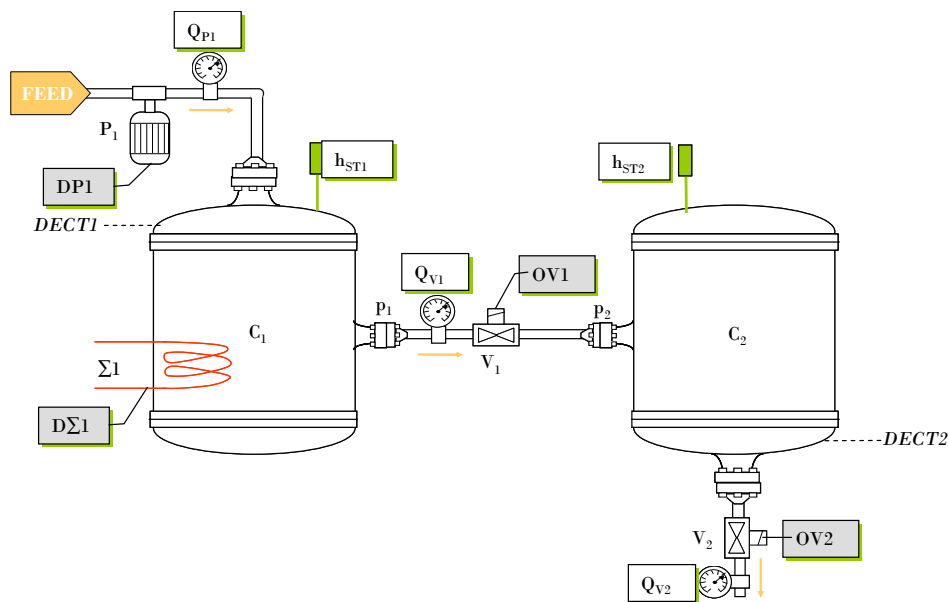


Figure 7.1 L'exemple didactique

Le tableau 2.1 répertorie les données techniques associées à cet exemple. Les conditions opératoires sont, quant à elles, listées dans le tableau suivant :

| | Cuve C1 | Cuve C2 | Alimentation |
|-----------------|---------|---------|--------------|
| T (K) | 298,15 | 298,15 | 298,15 |
| U_1 (mol) | 800 | 2800 | - |
| h_1 (cm) | 15,08 | 52,77 | - |
| Débit (mol/min) | - | - | 10 |

Tableau 7.1 Les conditions opératoires de l'exemple didactique

Soulignons que nous travaillons ici en monophasique avec un mélange binaire eau/méthanol.

7.2 LA MODÉLISATION DU PROCÉDÉ

Le modèle mathématique de ce système a été développé dans le point 0. Nous ne présenterons ici que les aspects liés à la modélisation du procédé sous *PRODHYS*.

7.2.1 Le procédé de référence

Le procédé de référence représente le procédé « idéal » qu'on souhaiterait avoir dans le meilleur des cas. Ainsi, le modèle adopté pour le représenter est constitué uniquement d'appareils non défailants. La figure 2.14 du chapitre 2 illustre le processus de décomposition adopté pour représenter le procédé de référence.

Nous rappelons que le modèle comporte :

- deux cuves (de type StorageTank) ;
- une alimentation matière (de type MaterialFeed);
- une vanne V_2 (de type OneWayValve) pour contrôler le soutirage du bac C_2 ;
- une vanne à inversion de flux $V1$ (de type TwoWayValve) ;
- deux détecteurs de niveau (de type LevelDetector) indiquant respectivement un dépassement du niveau haut dans la cuve $C1$ et du niveau bas dans la cuve C_2 ;
- l'environnement (de type Surroundings).

7.2.2 Le procédé réel

Le procédé réel possède le même flowsheet que le procédé de référence mais avec des appareils potentiellement défectueux. Le principe du procédé précédent est ainsi dupliqué et le modèle est composé alors de :

- deux cuves (de type StorageTank) ;
- une alimentation matière (de type FaultyMaterialFeed);
- une vanne V_2 (de type FaultyOneWayValve) pour contrôler le soutirage du bac C_2 ;
- une vanne à inversion de flux $V1$ (de type FaultyTwoWayValve) ;
- deux détecteurs de niveau (de type FaultyLevelDetector) indiquant respectivement un dépassement du niveau haut dans la cuve $C1$ et du niveau bas dans la cuve C_2 ;
- l'environnement (de type Surroundings).

Nous supposons que pour cet exemple l'environnement et l'appareil cuve sont deux entités ne pouvant pas être défectueuses.

La figure 7.2 illustre le processus de décomposition adopté pour représenter le procédé réel.

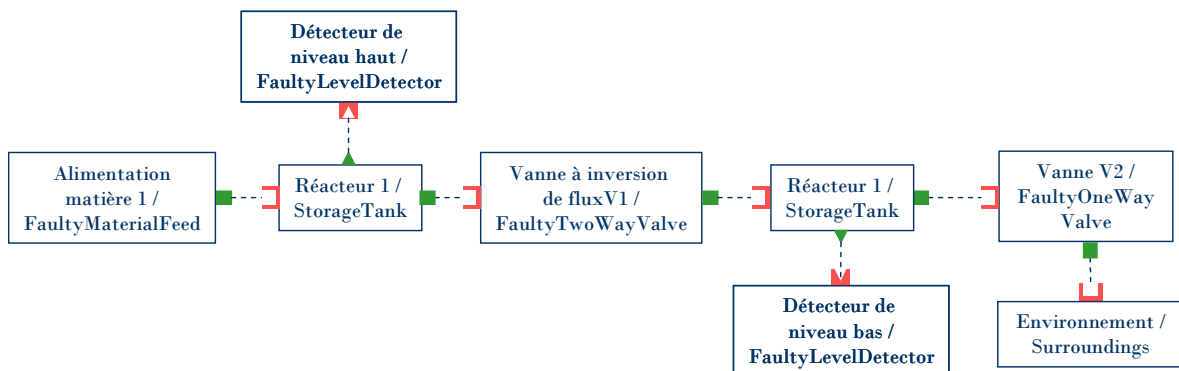


Figure 7.2 Le schéma bloc associé au système réel des vases communicants

La création du flowsheet du procédé est une procédure générale et systématique. Le caractère générique de cette démarche est d'ailleurs un point positif pour automatiser cette opération.

7.3 LA MODÉLISATION DE LA RECETTE PRINCIPALE

La deuxième étape de la conception d'un test du système de surveillance consiste en l'établissement de la recette principale. Celle-ci regroupe trois recettes : la recette pilotant le procédé réel, celle pilotant le procédé de référence et enfin la recette générant les défauts.

7.3.1 La recette pilotant le procédé

7.3.1.1 Le procédé de référence

Dans le cadre de cette application, la séquence opératoire à réaliser est décrite sur la figure 7.3 :

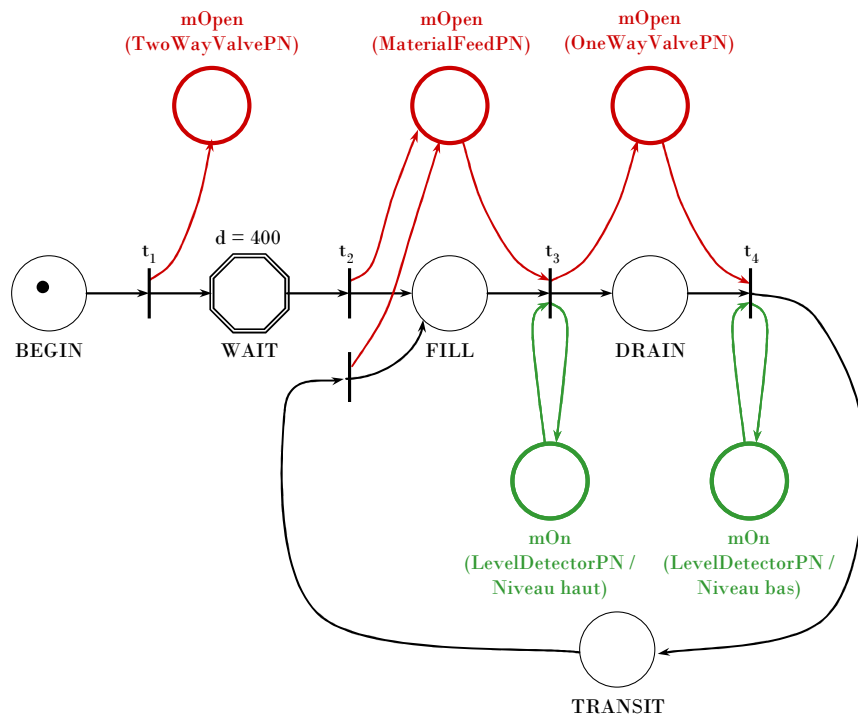
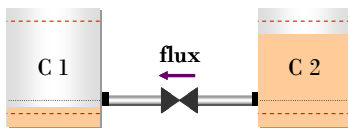
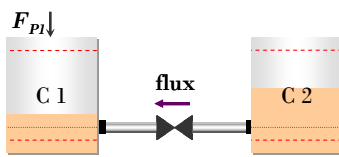


Figure 7.3 La recette de contrôle associée au système des vases communicants

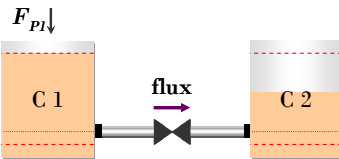
Initialement, d'après le tableau 7.1, chaque cuve contient un mélange binaire méthanol/eau et la hauteur de liquide dans la cuve C1 est inférieure à celle du port p_1 (port de sortie de la cuve C1) tandis que celle de la cuve C2 est supérieure à celle du port p_2 (port d'entrée de la cuve C2). Nous supposons de plus que les vannes et l'alimentation matière sont supposées fermées. Le scénario étudié, dans le cas idéal, est présenté ci-dessous :



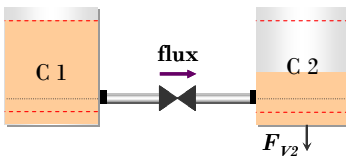
1. La première opération débute par l'ouverture de la vanne V1, vanne reliant les deux bacs. Un flux apparaît alors de la cuve C2 vers C1. Cette configuration est maintenue pendant un temps égal à la temporisation de la place différentielle WAIT. Elle caractérise l'état WaitState.



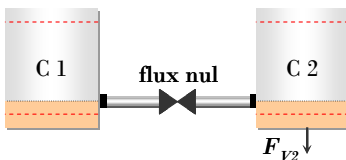
2. La cuve C1 est ensuite alimentée par la pompe P1. Le sens de l'écoulement continue dans un premier temps de la cuve C2 vers la cuve C1 (car $h_{11} < h_{12}$). Ceci définit l'état continu et discret Fill2/1State.



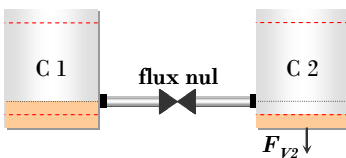
3. Une fois que la hauteur de liquide dans la cuve C2 devient inférieure à celle dans la cuve C1, le flux s'inverse, allant de la cuve C1 vers la cuve C2. L'alimentation est fermée lorsque le niveau maximum est atteint (détecté par DECT1). Cette configuration représente le même état discret que l'étape précédente, mais c'est la partie continue qui a évolué. Nous faisons donc une distinction avec l'état continu et discret précédent en le notant Fill1/2State.



4. Le dépassement du niveau haut de la cuve C1 entraîne le franchissement de la transition t_3 . Cela s'accompagne de l'ouverture de la vanne V2 et le soutirage de la cuve C2 débute. L'état continu et discret Drain1/2State symbolise cette étape.



5. La cuve C1 se vidange dans la cuve C2 jusqu'à la hauteur de liquide dans la cuve C1 atteigne la hauteur du port p_1 . En effet, la différence de pression entre les ports p_1 et p_2 s'annule et le flux qui traversait la vanne V1 disparaît. Cette étape est caractérisée par le même état discret que l'étape précédente, mais par contre la partie continue a évolué (le flux entre les deux cuves a disparu). Nous notons donc cet état continu et discret Drain2State.



6. La vidange de la cuve C2 se poursuit jusqu'à ce que le niveau bas soit atteint par le détecteur DECT2. Une fois cette valeur atteinte, la transition t_4 est franchie, ce qui conduit à la réouverture de l'alimentation de la cuve C1. Le système se trouve alors dans l'étape 3 et le cycle reprend. Ici, il n'existe aucune distinction entre les états continus et discrets de cette étape et de l'étape précédente. Cette configuration représente donc le même état Drain2State.

7.3.1.2 Le procédé réel émulé

Le procédé réel (c'est-à-dire potentiellement défaillant) est piloté par une recette identique à celle représentée sur la figure 7.3. Par contre, les liaisons de seconde instance de recette ne s'effectuent plus vers des appareils idéaux mais vers des appareils potentiellement défectueux :

- Ainsi, les places associées aux actionneurs : mOpen (TwoWayValvePN), mOpen (MaterialFeedPN) et mOpen (OneWayValvePN) deviennent respectivement les places mOpen (FaultyTwoWayValvePN), mOpen (FaultyMaterialFeedPN) et mOpen (FaultyOneWayValvePN) ;
- De même, les places associées aux détecteurs mOn (LevelDetectorPN/Niveau Haut) et mOn (LevelDetectorPN/Niveau Bas) se transforment en mOn (FaultyLevelDetectorPN/Niveau Haut) et mOn (FaultyLevelDetectorPN/Niveau Bas).

7.3.1.3 Le procédé reconstruit par le filtre de Kalman étendu

Le filtre de Kalman étendu s'appuie sur le modèle continu représentant la dynamique du système, pour reconstruire l'état de ce dernier. Ce modèle est obtenu par le jeu d'un réseau de Petri identique à celui de la recette de référence. C'est pourquoi, ici encore, le filtre de Kalman s'appuie sur la même recette que celle du modèle de référence (cf. figure 7.3).

7.3.2 La recette de génération de défauts

Nous considérons pour cette étude l'occurrence d'un défaut sur la vanne V1 : la vanne se ferme sans raison. La recette de génération du défaut est représentée sur la figure 7.4. Par exemple, au franchissement de la transition t_2 , l'occurrence du défaut est activée par la méthode Activate(c), et cela pendant une durée correspondant à la temporisation de la place différentielle DURATION_D1 (ici, à 302 min). Notons que la date d'occurrence du défaut correspond ici à la temporisation de la place différentielle TIME_D1 (ici, pendant 500 min).

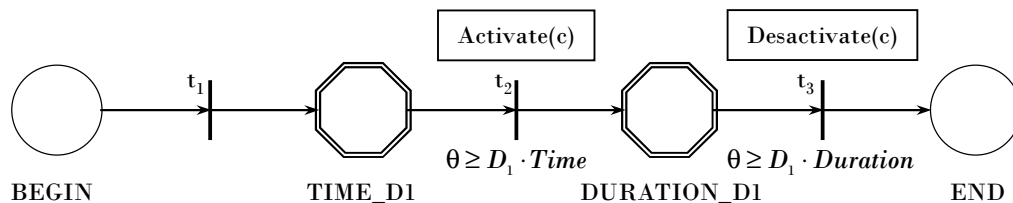


Figure 7.4 La recette de génération de défaut associée au système des vases communicants

7.3.3 La recette générale

La recette générale correspond au regroupement des trois recettes précédentes. La figure 7.5 représente la recette générale de l'exemple didactique étudié ici. Notons que pour différencier les places des réseaux de Petri composants, un préfixe est ajouté au nom des places :

- un « M- » pour la recette du procédé de référence,
- un « K- » pour la recette utilisée par le filtre de Kalman,
- un « F- » pour la recette du générateur de défaut,
- un « P - » pour la recette du procédé réel.

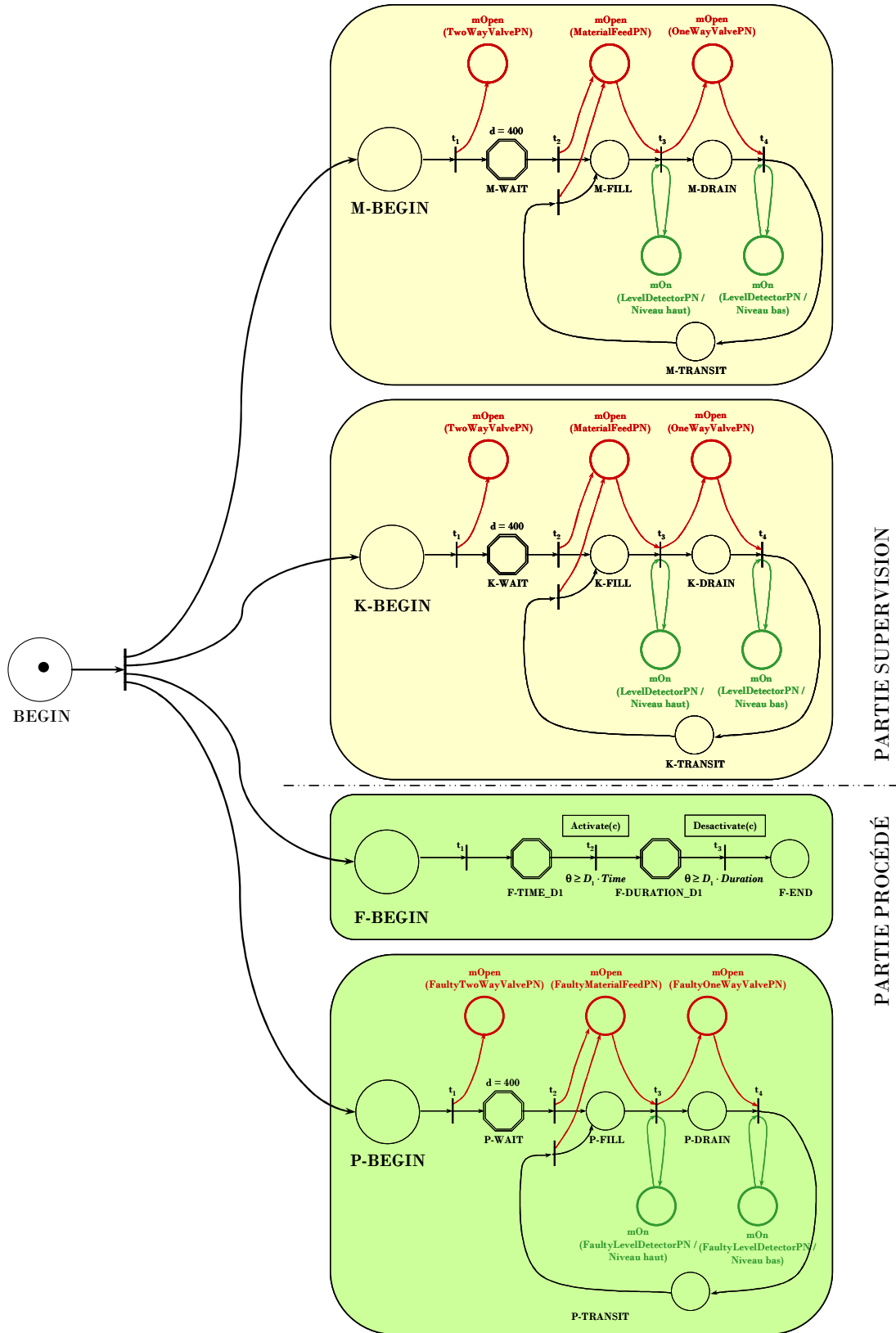


Figure 7.5 La recette générale associée au système des vases communicants

7.4 LA MISE EN OEUVRE DU FILTRE DE KALMAN

En pratique, la modélisation des processus physiques contrôlant la dynamique et la structure des procédés (systèmes dynamiques hybrides étudiés dans ce rapport) est une approximation du système et donc n'est pas parfaite. L'utilisation du filtre de Kalman étendu suppose que les erreurs de modélisation et de mesures soient identifiées. Pour cela, il faut évaluer correctement la variance des différents bruits affectant le système. Ces réglages sont le sujet de ce point.

Dans un premier temps, nous exposons les équations sur lesquelles s'appuie le filtre de Kalman étendu pour reconstruire l'état du système. Sachant que les équations du modèle du système ont été exposées précédemment, les équations des observations sont le sujet du point suivant.

7.4.1 Les équations des observations

Les mesures effectuées sur le système constituent l'observation utilisée pour corriger l'estimation du système faite par le filtre de Kalman étendu. Ces mesures sont relativement simples à manipuler puisqu'elles représentent le plus souvent directement une variable d'état du système. Rappelons que les observations sont décrites par l'équation suivante :

$$Z_k = h(X_k) + W_k \quad (7 - 1)$$

Lorsque les mesures sont directement des variables d'état du système, ou linéairement dépendantes, la fonction h corrélant les observations au vecteur d'état est donc une fonction linéaire et l'équation (7 - 1) devient :

$$Z_k = H \cdot X_k + W_k \quad (7 - 2)$$

Où $H \in \mathcal{R}^{m \times n}$ avec m le nombre d'observations et n la dimension du vecteur d'état.

Chaque composante de cette matrice est définie par la relation suivante :

$$\forall i \in [1; m], \forall j \in [1; n]: H_{i,j} = \begin{cases} 0 & \text{si } Z_{k,i} \text{ n'est pas une observation de la variable d'état } X_{k,j} \\ 1 & \text{si } Z_{k,i} \text{ est une observation de la variable d'état } X_{k,j} \end{cases} \quad (7 - 3)$$

Remarque : Notons que si toutes les variables d'état sont observables directement, la matrice H correspond à la matrice identité.

Dans la suite de ce mémoire, nous supposons que les mesures sur le système sont des observations d'une variable d'état.

Néanmoins, la structure de la classe `systemDynamic` permet l'implémentation d'équations d'observations plus complexes. Par exemple, dans le cas linéaire, on peut envisager que l'observation faite est une combinaison linéaire de variables d'état. Il suffira dans ce cas là d'écrire la relation reliant l'observation aux variables d'état :

$$Z_{k,i} = \alpha \cdot X_{k,j} + \beta \cdot X_{k,p} \quad (7 - 4)$$

Ainsi la ligne i de la matrice H s'écrit :

$$H_{i,\bullet} = [0 \dots \alpha \dots \beta \dots 0] \quad (7 - 5)$$

$i^{\text{ème}}$ $p^{\text{ème}}$: numéro de colonne

On peut également considérer qu'une observation est reliée aux variables d'état par une fonction non linéaire. Dans ce cas, on utilise la relation (7 - 1).

7.4.2 Le réglage du filtre

Le réglage des paramètres du filtre de Kalman étendu consiste principalement en l'ajustement des valeurs des matrices de covariance du processus et des observations.

7.4.2.1 Les bruits de mesure

Les observations du système sont obtenues par l'utilisation de capteurs spécifiques. Les erreurs de mesures sont alors relativement bien connues du fabricant ou acquises par expérimentation. Ainsi, il est aisé de déterminer une estimation de la matrice de covariance des observations. Dans notre cas, notons que chaque appareil de contrôle a comme caractéristique une erreur de mesure.

7.4.2.2 Les bruits de modèle

Avant d'implanter le filtre de Kalman étendu, il est important d'estimer la sensibilité du filtre aux paramètres du modèle. Pour réaliser cette étude, nous nous appuyons sur une méthode d'ensemble. Elle permet de calculer empiriquement les statistiques d'erreur. Pour cela, elle se base sur un certain nombre de simulations, au cours desquelles des erreurs sont générées par le code lui-même. Ces erreurs sont obtenues en perturbant aléatoirement un des paramètres d'entrée ou un des processus physiques modélisés (dans notre cas, nous avons utilisé une loi normale de variance σ^2 à définir). Ainsi, l'ensemble des trajectoires possibles du modèle est établi. Son utilisation est schématisée sur la figure suivante :

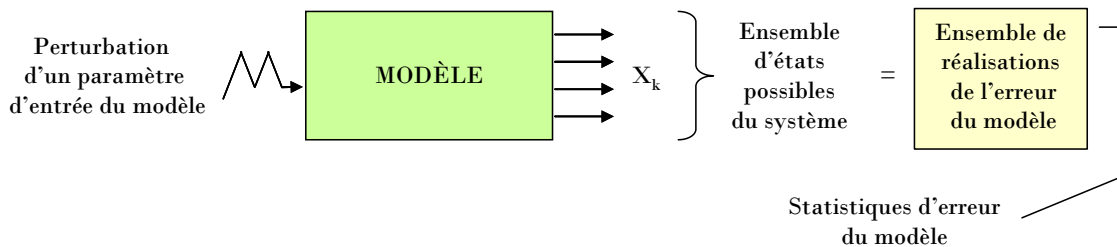


Figure 7.6 L'utilisation d'une méthode d'ensemble pour la spécification des statistiques d'erreur du modèle

Cette méthode repose sur l'hypothèse suivante : la moyenne d'ensemble correspond à la meilleure estimation de l'état du système (notre modèle étant supposé être non biaisé), et la dispersion autour de cette moyenne à une mesure de l'erreur associée à cette même estimation [Evensen, 2003 ; Maget, 2007]. Nous estimons ainsi une large incertitude sur les processus physiques modélisés afin d'être certain que la dispersion obtenue englobe le comportement réel du système.

| Processus,paramètres ou conditions initiales étudiés | Incertitudes appliquées |
|--|--|
| Rétention liquide dans la cuve C1 | $U_{I1}^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 50)$ |
| Rétention liquide dans la cuve C2 | $U_{I2}^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 50)$ |
| Alarme haute capteur de niveau cuve C1 | $h_{I1_{max}}^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 5)$ |
| Alarme basse capteur de niveau cuve C2 | $h_{I2_{max}}^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 5)$ |
| Débit nominal d'alimentation matière | $F_E^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 10)$ |
| Coefficient d'ouverture de la vanne V1 | $Op_{V1}^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 1)$ |
| Coefficient d'ouverture de la vanne V2 | $Op_{V2}^{ref} + \varepsilon$ avec $\varepsilon \in \mathcal{N}(0, \sigma = 1)$ |

Tableau 7.2 Les incertitudes appliquées sur les processus, paramètres et conditions initiales du modèle

Remarque : L'incertitude appliquée au coefficient d'ouverture des vannes revient à modéliser une incertitude sur le débit en sortie de la vanne. De plus, notons que les incertitudes appliquées auraient pu être traitées en relatif.

En pratique, on réalise un ensemble de simulations en parallèle en ajoutant un bruit au processus ou paramètre considéré, uniformément réparti entre $-\sigma$ et σ . Ainsi, il est possible de déduire les caractéristiques (moyenne et écart type) de la dispersion de la fonction de distribution du système. Le tableau 7.2 résume les incertitudes appliquées en fonction du processus considéré.

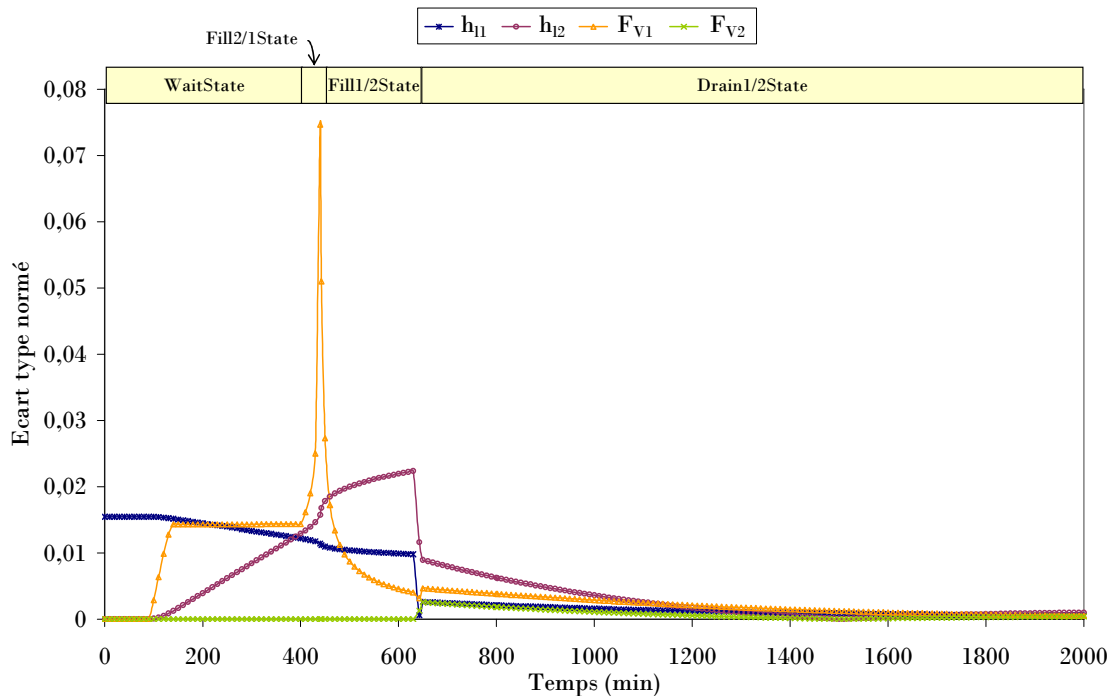


Figure 7.7 L'utilisation d'une méthode d'ensemble pour la spécification des statistiques d'erreur du modèle

Par exemple, présentons les résultats obtenus en modifiant la condition initiale de la rétention liquide dans la cuve C1. Les courbes présentées représentent l'évolution au cours du temps des écarts-types

pondérés des variables d'état suivantes : les hauteurs liquides des cuves C1 et C2 (h_{11} et h_{12}) les débits molaires traversant les vannes V1 et V2 (F_{V1} et F_{V2}). Notons que ces écarts-types ont été normés afin de pouvoir les comparer entre eux. Nous avons également fait figurer sur cette courbe le chronogramme d'état du système. Nous verrons que les variations sont liées aux changements d'état continu et/ou discret du système.

Remarque : L'incertitude appliquée au coefficient d'ouverture des vannes revient à modéliser une incertitude sur le débit en sortie de la vanne.

Nous avons fait varier la rétention liquide dans la cuve C1. La hauteur de liquide dans cette cuve h_{11} est directement proportionnelle à cette grandeur, c'est pourquoi nous constatons dès le début de la simulation un écart-type important (correspondant à la dispersion étudiée sur la rétention liquide initiale).

Première phase caractérisant l'état discret continu WaitState

À $t = 100$ min, nous constatons une augmentation de l'écart-type du flux traversant la vanne V1. Ceci est dû aux phénomènes hydrauliques régissant la vanne V1. En effet, lorsque le niveau de liquide dans la cuve C1 atteint le niveau du port p_1 , port d'entrée de la vanne V1, le débit traversant cette vanne diminue. La dispersion appliquée à la rétention dans la cuve C1 implique que cet événement n'est pas atteint en même temps pour toutes les simulations.

Nous constatons qu'à partir de $t=100$ min, l'écart-type de la hauteur de liquide dans la cuve C2 augmente. Ceci est traduit par les mêmes explications précédentes, puisque le niveau de la cuve C2 est proportionnel au flux traversant la vanne V1.

Deuxième phase caractérisant l'état discret continu Fill2/1State

À $t=400$ min, l'alimentation matière de la cuve C1 s'ouvre, ce qui explique le décrochement que l'on peut constater sur les différentes courbes.

Deuxième phase caractérisant l'état discret continu Fill1/2State

À $t=450$ min, on constate que l'écart-type associé à la hauteur de liquide dans la cuve C2 augmente rapidement. Ceci est dû à l'évènement suivant : c'est aux alentours de cet instant que la hauteur de liquide dans la cuve C2 devient inférieure à celle dans la cuve C1, et qu'alors le flux s'inverse, allant de la cuve C1 vers la cuve C2.

Deuxième phase caractérisant l'état discret continu Drain2State

Enfin, à $t= 650$ min, l'alimentation matière est fermée (lorsque le niveau de liquide dans la cuve C1 a atteint la valeur max détectée par le capteur). Ceci explique les décrochements obtenus sur les différentes courbes. Enfin les courbes finissent par se stabiliser, soulignant ainsi l'établissement d'un régime permanent.

Par conséquent, le modèle est plus sensible à l'incertitude sur les paramètres d'entrée entre $t=0$ min et $t= 600$ min. C'est donc au cours de cette période, que le filtre de Kalman étendu aura toute son importance.

Cette étude a été reproduite pour toutes les incertitudes exposées dans le tableau 7.2. Nous n'exposerons pas ici leurs résultats.

Ces différentes simulations ont donc permis d'estimer un ordre de grandeur des incertitudes de modélisation (cf. tableau 7.3) :

| h_{11} | h_{12} | F_{V1} | F_{V2} | F_{feed} | U_{11} | U_{12} |
|----------|----------|-------------|--------------|-------------|----------|----------|
| 2 cm | 1,6 cm | 0,1 mol/min | 0,15 mol/min | 0,4 mol/min | 108 mol | 56 mol |

Tableau 7.3 Les incertitudes du modèle

7.4.2.3 L'initialisation au changement de modèle

Nous rappelons que nous travaillons en simulation dynamique. Ainsi durant une même simulation, plusieurs modèles du système sont instanciés. À chaque changement de modèle, il faut donc judicieusement initialiser le vecteur d'état et la matrice de covariance :

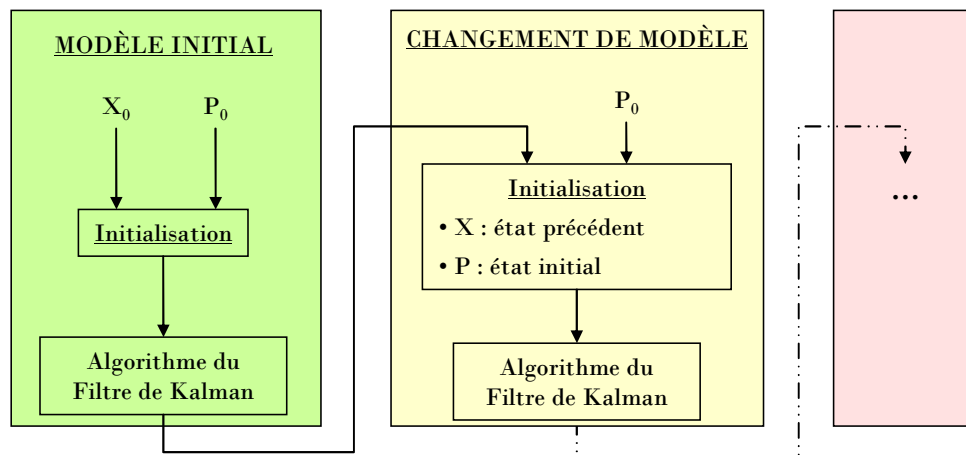


Figure 7.8 L'initialisation du filtre de Kalman étendu

L'initialisation du vecteur d'état

Puisque nous travaillons sur variables continues, l'initialisation du vecteur d'état du nouveau modèle est effectuée à partir de la valeur de l'état précédent (cf. figure 7.8).

L'initialisation de la matrice de covariance

Ici, nous pourrions également partir du principe que la matrice de covariance caractérisant les erreurs du nouveau modèle est initialisée à partir de celle du précédent modèle. Ceci signifierait que l'on fait autant confiance au nouveau modèle qu'à l'ancien. Or, les erreurs de modèle sont l'une des causes les plus importantes de la divergence du filtre. Ceci est notamment dû à la trop grande confiance que le filtre accorde au modèle. C'est le cas lorsque le bruit du modèle est faible. En effet, dans ce cas, les termes des

matrices de covariance de l'erreur d'estimation et la matrice de gain diminuent. Le filtre ne prend alors plus en compte les observations. Il faut donc ajuster intelligemment les matrices de covariance afin de remédier à ce problème. Une des solutions qui sera principalement utilisée ici est d'augmenter l'incertitude du modèle. Nous avons donc fait le choix qu'à chaque changement de modèle, la matrice de covariance du nouveau modèle est initialisée par la valeur initiale de la matrice de covariance (cf. figure 7.8), c'est-à-dire la valeur obtenue à partir du tableau 7.3.

7.5 LA MATRICE D'INCIDENCE

La matrice d'incidences regroupe les signatures de défauts théoriques. Pour l'obtenir, on s'appuie généralement sur l'expérience. Dans notre cas, nous nous basons sur la simulation de procédé défectueux.

7.5.1 Les défauts pris en compte

Nous avons donc dans un premier temps listé les défauts susceptibles d'être présents sur le procédé. Notre exemple traitant le principe des vases communicants, nous nous sommes concentrés sur les défauts liés au débit. Nous avons donc considéré les défauts suivants :

- Défaut 1 : La vanne V1 (cf. figure 7.1) se ferme alors qu'elle ne devrait pas.
- Défaut 2 : La vanne V1 fournit un débit dégradé.
- Défaut 3 : L'alimentation matière (cf. figure 7.1) ne fournit plus de débit.
- Défaut 4 : L'alimentation fournit de la matière à un débit dégradé.
- Défaut 5 : La vanne V2 (cf. figure 7.1) se ferme alors qu'elle ne devrait pas.
- Défaut 6 : La vanne V2 fournit un débit dégradé.
- Défaut 7 : Le détecteur de niveau haut de la cuve C1 ne fonctionne pas.

7.5.2 La construction initiale

En pratique, on réalise un ensemble de simulations en parallèle en introduisant un défaut, à des dates d'occurrence uniformément réparties dans le temps pour chaque état potentiel du système dynamique hybride. On obtient alors l'impact d'un défaut à chaque point de fonctionnement, à chaque état du système. Ensuite, l'analyse exposée au point 5.4 est réalisée. Il est possible alors d'en déduire la signature théorique du défaut considéré.

Par exemple, présentons les résultats obtenus pour le défaut suivant : la vanne V1 (cf. figure 7.1) se ferme alors qu'elle devrait être ouverte. Nous avons effectué un ensemble de simulations pour chaque état du système correspondant à une date d'occurrence du défaut différente pour chacune. Rappelons le chronogramme d'état continu et discret du système :

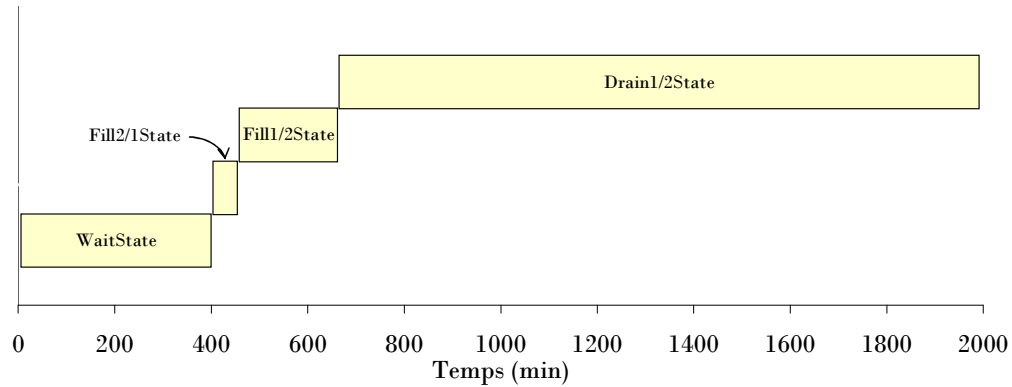


Figure 7.9 Le chronogramme d'état continu et discret du système

La figure 7.10 représente des signatures obtenues à des dates données pour l'occurrence du défaut pendant un état donné ou au cours d'un régime transitoire. Nous n'avons représenté ici que quelques unes des simulations effectuées.

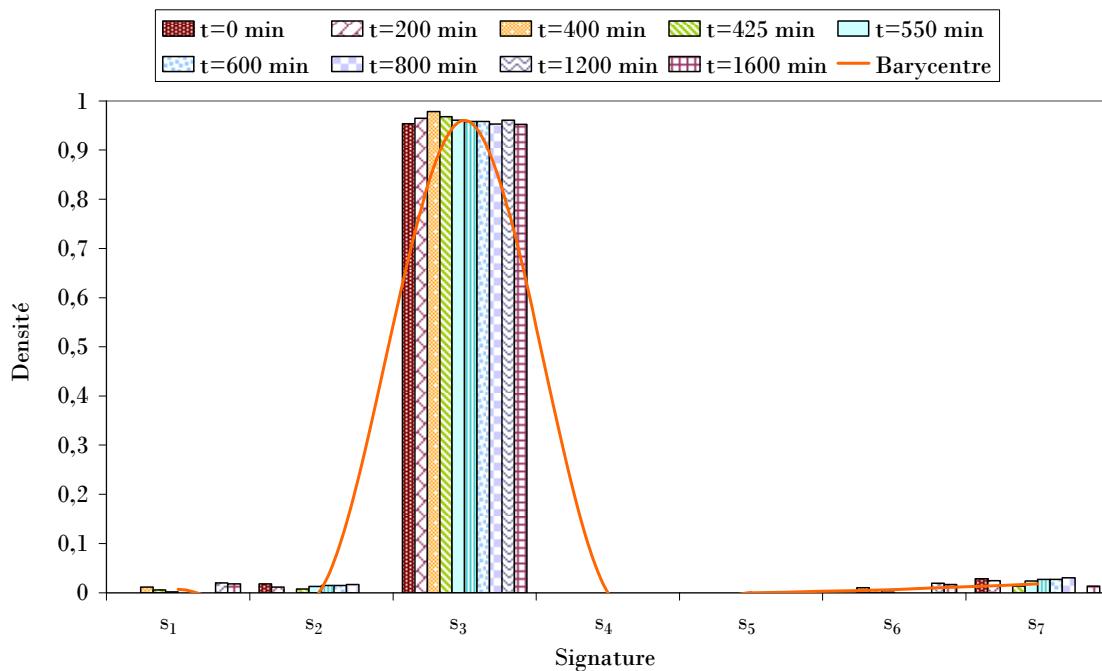


Figure 7.10 Les résidus caractérisant l'occurrence du défaut

Nous avons considéré sept types de signatures :

- La signature s_1 représente la hauteur de liquide dans la cuve C1.
- La signature s_2 caractérise la hauteur de liquide de la cuve C2.
- La signature s_3 est calculée à partir de la variable de flux traversant la vanne V1.
- La signature s_4 fait intervenir la variable de flux traversant la vanne V2.
- La signature s_5 représente le débit d'alimentation.

- La signature s_6 exploite la variable de rétention liquide de la cuve C1,
- Tandis que la signature s_7 exploite celle de la cuve C2.

Les histogrammes présentés soulignent que la signature s_3 est particulièrement sensible au défaut étudié. Cette signature fait en effet intervenir directement la variable de flux traversant la vanne V1, d'où sa sensibilité au défaut de cette vanne.

De ces différentes simulations, nous en déduisons la signature théorique du défaut, correspondant au barycentre des signatures obtenues à partir des différentes simulations. Cette signature est représentée sur la figure 7.10 par une courbe orange. Nous obtenons donc la signature de défaut théorique suivante, que nous avons arrondie, afin de faire ressortir les symptômes :

| | Signature de défaut théorique défaut 1 non arrondie | Signature de défaut théorique défaut 1 arrondie |
|-------|---|---|
| s_1 | 0,007025503 | 0,01 |
| s_2 | 0,007909611 | 0,01 |
| s_3 | 0,960535415 | 0,96 |
| s_4 | 0 | 0 |
| s_5 | 0 | 0 |
| s_6 | 0,01054423 | 0,01 |
| s_7 | 0,01398524 | 0,01 |

Tableau 7.4 La signature théorique du défaut 1

Cette étude a été reproduite pour tous les défauts considérés (exposés au point 7.5.1). Nous n'exposerons pas ici leurs résultats.

Ces différentes simulations ont donc permis de construire la matrice d'incidence de référence exposée dans le tableau 7.5.

| | Défaut 1 | Défaut 2 | Défaut 3 | Défaut 4 | Défaut 5 | Défaut 6 | Défaut 7 |
|-------|----------|----------|----------|----------|----------|----------|----------|
| s_1 | 0,01 | 0 | 0 | 0 | 0 | 0 | 0,3 |
| s_2 | 0,01 | 0 | 0 | 0 | 0,01 | 0 | 0 |
| s_3 | 0,96 | 0,97 | 0,04 | 0,15 | 0 | 0 | 0,4 |
| s_4 | 0 | 0 | 0 | 0 | 0,98 | 0,99 | 0 |
| s_5 | 0 | 0 | 0,96 | 0,85 | 0 | 0 | 0 |
| s_6 | 0,01 | 0 | 0 | 0 | 0 | 0 | 0,3 |
| s_7 | 0,01 | 0,03 | 0 | 0 | 0,01 | 0,01 | 0 |

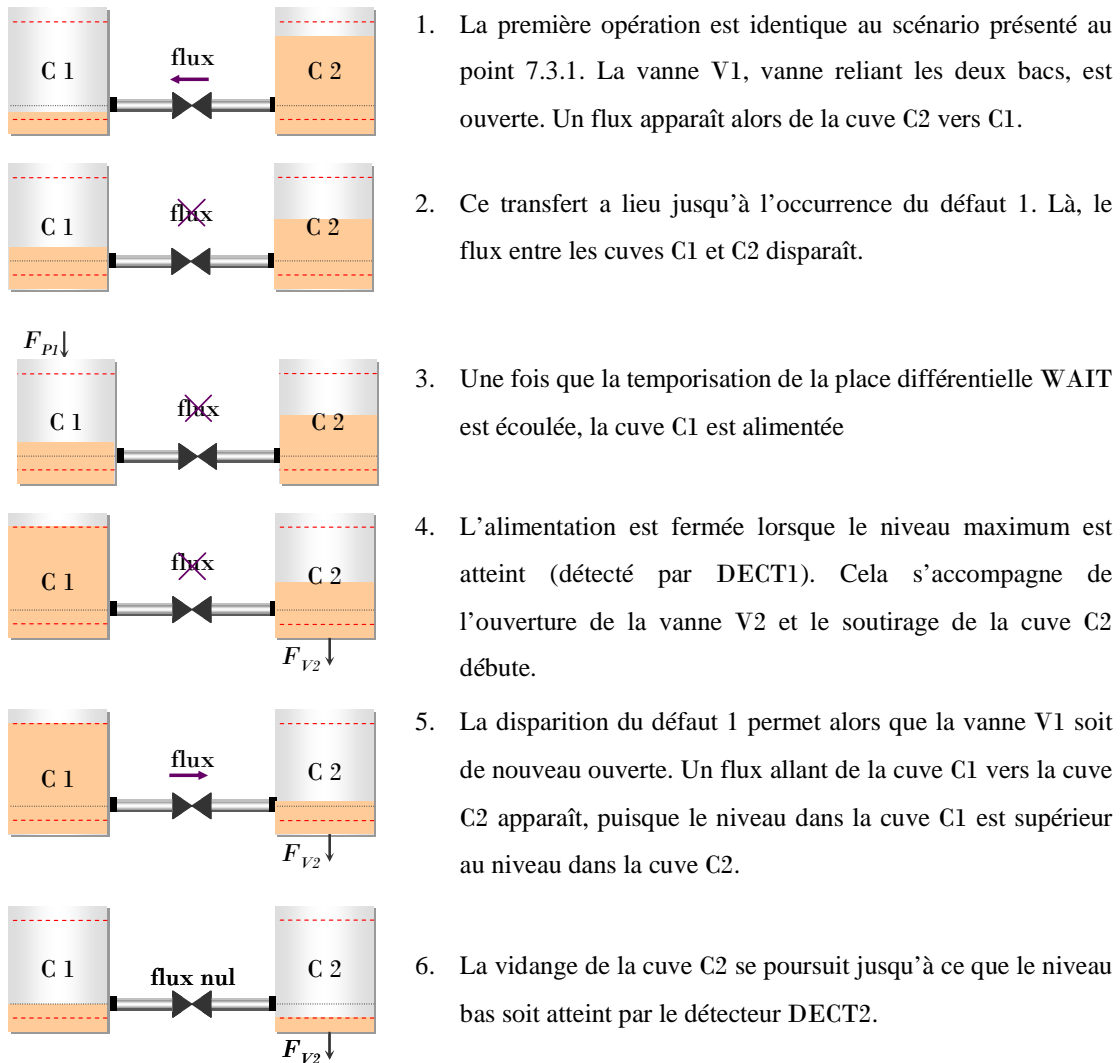
Tableau 7.5 La matrice d'incidence

Remarque : si nous avons travaillé en binaire les défauts 3 et 4 n'auraient pas pu être distingués, puisqu'ayant les mêmes symptômes ils auraient eu la même signature. De plus, les nouveaux diagnostics enrichissent la connaissance du système. En particulier, les signatures de défauts instantanées diagnostiquées sont ajoutées à la forme caractéristique du défaut. Ceci constitue un aspect évolutif du système.

7.6 LES RÉSULTATS

Afin de valider notre méthode, nous avons introduit le défaut 1 : La vanne V1 se ferme alors qu'elle ne devrait pas. Décrivons dans un premier temps le scénario étudié.

7.6.1 Le scénario



7.6.2 La génération de signatures

Tout d'abord illustrons la reconstruction d'état obtenue par le filtre de Kalman étendu. La figure 7.11 représente la reconstruction d'état et les mesures du système. On constate que la reconstruction obtenue par le filtre de Kalman étendu se superpose bien aux mesures du procédé réel.

Afin de déterminer les erreurs de modélisation pour le filtre de Kalman, nous nous sommes basées sur l'hypothèse que la dispersion obtenue englobe le comportement réel du système. En d'autres termes, si le comportement du système sort de cette « enveloppe », son comportement est anormal. Ainsi

les seuils de détection que nous nous sommes fixés correspondent aux incertitudes de modélisation présentées dans le tableau 7.3.

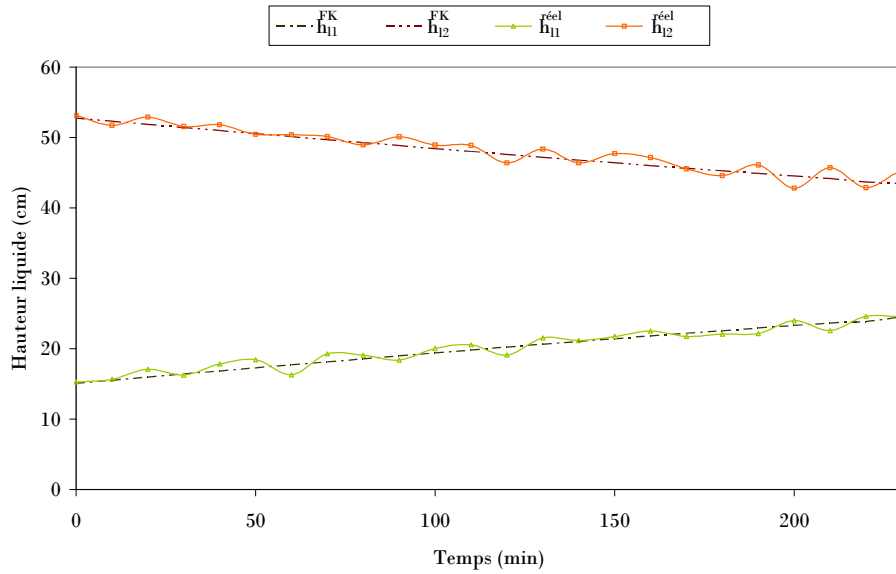


Figure 7.11 La reconstruction d'état obtenue par le filtre de Kalman étendu

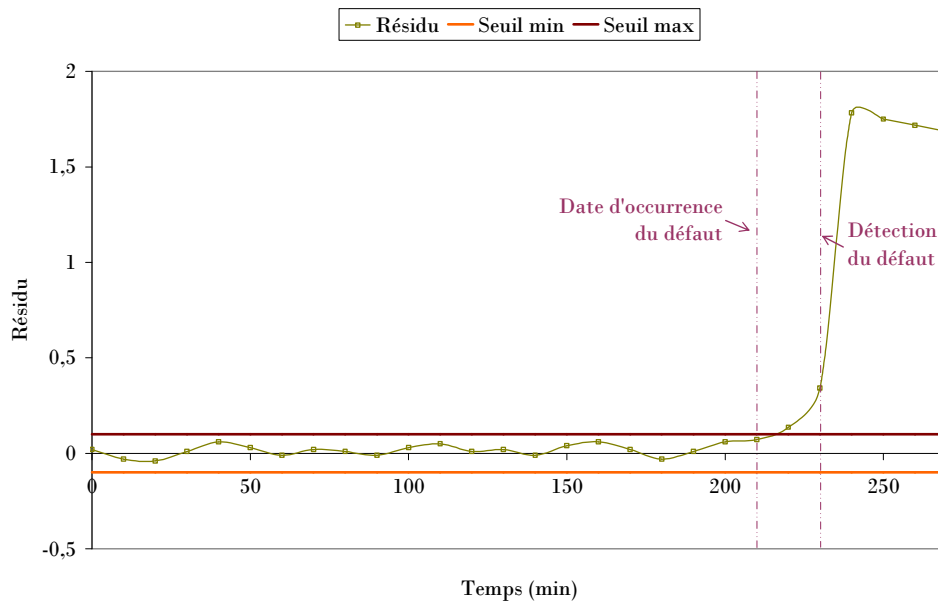


Figure 7.12 La détection

À $t = 210$ min, le défaut 1 est introduit. À $t = 220$ min, on constate que le résidu dépasse le seuil maximum. On ne déclenche pas encore à ce moment là le diagnostic. En effet, afin d'éviter les fausses alarmes, on attend le pas de temps suivant pour vérifier la valeur du résidu. Dans notre cas, le résidu reste hors norme ce qui confirme la possibilité de la présence d'un défaut.

Remarque : Si on attend un pas de temps supplémentaire, il y aurait bien sûr confirmation mais le résidu aurait alors une valeur très importante, signe que le défaut est déjà bien établi. L'intérêt d'un système de surveillance est la détection au plus tôt de ce type d'incident.

On évalue alors le résidu et on obtient la signature de défaut instantanée correspondante. En écrivant la relation (5 - 15), on obtient la signature suivante :

| | Signature de défaut instantanée |
|-------|---------------------------------|
| s_1 | 0,09806576 |
| s_2 | 0,08424535 |
| s_3 | 0,81768889 |
| s_4 | 0 |
| s_5 | 0 |
| s_6 | 0 |
| s_7 | 0 |

Tableau 7.6 La signature de défauts instantanée

7.6.3 La localisation et l'identification de défauts

On compare alors la signature instantanée de défauts (tableau 7.6) à la matrice d'incidence (tableau 7.5) en calculant les indicateurs de défauts relatifs aux distances de Manhattan relative (cf. définition 5-10) et améliorée (cf. définition 5-14). On obtient alors les indicateurs suivants :

| | Défaut 1 | Défaut 2 | Défaut 3 | Défaut 4 | Défaut 5 | Défaut 6 | Défaut 7 |
|----------------------------------|-------------|------------|------------|------------|------------|------------|------------|
| Indicateur de Manhattan relatif | 0,953625398 | 0,94791111 | 0,72571429 | 0,75714286 | 0,71714286 | 0,71428571 | 0,85659022 |
| Indicateur de Manhattan amélioré | 0,86155823 | 0,75026266 | 0,03093867 | 0,11602 | 0,03875755 | 0,0198 | 0,68234417 |

Tableau 7.7 Les indicateurs de défauts de l'exemple didactique

L'indicateur de Manhattan relatif ne fournit pas de résultats satisfaisants. Aucun défaut n'est réellement discriminé puisque leurs indicateurs sont tous supérieur à 0,68 (cf. le point 5.5.3). Par contre, l'indicateur de Manhattan amélioré permet d'écarter les défauts 3, 4, 5 et 6 correspondant respectivement à la fermeture de l'alimentation matière, à l'alimentation dégradée, à la fermeture de la vanne V2 et à la vanne V2 dans un état dégradé (le débit la traversant est dégradé). Les trois possibilités restantes sont :

- Le défaut 1 qui correspond à la fermeture de la vanne V1 sans raison avec un indicateur de défaut de 86% ;
- Le défaut 2 (la vanne V1 fournit un débit dégradé) avec un indicateur de défaut de 75% ;
- Et enfin, le défaut 7 qui consiste en le dysfonctionnement du détecteur de niveau haut avec un indicateur de défaut de 68%.

Deux défauts sont liés à la vanne V1 et qui plus est, ils ont les deux taux les plus élevés. De ce fait, on peut conclure sur la localisation du défaut : le défaut est présent sur la vanne V1 et on estime qu'elle a plus de chance d'être dans un état fermé que dégradé. La localisation et l'identification du défaut sont donc validées.

7.7 LE BILAN

Ce chapitre nous a permis de transcrire une procédure générale et systématique à suivre afin d'effectuer la surveillance d'un procédé. Cette procédure comprend les étapes suivantes :

1. déclarations des objets,
2. création du lien avec l'univers thermodynamique,
3. création du procédé (création de l'objet de type Flowsheet entraînant la création automatique de l'appareil composé de simulation et de l'environnement),
4. création de la recette pilotant le procédé réel, de celle pilotant le procédé référence et de celle utilisée par le filtre de Kalman étendu,
5. création de la recette générale par concaténation des recettes précédentes,
6. définition des constituants intervenant dans le procédé,
7. définition du modèle thermodynamique,
8. définition des réactions chimiques éventuelles,
9. sélection des appareils du procédé de référence et définition de leur configuration (au fur et à mesure qu'ils sont définis, les appareils et leur réseau de Petri sont ajoutés respectivement au flowsheet global et à la recette),
10. copie du flowsheet du procédé de référence en faisant intervenir des appareils potentiellement défaillants, afin de caractériser le procédé réel,
11. établissement des connexions,
12. description de la recette pilotant le procédé réel, de celle pilotant le procédé référence et de celle utilisée par le filtre de Kalman étendu,
13. création du module de surveillance,
14. création du solveur,
15. application de la configuration globale de la matière à tous les éléments du procédé,
16. initialisation des appareils (pression, température, composition, flux, etc.),
17. mise en place du marquage initial des RdP,
18. simulation,
19. destruction des objets créés.

CHAPITRE 8 L'EXPLOITATION DU SYSTÈME DE SURVEILLANCE

Ce chapitre a pour objectif de montrer comment un utilisateur peut, à partir des différents objets présents dans les bibliothèques qui lui sont proposées, concevoir et mettre en œuvre un système de surveillance de procédé. Dans ce cadre, trois applications sont développées :

- *La première concerne une opération de changement de solvant et avec l'apparition d'un défaut de température ; elle met l'accent sur la capacité à modéliser, simuler et détecter des défauts thermiques.*
- *La seconde est relative au benchmark SDH ; l'objectif est ici d'illustrer l'apparition de défauts multiples et la manière de les détecter et localiser.*
- *Enfin, la dernière concerne la surveillance d'un procédé batch ; cette application permet de montrer comment il est possible de gérer la complexité de plusieurs opérations unitaires.*

La démarche exposée au chapitre précédent a été appliquée à trois cas d'études :

- une opération de changement de solvant,
- un benchmark défini par l'*AS193 Diagnostic des systèmes hybrides* (www.univ-lille1.fr/lail/AS193),
- et une variante d'un procédé batch décrit dans [Joglekar *et al.*, 1985].

Pour chacun des cas, nous décrivons tout d'abord le procédé mis en œuvre, et le scénario que l'on souhaite simuler. Des éléments de description d'ordre général sont fournis. Ensuite, l'accent est mis sur la modélisation du procédé en termes de topologie, connexions, configuration, constituants, modèle thermodynamique. On distingue alors la modélisation du procédé de référence de celle du procédé réel. La recette à suivre est ensuite décrite selon le formalisme RdPDO. Pour finir, nous présentons pour chaque cas :

- les résultats de la simulation du modèle de référence,
- les résultats de la détection de l'occurrence d'une défaillance ; ici, sont illustrés les résultats de la reconstruction d'état par le filtre de Kalman étendu ;
- et les résultats du diagnostic.

À partir des valeurs des indicateurs de défauts répertoriés, la validation ou non de l'occurrence du/des défaut(s) est réalisée.

8.1 L'OPÉRATION DE CHANGEMENT DE SOLVANT : UN DÉFAUT DE TEMPÉRATURE

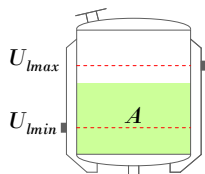
Cette étude se consacre sur l'étude de solvants. Un solvant est un liquide ayant la propriété de dissoudre, diluer ou extraire d'autres produits sans provoquer de modification chimique. Il ne réagit pas chimiquement avec le composé dissous : il est inerte. Un solvant permet donc de mettre en œuvre, d'appliquer, de nettoyer ou de séparer des produits.

Dans le chapitre 1, nous avons souligné qu'un produit passe par différentes étapes réactionnelles. Chacune de ces étapes peut utiliser un solvant différent choisi de façon optimale en fonction des critères économiques et/ou environnementaux. Dans de tels cas, une opération de changement de solvant est alors effectuée.

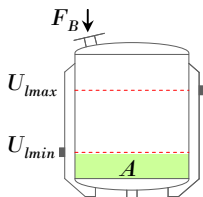
8.1.1 Le procédé mis en œuvre

Le procédé d'ajouts-évaporations fait partie des procédés industriels discontinus de changement de solvant généralement utilisés. Cette procédure industrielle standard de changement de solvant consiste en une succession d'évaporations et de chargements de solvant de substitution. Ce procédé fait l'objet de cette étude.

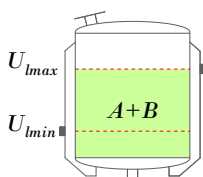
La substitution du solvant A par le solvant B s'effectue selon la procédure suivante :



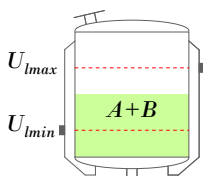
1. Considérons le cas où le volume initial en solvant de réaction (solvant A) est supérieur à la rétention minimum. La première opération consiste en la concentration du solvant A. La charge présente dans le réacteur est chauffée afin de l'amener à son point d'ébullition et la vaporiser en partie. Cette configuration est maintenue jusqu'à atteindre le volume minimum (U_{\min}).



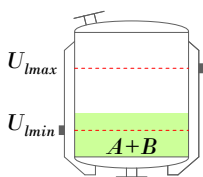
2. Le réacteur est ensuite alimenté par une alimentation continue. Une quantité de solvant de substitution (solvant B) est ainsi ajoutée au milieu réactionnel.



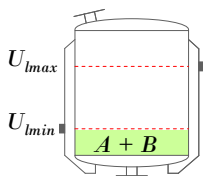
3. Deux évolutions sont envisageables à ce niveau :
 - Soit la pureté souhaitée du produit B est atteinte et, dans ce cas là, la séquence opératoire est terminée.
 - Soit l'alimentation est fermée lorsque la rétention maximum est atteinte (U_{\max}). La séquence opératoire se poursuit avec une nouvelle étape d'évaporation (étape 4).



4. Le milieu réactionnel est ensuite évaporé. La charge présente dans le réacteur est amenée à son point d'ébullition afin de la vaporiser en partie.



5. Deux évolutions sont envisageables à ce niveau :
 - Soit la pureté désirée du produit B est atteinte et, dans ce cas là, la séquence opératoire est terminée.
 - Soit l'évaporation dans le réacteur se poursuit jusqu'à ce que la rétention minimale soit atteinte (U_{\min}). La séquence se poursuit avec un refroidissement de la charge



6. Le refroidissement est maintenu jusqu'à ce qu'une température suffisamment basse et spécifiée au préalable soit atteinte. La séquence opératoire se poursuit avec une nouvelle étape d'ajout dans le réacteur du solvant de substitution (retour en 2).

Les opérations de chargements et d'évaporations se poursuivent de manière itérative jusqu'à l'obtention de la pureté finale désirée en solvant de substitution.

8.1.2 La modélisation

Le modèle mathématique de ce système n'est pas fourni, nous ne développons ici que les aspects modélisation du procédé sous *PrODHyS*.

8.1.2.1 La modélisation du procédé

8.1.2.1.1 Le procédé de référence

Le procédé d'ajouts-évaporations est modélisé comme indiqué sur la figure 8.1 Le schéma donne, pour chaque élément, son type et la classe d'objet qui le décrit.

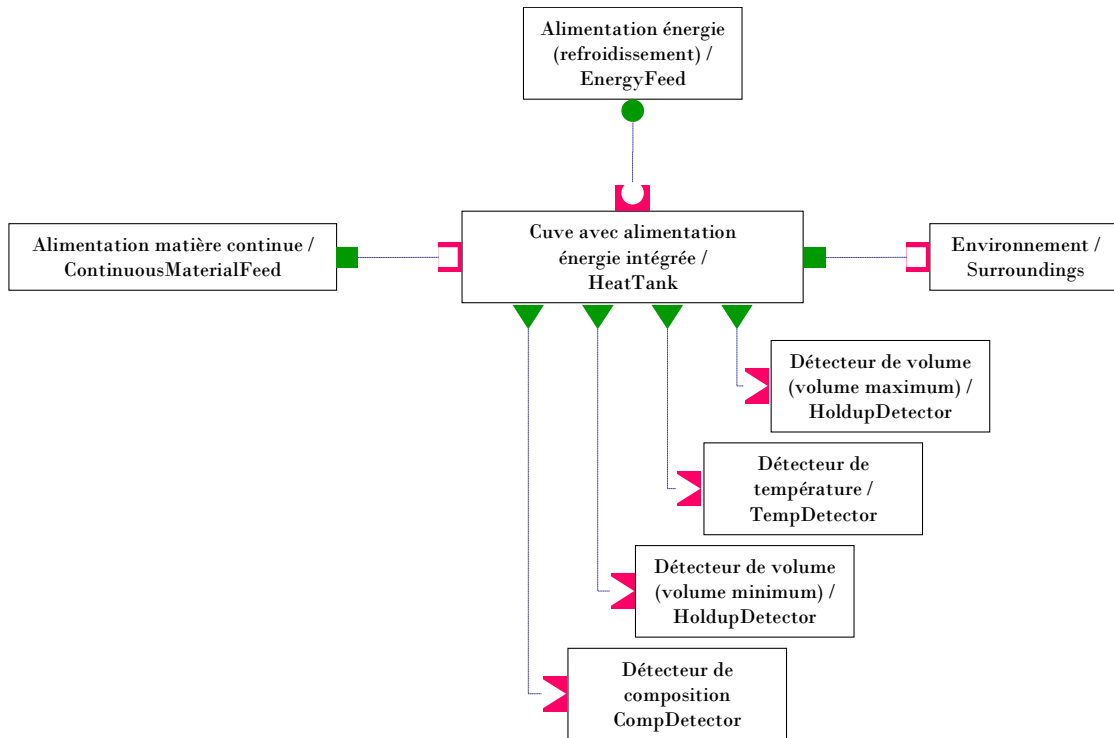


Figure 8.1 La modélisation du procédé de référence d'ajouts-évaporations

Il comporte :

- une cuve munie d'une alimentation énergie (de type `HeatTank`) ; elle représente le réacteur dans lequel se déroule la synthèse des produits ;
- une alimentation matière continue en solvant de substitution (de type `ContinuousMaterialFeed`) ; elle permet une introduction progressive de la charge ;
- une alimentation énergie (de type `EnergyFeed`) pour assurer le refroidissement du réacteur avant la coulée de solvant ;
- l'environnement (de type `Surroundings`) ;

- un ensemble de détecteurs (de type TempDetector, CompDetector, HoldupDetector) permettant de surveiller la température du milieu, la composition du mélange et d'indiquer un éventuel dépassement des volumes minimum et maximum autorisés.

8.1.2.1.2 Le procédé réel

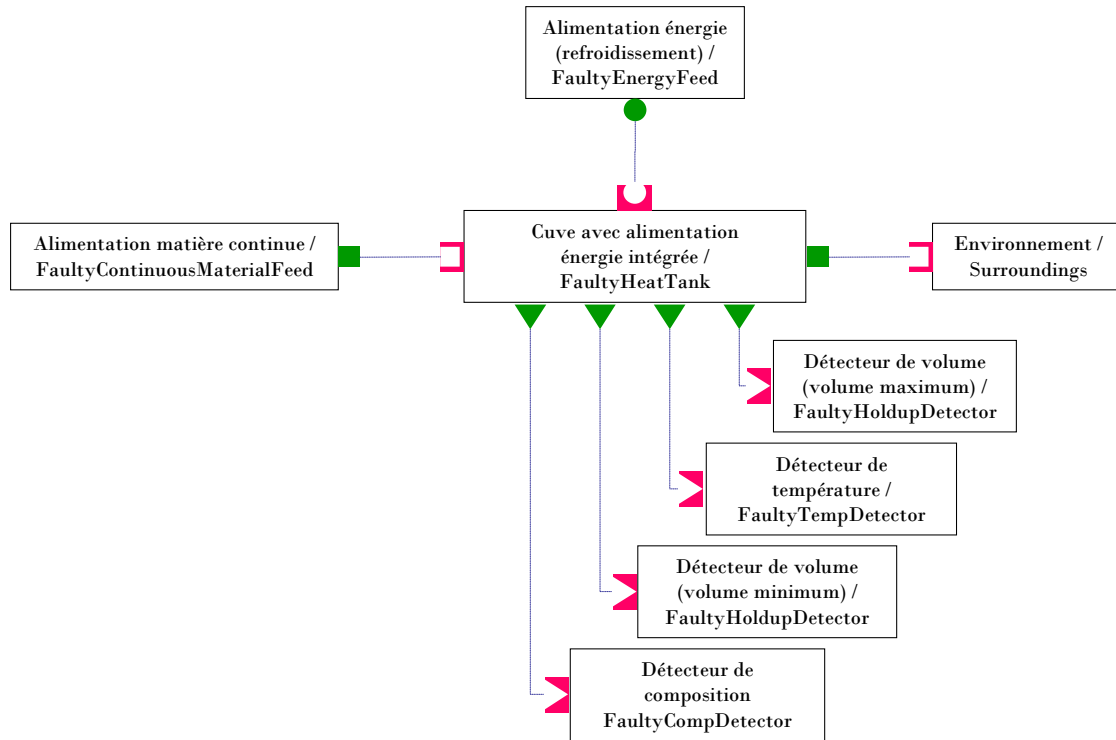


Figure 8.2 Le schéma bloc associé au système réel d'ajouts-évaporations

Le procédé réel possède le même flowsheet que le procédé de référence mais avec des appareils potentiellement défectueux. Le principe du procédé précédent est ainsi dupliqué et le modèle est composé alors de :

- une cuve munie d'une alimentation énergie (de type FaultyHeatTank) ; ici on suppose que seule son alimentation d'énergie peut être défectueuse (de type FaultyEnergyFeed).
- une alimentation matière continue en solvant de substitution (de type FaultyContinuousMaterialFeed) ;
- une alimentation énergie (de type FaultyEnergyFeed) pour assurer le refroidissement du réacteur avant la coulée de solvant ;
- l'environnement (de type Surroundings) ;
- un ensemble de détecteurs (de type FaultyTempDetector, FaultyCompDetector, FaultyHoldupDetector) dont le rôle est identique à ceux présents dans le procédé de référence.

Nous supposons que seuls l'environnement et l'appareil cuve sont deux entités ne pouvant ici être défectueuses.

La figure 8.2 illustre le processus de décomposition adopté pour représenter le procédé réel.

8.1.2.2 La modélisation de la recette

La recette associée au procédé de changement de solvant décrit une succession d'ajouts et d'évaporations conduisant à l'obtention de la pureté désirée en solvant de substitution. Le réseau de Petri qui lui est associé apparaît sur la figure 8.3. Initialement, le réacteur contient les produits et le solvant de réaction et la rétention liquide est comprise entre les valeurs minimale et maximale autorisées. Les alimentations matière et énergie sont fermées et les détecteurs, inactivés.

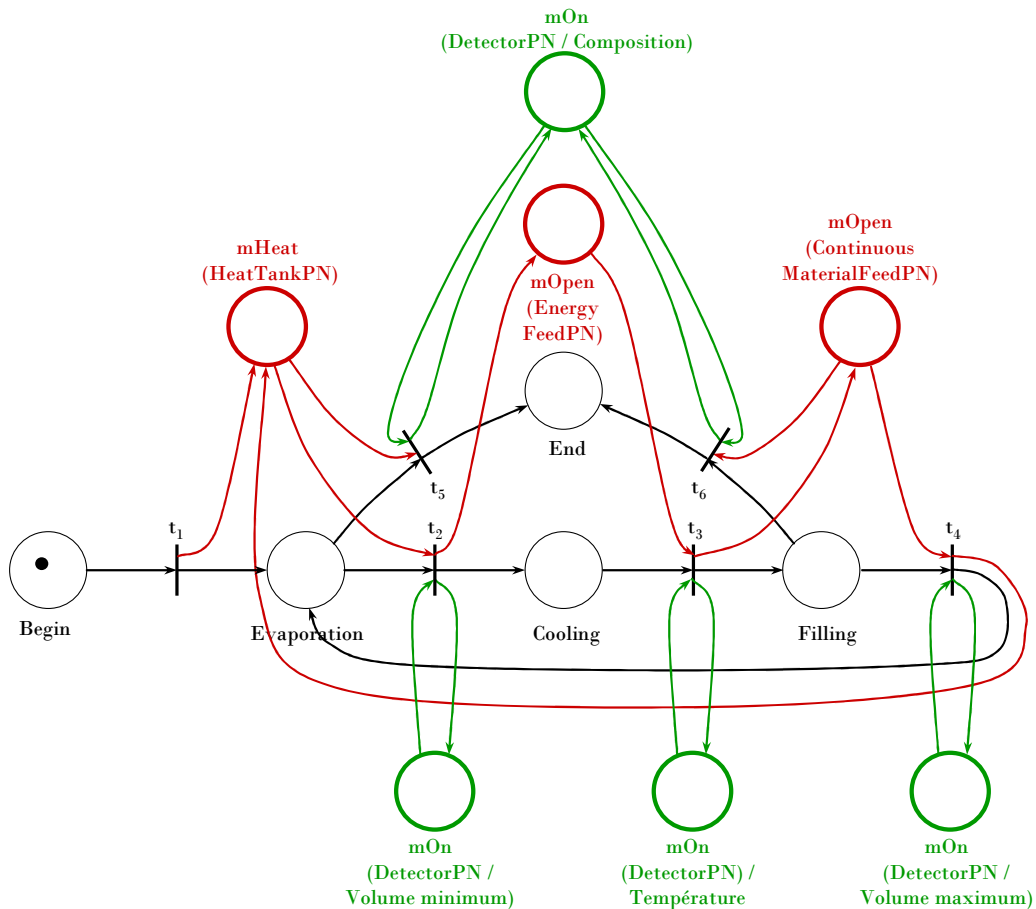


Figure 8.3 La recette décrivant le procédé d'ajouts-évaporations

Les principales étapes de cette opération sont les suivantes :

- Le marquage de la place *Begin* marque le début de l'opération de changement de solvant. Un jeton présent sur cette place sensibilise la transition t_1 située en aval qui est aussitôt tirée puisqu'elle ne comporte aucune condition.
- La rétention liquide étant supposée supérieure au volume minimum, la première phase de l'opération, représentée par la place *Evaporation*, correspond à l'étape 4 du scénario décrit dans le point 8.1.1.

Cette opération est pilotée par la recette qui émet un signal binaire sur la place de commande mHeat de l'alimentation énergie associée au réacteur.

- À partir de cette configuration, deux évolutions sont envisageables et correspondent à l'étape 5 du scénario décrit dans le point 8.1.1 :
 - Soit la fraction molaire du composant spécifié atteint la valeur cible. Cette information est transmise à la recette par le détecteur de composition qui émet un signal binaire sur sa place d'information mOn. Dans ce cas, la transition t_5 est tirée et conduit au marquage de la place End.
 - Soit le volume minimum est atteint. Cette information est transmise à la recette par le détecteur de volume minimum qui émet un signal binaire sur sa place d'information mOn. Dans ce cas, la transition t_2 est tirée et conduit au marquage de la place Cooling. Le refroidissement est piloté par la recette qui envoie un jeton binaire sur la place de commande mOpen de l'alimentation énergie dédiée au refroidissement de la charge.
- Le refroidissement est maintenu jusqu'à ce qu'une température suffisamment basse et spécifiée au préalable soit atteinte (cf. étape 6 du scénario décrit dans le point 8.1.1). Cet événement est transmis à la recette par le détecteur de température au niveau de la transition t_3 .
- L'étape suivante consiste alors à charger le réacteur en solvant de substitution (cf. étape 2 du scénario décrit dans le point 8.1.1). Elle est traduite par le marquage de la place Filling.
- À partir de cette configuration, deux évolutions sont possibles et décrites dans l'étape 3 du scénario décrit dans le point 8.1.1 :
 - Soit la pureté désirée est atteinte et l'opération de changement de solvant est terminée. Cet événement est détecté par le détecteur de composition et transmis à la recette au niveau de la transition t_6 .
 - Soit le volume maximum est atteint et la séquence opératoire se poursuit avec une nouvelle étape d'évaporation. Cet événement est détecté par le détecteur de volume maximal et transmis à la recette au niveau de la transition t_4 .

Le procédé réel (c'est-à-dire potentiellement défaillant) est piloté par la même recette que celle représentée sur la figure 8.3. Seuls les liens vers les appareils sont modifiés : la liaison ne s'effectue plus vers un appareil idéal mais vers un appareil potentiellement défectueux.

8.1.3 Les résultats

Nous avons modélisé ici la surveillance du procédé de changement de solvant. Un défaut du système de chauffe du réacteur (quantité d'énergie fournie inférieure à la normale) a été introduit. Les résultats de la surveillance font l'objet de ce point. Présentons tout d'abord ceux de la simulation en mode de fonctionnement normal.

8.1.3.1 La simulation du modèle de référence

L'exemple traité dans le cadre de cette application concerne le changement de solvant de l'eau (le plus lourd) par le méthanol (le plus léger).

Au début de l'opération, l'état initial du système est le suivant :

| | Cuve C1 | Alimentation |
|-----------------------|---------|--------------|
| T (K) | 298,15 | 298,15 |
| P (atm) | 1 | 1 |
| x_{eau} | 0,6 | 0,01 |
| $x_{\text{méthanol}}$ | 0,4 | 0,99 |
| U_1 (mol) | 300 | - |
| Débit (mol/min) | - | 5 |

Tableau 8.1 Les conditions opératoires du procédé d'ajouts-évaporations

Les volumes minimum et maximum fixés sont respectivement de 200 et 800 moles. Avant chaque coulée de solvant, les refroidissements intermédiaires amènent le réacteur à une température de 300,15K. La pression est supposée constante pendant toute la durée de l'opération. L'objectif est d'atteindre une fraction molaire en méthanol dans le réacteur de 0,95.

La simulation des phases successives sans défaillance est effectuée et les dynamiques résultantes sont illustrées sur les graphes suivants : ils représentent respectivement les évolutions au cours du temps de la composition liquide (cf. figure 8.4) et du volume réactionnel (cf. figure 8.5).

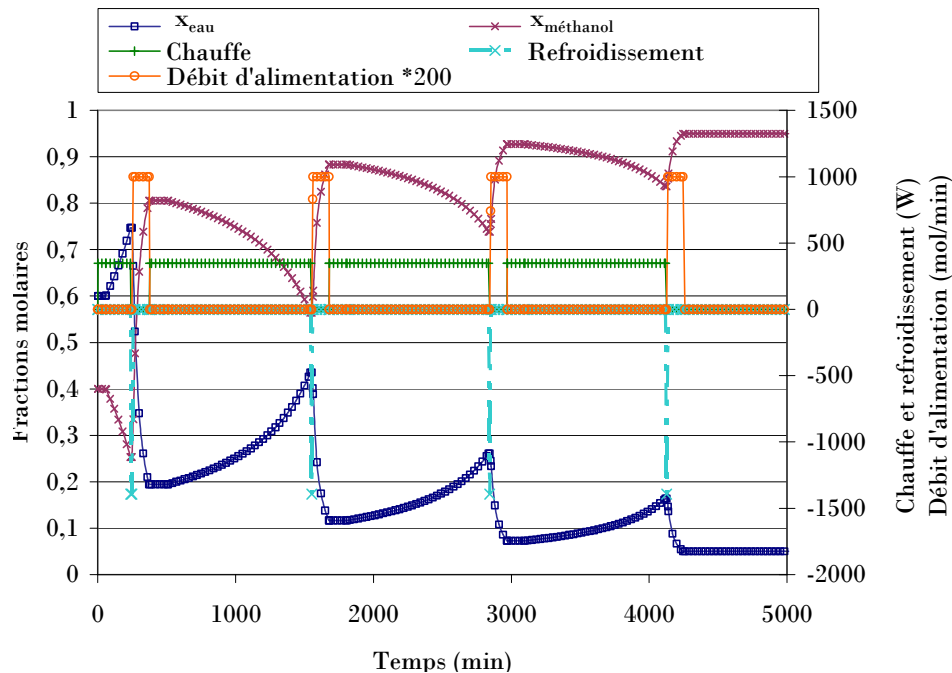


Figure 8.4 L'évolution de la composition liquide dans le réacteur

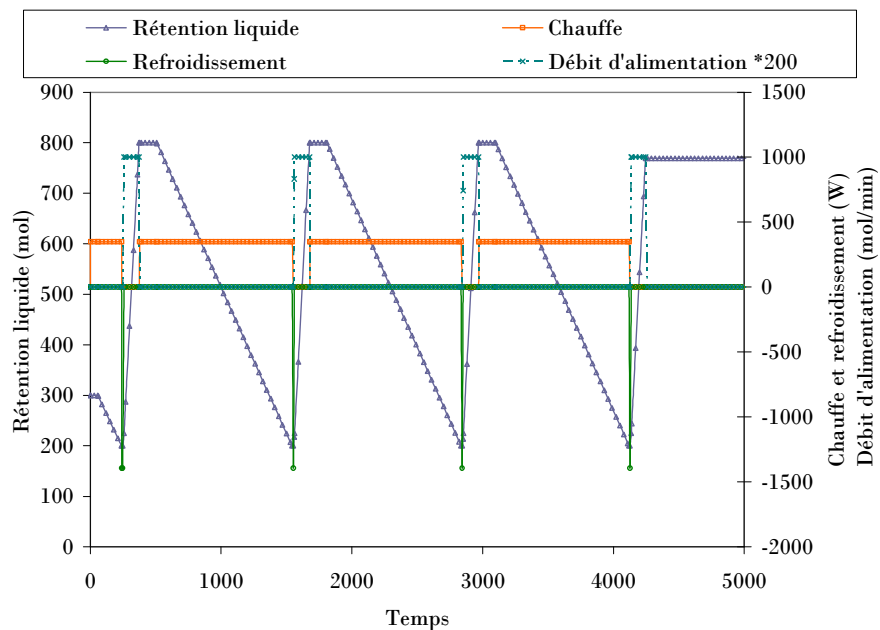


Figure 8.5 L'évolution de la rétention molaire liquide dans le réacteur

Au terme de cette première étude, deux points particuliers peuvent être soulignés. La fraction molaire finale en méthanol dans le réacteur est de 0,95 (cf. figure 8.4) et a été obtenue au cours d'une étape d'ajout de solvant. L'opération totale a nécessité quatre ajouts et quatre évaporations.

8.1.3.2 La détection

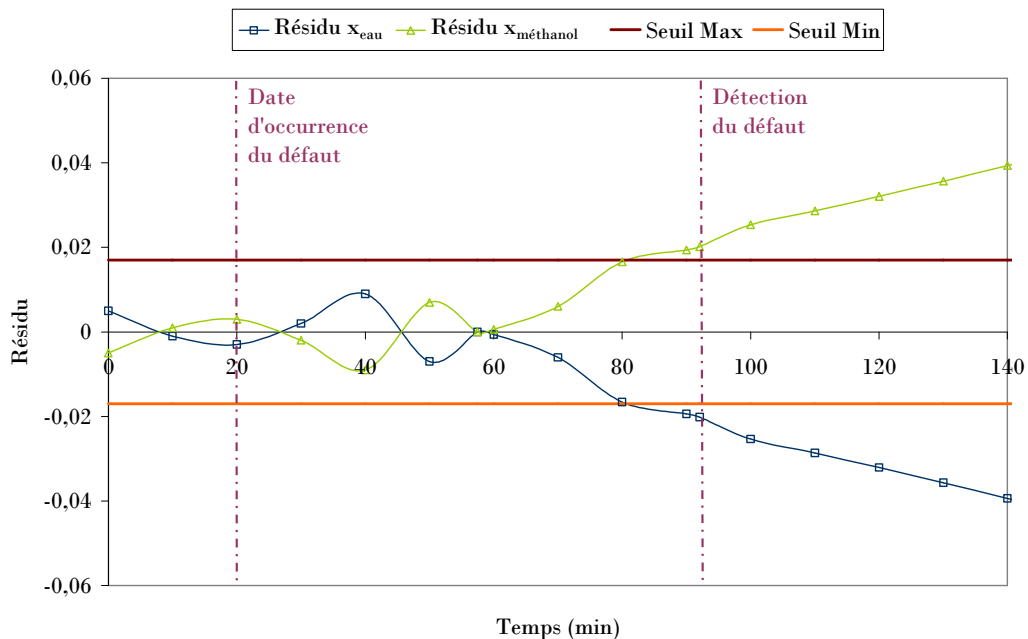


Figure 8.6 La détection du défaut « chauffe » lors du changement de solvant

Ce procédé est un système mettant en œuvre des phénomènes thermiques. Un défaut sur le système de la cuve thermique peut nuire à la réalisation de l'opération. Il est donc important de le détecter au plus tôt. Nous rappelons que les seuils définis pour la détection sont obtenus à partir de l'étude sur le réglage du filtre de Kalman (cf. les points 7.4.2.2 et 7.6.2).

À $t = 20$ min, un défaut sur la source d'énergie de chauffe du réacteur se produit. Cette dernière fournit une quantité de chaleur inférieure à la normale. La figure 8.6 illustre l'étape de détection. Les résidus associés respectivement aux compositions liquide de l'eau et du méthanol y sont représentés. À partir de $t = 80$ min, les deux résidus sortent de la zone de fonctionnement normal : le résidu associé à la composition liquide de l'eau dépasse le seuil minimum, tandis que celui associé à la composition liquide du méthanol dépasse le seuil maximum. Le diagnostic est lancé à $t = 95$ min.

On a considéré sept types de signatures chacune reliées à une grandeur physique :

- la signature s_1 à la température du réacteur,
- la signature s_2 à la composition molaire vapeur de l'eau,
- la signature s_3 à la composition molaire vapeur du méthanol,
- la signature s_4 à la composition molaire liquide de l'eau,
- la signature s_5 à la composition molaire liquide du méthanol,
- la signature s_6 à la rétention liquide du réacteur,
- et la signature s_7 à sa rétention vapeur.

On évalue alors le résidu et on obtient la signature de défaut instantanée correspondante :

| | Signature de défaut instantanée |
|-------|---------------------------------|
| s_1 | 0,0044098 |
| s_2 | 0,49367559 |
| s_3 | 0,50191462 |
| s_4 | 0 |
| s_5 | 0 |
| s_6 | 0 |
| s_7 | 0 |

Tableau 8.2 La signature de défauts instantanée

8.1.3.3 Le diagnostic

Les défauts considérés dans cette étude sont les suivants :

- Le défaut 1 correspond à un défaut du capteur de niveau haut qui détecte le niveau haut pour une valeur supérieure au niveau haut nominal.
- Le défaut 2 représente le défaut du capteur de niveau haut qui cette fois ci détecte le niveau haut pour une valeur inférieure au niveau haut nominal.

8.1 L'opération de changement de solvant : un défaut de température

- Le défaut 3 est un défaut du capteur de température qui détecte une température supérieure à la température nominale.
- Le défaut 4 symbolise un défaut du capteur de température détectant une température inférieure à la température nominale.
- Le défaut 5 correspond à un défaut de l'alimentation matière qui fournit de la matière à un débit dégradé.
- Le défaut 6 représente un défaut de la source d'énergie de chauffe du réacteur qui a une température inférieure à la normale.
- De même, le défaut 7 est un défaut de la source d'énergie de chauffe du réacteur qui ici fournit une quantité de chaleur inférieure à la normale.
- Le défaut 8 représente un défaut de l'alimentation d'énergie, utilisée pour le refroidissement du réacteur, qui a une température supérieure à la normale.
- Enfin, le défaut 9 désigne un défaut de l'alimentation d'énergie, utilisée pour le refroidissement du réacteur, qui ici fournit une quantité de chaleur inférieure à la normale.

Afin de construire la matrice d'incidence, nous avons considéré les trois états discrets du système correspondant aux places Evaporation, Cooling et Filling de la figure 8.3. Pour ces trois états nous avons effectué un certain nombre de simulations au cours desquelles les défauts exposés précédemment ont été générés.

On compare alors la signature instantanée de défauts (cf. tableau 8.2) à la matrice d'incidence en calculant les indicateurs de défauts relatifs aux distances de Manhattan relative (cf. définition 5-10) et améliorée (cf. définition 5-14). On obtient alors les indicateurs suivants :

| | Indicateur de Manhattan relatif | Indicateur de Manhattan amélioré |
|----------|---------------------------------|----------------------------------|
| Défaut 1 | 0,71428571 | 0,605 |
| Défaut 2 | 0,71554566 | 0,7254961 |
| Défaut 3 | 0,71428571 | 0,64 |
| Défaut 4 | 0,71554566 | 0,7104961 |
| Défaut 5 | 0,71714286 | 0,645 |
| Défaut 6 | 0,71428571 | 0,645 |
| Défaut 7 | 0,99819303 | 0,75330735 |
| Défaut 8 | 0,71554566 | 0,7104961 |
| Défaut 9 | 0,71428571 | 0,585 |

Tableau 8.3 Les indicateurs de défauts de l'exemple didactique

L'indicateur de Manhattan relatif détecte la présence du défaut 7 à 99,8%. Néanmoins, aucun défaut n'est réellement discriminé puisque leurs indicateurs sont tous supérieurs à 0,68. Par contre,

l'indicateur de Manhattan amélioré permet de discriminer les défauts 1, 3, 5, 6 et 9, puisque leurs indicateurs sont tous inférieurs à 0,68. Quatre possibilités sont proposées :

- Le défaut 7 reste le défaut le plus probable avec un indicateur de défaut de 75%.
- Ensuite, le défaut 2 est proposé avec un taux de 72%.
- Enfin, les défauts 4 et 8 sont également suspects avec un indicateur de défaut identique de 71%.

Cet exemple souligne l'importance d'utiliser les deux indicateurs de défaut utilisés dans ce travail pour pouvoir conclure. L'indicateur de Manhattan amélioré ne peut à lui tout seul nous permettre de conclure sur l'éventualité la plus probable. Néanmoins, en combinant les résultats de cet indicateur avec ceux de l'indicateur de Manhattan relatif, on peut statuer sur le fait que le défaut 7 reste la cause la plus probable de la défaillance, puisque son indicateur est le maximum. On a donc localisé le défaut sur le système thermique du réacteur. Le défaut a par ailleurs été identifié : le système thermique du réacteur fournit une quantité de chaleur inférieure à la normale.

8.2 LE BENCHMARK SDH : APPARITION DE DÉFAUTS MULTIPLES

Le système hydraulique (cf. figure 8.7) considéré est inspiré d'un benchmark défini par l'AS193 *Diagnostic des systèmes hybrides* (www.univ-lille1.fr/lail/AS193).

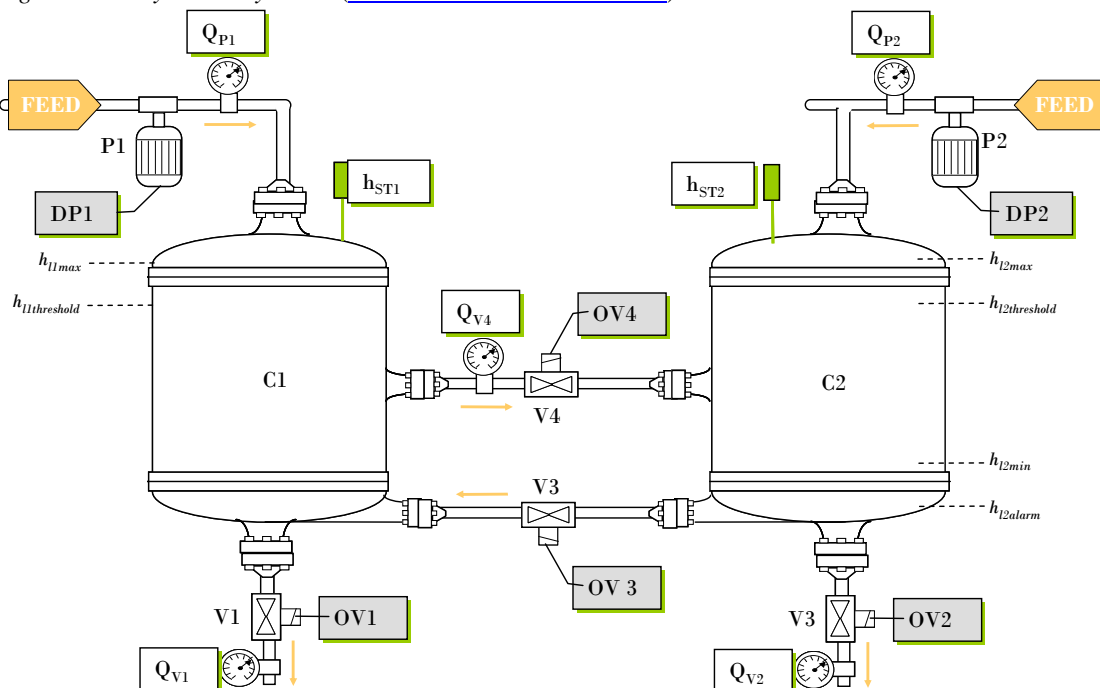


Figure 8.7 Le flowsheet du benchmark SDH

8.2.1 Le procédé mis en œuvre

Celui-ci est constitué de deux cuves cylindriques C1 et C2, reliés par deux conduites munies respectivement de vannes TOR V3 et V4. L'alimentation des cuves est réalisée par les pompes P1 et P2 pilotées en tout ou rien. Un soutirage au niveau de la cuve C2 est effectué grâce à la vanne TOR V2. De même, un soutirage au niveau de la cuve C1 peut être effectué au travers de la vanne TOR V1. Enfin, l'instrumentation du procédé comprend :

- cinq capteurs de débit : un mesurant le débit en sortie de chaque pompe, un mesurant le débit de sortie du soutirage de la cuve C2, un autre mesurant le débit de sortie du soutirage de la cuve C1 et un dernier mesurant le débit traversant la vanne V4.
- et deux capteurs de niveau : chacun mesurant le niveau de liquide d'une cuve.

Les deux capteurs de niveaux servent à la commande du système tandis que les cinq autres capteurs rendent le système totalement observable pour la surveillance du procédé.

Différents scénarii peuvent être simulés par action sur les pompes P1 et P2 et la vanne V2. Ici, la cuve C1 est alimentée cycliquement tandis que la cuve C2 est alimentée une seule fois. Enfin, un soutirage de la cuve C2 est effectué périodiquement.

8.2.2 La modélisation

Seuls les aspects « modélisation du procédé » sont développés ici.

8.2.2.1 La modélisation du procédé

8.2.2.1.1 Le procédé de référence

Le benchmark étudié est modélisé selon l'architecture représenté sur la figure 8.8. Le schéma donne, pour chaque élément, son type et la classe d'objet qui le décrit.

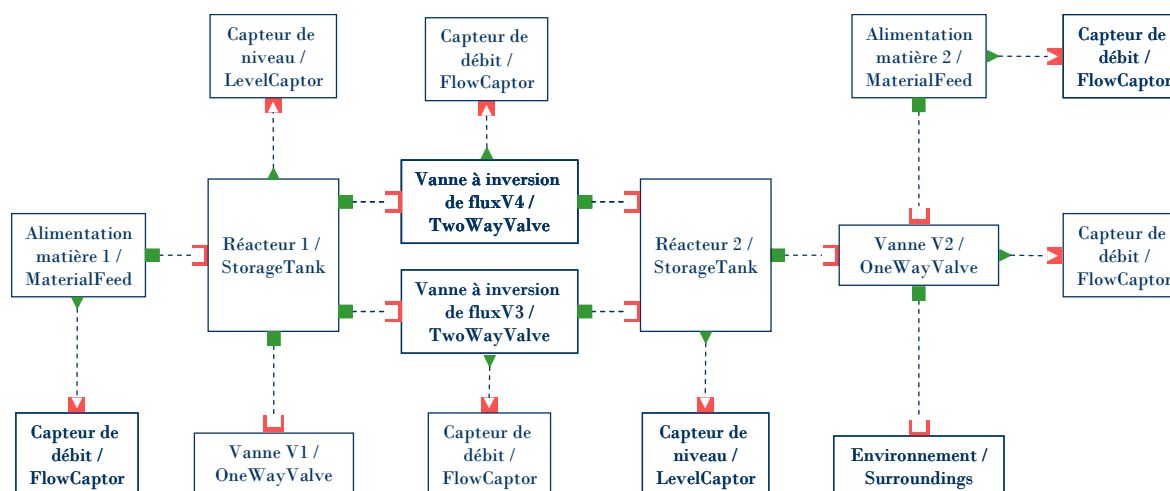


Figure 8.8 La modélisation du benchmark SDH

Il comporte :

- deux cuves (de type StorageTank) ;
- de deux alimentations de matière (de type MaterialFeed), une pour chaque cuve;
- deux vannes V_3 et V_4 (de type TwoWayValve), dans lesquelles les inversions de flux sont possibles ;
- deux vannes identiques V_1 et V_2 (de type OneWayValve) pour contrôler le soutirage des bacs C_1 et C_2 ;
- deux capteurs de niveau (de type LevelCaptor) indiquant respectivement le niveau de liquide dans la cuve C_1 et dans la cuve C_2 ;
- un ensemble de capteurs de débit (de type FlowCaptor) permettant de surveiller le système ;
- et l'environnement (de type Surroundings).

8.2.2.1.2 Le procédé réel

Le procédé réel possède le même flowsheet que le procédé de référence mais avec des appareils potentiellement défectueux. La figure 8.2 illustre le processus de décomposition adopté pour représenter le procédé réel. Le principe du procédé précédent est ainsi dupliqué et le modèle est composé alors de :

- deux cuves (de type StorageTank) qui sont supposées non défectueuses ;
- de deux alimentations matière (de type FaultyMaterialFeed) ;
- deux vannes V_3 et V_4 (de type FaultyTwoWayValve) autorisant les inversions de flux;
- deux vannes anti-retour V_1 et V_2 (de type FaultyOneWayValve) ;
- deux capteurs de niveau (de type FaultyLevelCaptor) ;
- un ensemble de capteurs de débit (de type FaultyFlowCaptor) ;
- et l'environnement (de type Surroundings), entité supposée non défectueuse.

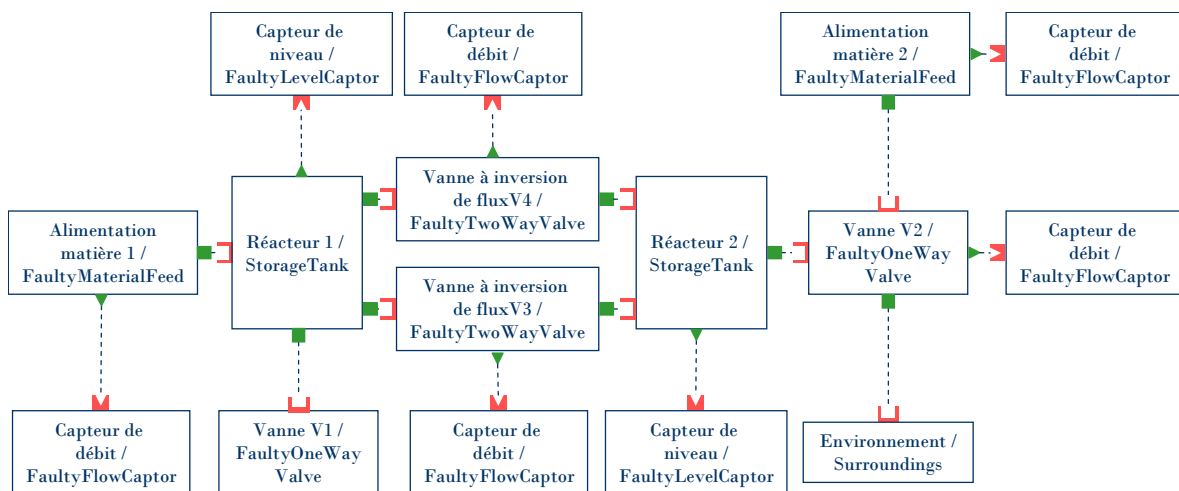


Figure 8.9 Le schéma bloc associé au système réel du benchmark SDH

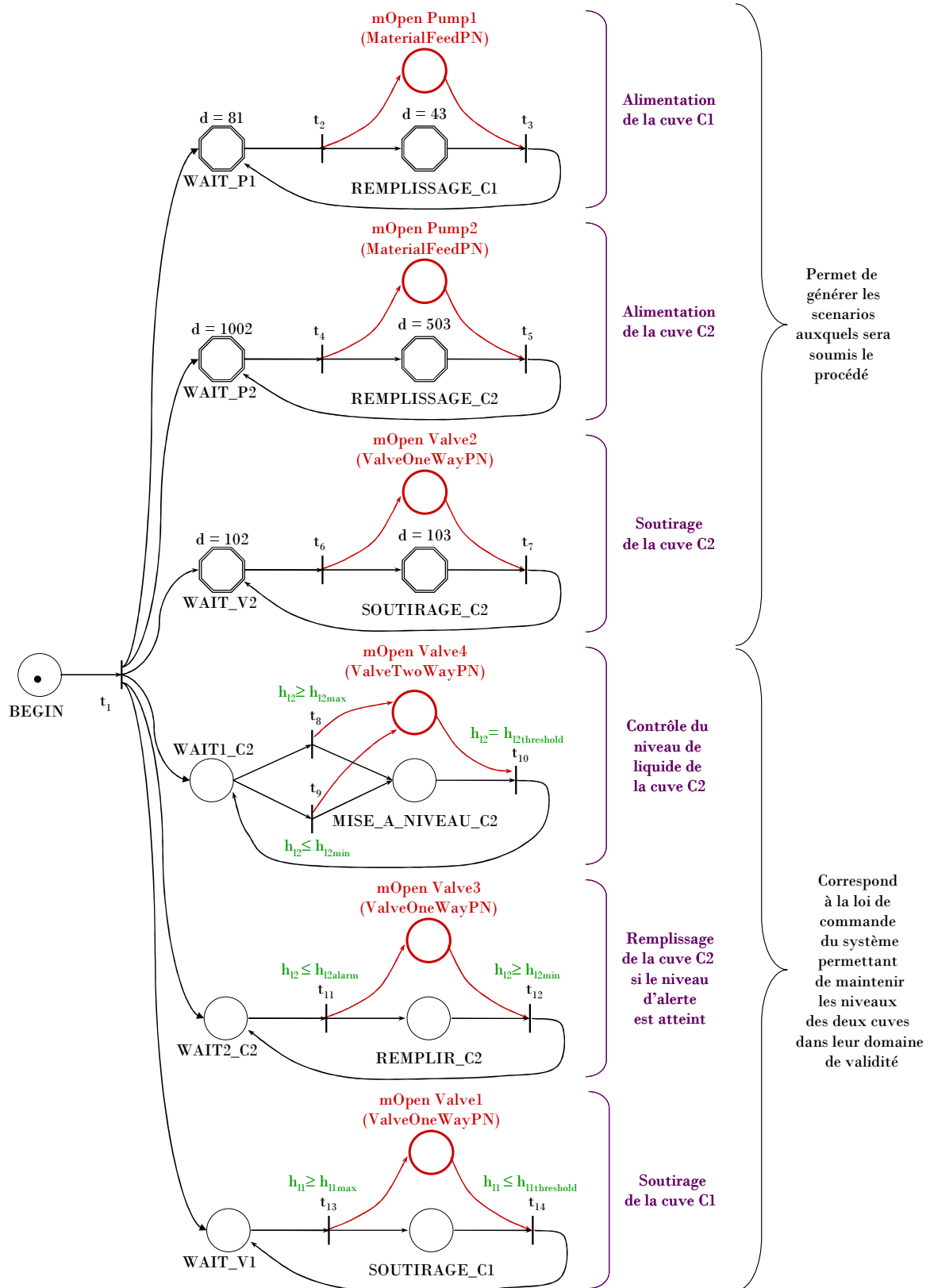


Figure 8.10 Le réseau de Petri de la recette du benchmark SDH

8.2.2.2 La modélisation de la recette

La recette associée au benchmark SDH est décrite par le réseau de Petri de la figure 8.10. Elle se décompose en deux parties principales.

- Les trois premières branches du RdP permettent de générer les scénarios auxquels est soumis le système :
 - La première consiste à remplir la cuve C1 de façon cyclique (un cycle correspond à la somme de la temporisation de la place WAIT_P1 et celle de la place REEMPLISSAGE_C1) pendant une durée coïncidant avec la durée de la temporisation de la place REEMPLISSAGE_C1.
 - La seconde partie a pour but également le remplissage d'une cuve, mais cette fois-ci de la cuve C2. Ce remplissage s'effectue de façon cyclique (un cycle correspond à la somme de la temporisation de la place WAIT_P21 et celle de la place REEMPLISSAGE_C2) pendant une durée coïncidant avec la durée de la temporisation de la place REEMPLISSAGE_C2.
 - Le soutirage de la cuve C2 s'opère grâce à l'action cyclique de la vanne V2 (un cycle correspond à la somme de la temporisation de la place WAIT_V3 et celle de la place SOUTIRAGE_C2) pendant une durée coïncidant avec la durée de la temporisation de la place SOUTIRAGE_C2.
- Les trois dernières mettent en œuvre la loi de commande :
 - Le système de commande a pour objectif principal de maintenir le niveau de liquide h_{12} dans la cuve C2 entre les hauteurs h_{12min} et h_{12max} par action sur la vanne V4.
 - De plus, la vanne V3 peut être ouverte pour éviter une rupture au niveau de C2 ($h_{12} \leq h_{12alarm}$).
 - Enfin, la vanne V1 est ouverte en cas de détection du niveau haut atteint dans la cuve C1 afin d'éviter un débordement de la cuve C1 ($h_{11} \geq h_{11max}$).

Le procédé réel (c'est-à-dire potentiellement défaillant) est piloté par la même recette que celle représentée sur la figure 8.10. Seuls les liens vers les appareils sont modifiés : la liaison ne s'effectue plus vers un appareil idéal mais vers un appareil potentiellement défectueux.

8.2.3 Les résultats

Nous avons modélisé ici la surveillance du benchmark SDH. Deux défauts simultanés ont été introduits : un défaut de capteur (capteur de niveau de la cuve C1) et un défaut d'actionneur (vanne de soutirage V1). Les résultats de la surveillance font l'objet de ce point. Présentons tout d'abord ceux du système en fonctionnement normal.

8.2.3.1 La simulation du modèle de référence

Nous avons effectué les simulations pour le jeu de paramètres listés dans le tableau 8.4. Le scénario étudié ici est exposé sur la figure 8.11. On y voit l'évolution au cours de la simulation du débit des pompes d'alimentation P1 et P2 ainsi que le débit de la vanne de soutirage V2 de la cuve C2.

8.2 Le benchmark SDH : apparition de défauts multiples

| | Cuve C1 | Cuve C2 | Alimentation 1 | Alimentation 2 |
|----------------------------|---------|---------|----------------|----------------|
| T (K) | 298,15 | 298,15 | 298,15 | 298,15 |
| P (atm) | 1 | 1 | 1 | 1 |
| U_1 (mol) | 400 | 180 | - | - |
| Débit (cm ³ /s) | - | - | 2 | 3 |

Tableau 8.4 Les conditions opératoires du benchmark SDH

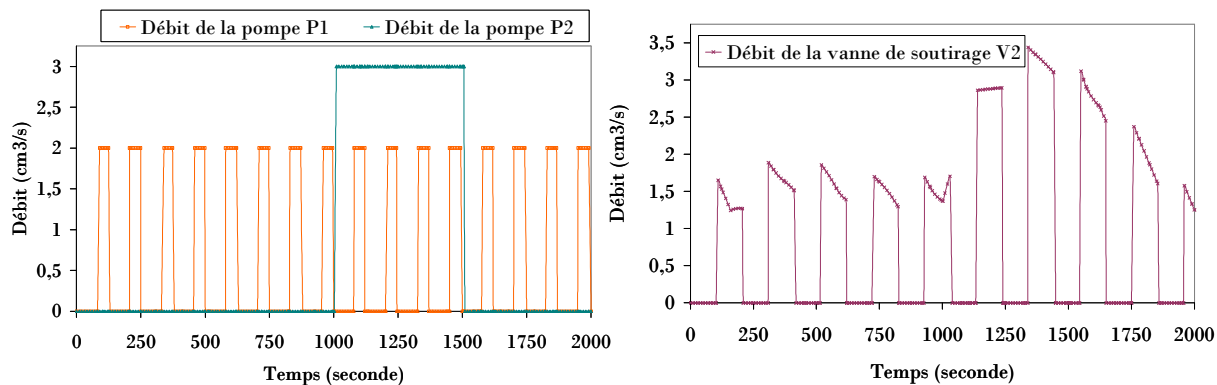


Figure 8.11 Le scénario étudié

Pour mieux comprendre ce scénario, le chronogramme d'état a été établi, illustrant ainsi l'enchaînement des étapes d'alimentation et de vidange.

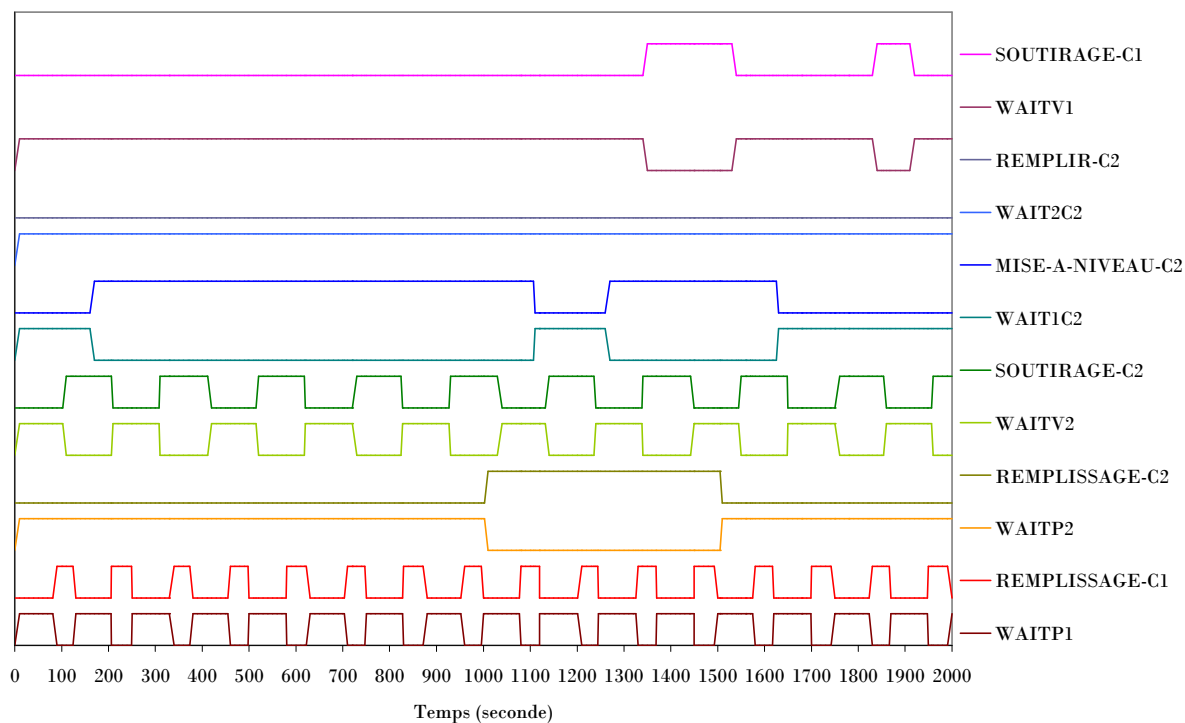


Figure 8.12 Le chronogramme d'état

Les résultats sont exposés sur les figures suivantes :

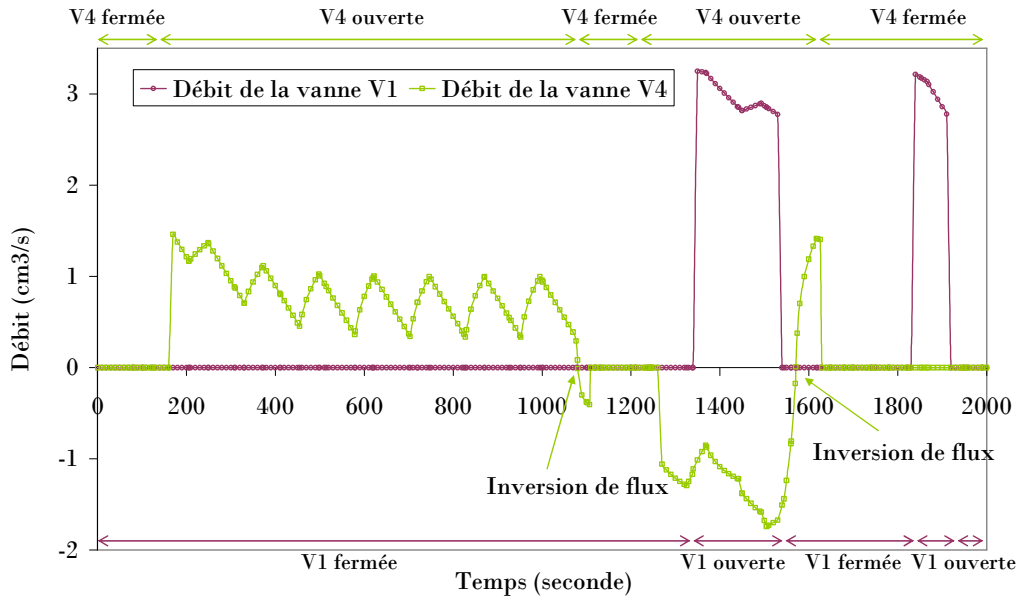


Figure 8.13 L'évolution des débits traversant les vannes V1 et V4 au cours du temps

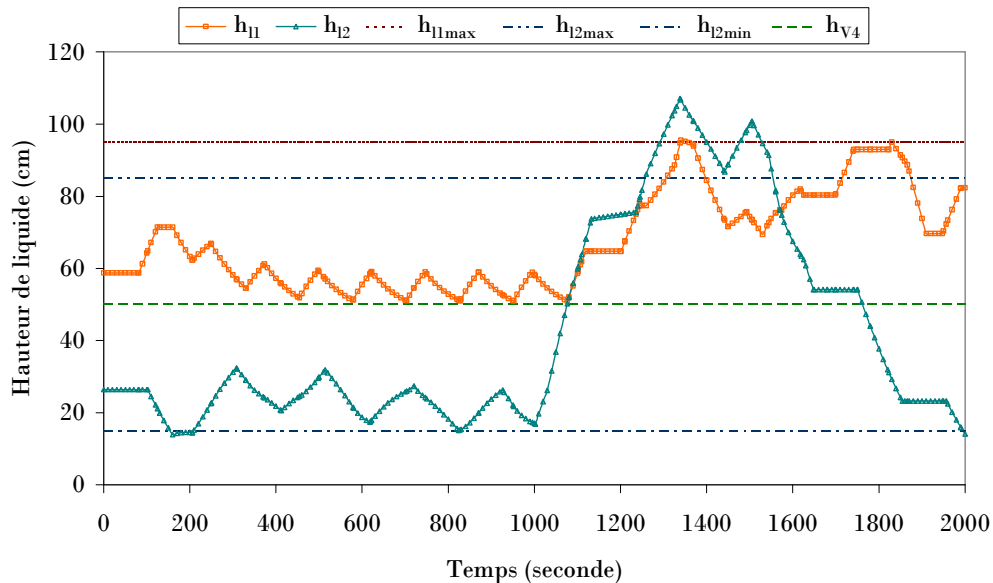


Figure 8.14 L'évolution des hauteurs de liquide des cuves C1 et C2 au cours du temps

À $t \approx 160$ s, la vanne V4 est ouverte car la hauteur de liquide dans la cuve C2 a atteint la valeur seuil h_{12min} . À $t \approx 1080$ s, on constate une inversion de flux ceci est dû au fait que la hauteur de liquide dans la cuve C2 initialement inférieure à la hauteur de liquide dans la cuve C1 devient supérieure à cette dernière hauteur. Le même phénomène est observé à $t \approx 1570$ s mais dans l'autre sens : la hauteur de liquide dans la cuve C1 devient supérieure à la hauteur de liquide dans la cuve C2. La vanne V4 est fermée à $t \approx 1100$ s, en effet la hauteur de liquide dans la cuve C2 a atteint la valeur seuil $h_{12threshold}$. Elle est ensuite rouverte à $t \approx$

1260 s puisque la hauteur de liquide dans la cuve C2 a atteint la valeur h_{2max} . Cette vanne est finalement fermée, quand la hauteur de liquide dans la cuve C2 a atteint valeur $h_{2threshold}$, soit à $t \approx 1260$ s.

Remarquons que la loi de commande mise en œuvre, volontairement rudimentaire, ne permet pas de garantir à tout instant ces objectifs car elle ne tient pas compte du niveau h_{1l} de la cuve C1.

La vanne V1 est ouverte à deux reprises (à $t \approx 1340$ s et à $t \approx 1830$ s), lorsque l'évènement suivant se produit : $h_{1l} \geq h_{1lmax}$. Elle est fermée quand la hauteur de liquide dans la cuve C1 devient inférieure ou égale à la valeur seuil $h_{1lthreshold}$. Notons que la loi de commande mise en œuvre, volontairement rudimentaire, ne permet pas de garantir à tout instant ces objectifs car elle ne tient pas compte du niveau h_{1l} de C1 (cf. figure 8.10).

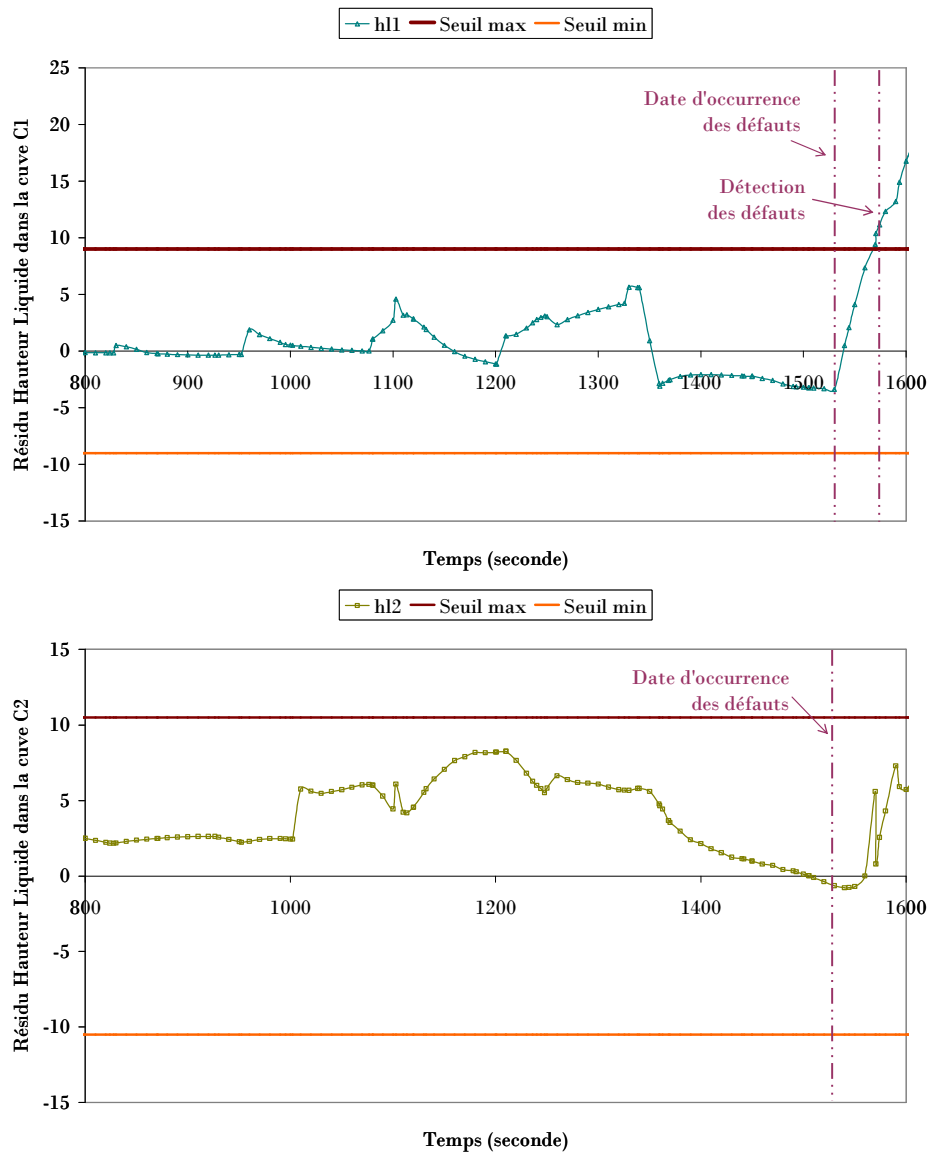


Figure 8.15 La détection des défauts sur le benchmark SDH

8.2.3.2 La détection

Ce procédé est un système fonctionnant sur des systèmes hydrauliques. Ici, nous allons étudier le diagnostic de défauts multiples. À $t \approx 1550$ s, deux défauts sur l'appareillage de la cuve C1 se produisent : la vanne de soutirage V1 reste bloquée en position ouverte et le capteur de niveau de la cuve C1 ne retourne aucune valeur.

On a considéré huit types de signatures chacune reliées à une grandeur physique :

- la signature s_1 au niveau du liquide dans la cuve C1,
- la signature s_2 au niveau du liquide dans la cuve C2,
- la signature s_3 au flux traversant la vanne V1,
- la signature s_4 au flux traversant la vanne V2,
- la signature s_5 au flux traversant la vanne V3,
- la signature s_6 au flux traversant la vanne V4,
- la signature s_7 au débit d'alimentation de la pompe P1,
- et la signature s_8 au débit d'alimentation de la pompe P2.

La figure 8.15 illustre l'étape de détection. Les résidus associés respectivement aux hauteurs de liquide dans les deux cuves y sont représentés. On constate que le résidu associé à la hauteur de liquide dans la cuve C2 reste dans le domaine de confiance. À l'inverse, à partir de $t \approx 1570$ s, le résidu associé à la hauteur de liquide dans la cuve C1 sort de la zone de fonctionnement normal. Le diagnostic est lancé à $t \approx 1580$ s. Notons que le système a une dynamique rapide par rapport à la période d'échantillonnage. C'est pourquoi les symptômes du défaut apparaissent rapidement.

On évalue alors le vecteur résidu et on obtient la signature de défaut instantanée correspondante :

| | Signature de défaut instantanée |
|-------|---------------------------------|
| s_1 | 0,01788035 |
| s_2 | 0 |
| s_3 | 0 |
| s_4 | 0 |
| s_5 | 0 |
| s_6 | 0,98211965 |
| s_7 | 0 |
| s_8 | 0 |

Tableau 8.5 La signature de défauts instantanée

8.2.3.3 Le diagnostic

Les défauts considérés dans cette étude sont les suivants :

- Le défaut 1 correspond à un défaut du capteur de niveau de la cuve C1 qui ne retourne aucune valeur.
- Le défaut 2 représente un défaut de la vanne V1 : elle reste bloquée en position ouverte.

- Le défaut 3 est également un défaut de la vanne V1 mais cette fois ci représente le fait que celle-ci reste bloquée en position fermée.
- Le défaut 4 symbolise un défaut de la vanne V4 : elle reste bloquée en position ouverte.
- De même, le défaut 5 concerne la vanne V4, mais quand celle-ci reste bloquée en position fermée.
- Le défaut 6 est un défaut d'actionneur : la vanne V2 qui reste bloquée en position ouverte.
- Le défaut 7 symbolise le défaut de la vanne V2 qui reste bloquée en position fermée.
- Le défaut 8 correspond à un défaut de la pompe d'alimentation P1 de la cuve C1 qui fournit en continu de la matière.
- À l'inverse, le défaut 9 représente le défaut de la pompe d'alimentation P1 de la cuve C1 qui ne fournit plus de matière.
- Le défaut 10 désigne un défaut de la pompe d'alimentation P2 de la cuve C2, qui ici reste bloquée en position ON.
- Enfin, le défaut 11 correspond à un défaut du capteur de niveau de la cuve C2, qui ne retourne plus aucune valeur, empêchant alors tout contrôle sur le niveau de liquide de la cuve C2.

La signature instantanée de défauts (tableau 8.5) est comparée à la matrice d'incidence construite par le calcul des indicateurs de défauts relatifs aux distances de Manhattan relative (cf. définition 5-10) et améliorée (cf. définition 5-14). Nous obtenons alors les indicateurs suivants :

| | Indicateur de Manhattan relatif | Indicateur de Manhattan amélioré |
|-----------|---------------------------------|----------------------------------|
| Défaut 1 | 0,99676174 | 0,99676174 |
| Défaut 2 | 0,99954261 | 0,99954261 |
| Défaut 3 | 0,76169712 | 0,76190627 |
| Défaut 4 | 0,75447009 | 0,7555121 |
| Défaut 5 | 0,86747521 | 0,8695757 |
| Défaut 6 | 0,81459252 | 0,81574746 |
| Défaut 7 | 0,78329091 | 0,78388617 |
| Défaut 8 | 0,99552991 | 0,87947009 |
| Défaut 9 | 0,77238858 | 0,7727889 |
| Défaut 10 | 0,77444467 | 0,71020235 |
| Défaut 11 | 0,75447009 | 0,87715592 |

Tableau 8.6 Les indicateurs de défauts de l'exemple didactique

Par l'étude des valeurs des indicateurs de Manhattan relatifs, trois défauts sont mis en avant : le défaut 1 et 2, avec des taux supérieurs à 99%. L'indicateur de Manhattan amélioré permet d'éliminer le défaut 8, puisque seuls les défauts 1 et 2 ont un indicateur de défaut supérieur à 99%. On peut donc conclure sur la présence de défauts multiples : les défauts 1 et 2.

Cet exemple met l'accent sur l'intérêt d'utiliser l'indicateur de Manhattan amélioré, qui permet de disculper le défaut 8 et de mettre en avant la présence de défauts multiples. La présence d'un défaut sur le capteur de niveau de la cuve C1 ainsi que le défaut sur la vanne de soutirage V1 ont été prouvés.

Notons que le système de diagnostic est capable de détecter et localiser aussi bien les défauts d'actionneurs que de capteurs.

8.3 LA SURVEILLANCE D'UN PROCÉDÉ COMPLEXE

Dans ce dernier exemple d'application, l'étude porte sur une variante d'un procédé décrit dans [Joglekar et al, 1985].

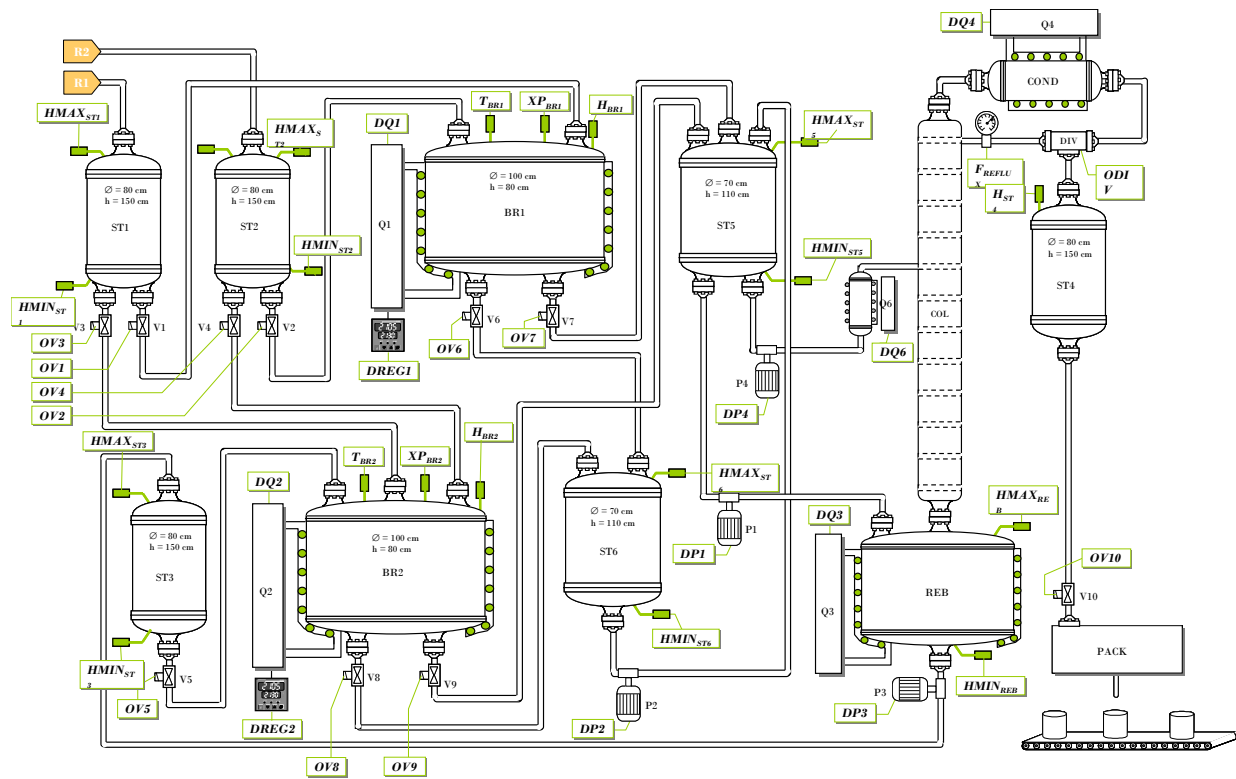


Figure 8.16 Le schéma du procédé [Joglekar et al., 1985]

8.3.1 Le procédé mis en œuvre

Le procédé mis en œuvre est décrit sur la figure 8.16. Le but de cette installation est de fabriquer et de conditionner un produit P dont la pureté doit être au moins égale à 98% molaire à partir de deux réactifs R1 et R2. La réaction considérée est une réaction équilibrée endothermique impliquant deux réactifs R1 et R2 et un produit P :



Elle suit une loi cinétique d'ordre 2 de la forme :

$$r_R = k_1 \cdot e^{-\frac{Ea_1}{RT}} \cdot x_{R1} \cdot x_{R2} - k_{-1} \cdot e^{-\frac{Ea_{-1}}{RT}} \cdot x_P \quad (8 - 2)$$

Remarque : Une mauvaise connaissance de la cinétique pourrait être prise en compte en utilisant une loi cinétique différente entre le procédé réel et le procédé de référence. N'étant pas l'objet de cette étude, ici, la loi cinétique est la même mais avec des paramètres cinétiques différents.

Comme le montre la figure 8.17, la fraction molaire de produit P ne peut excéder 0,83 et un temps de séjour infini serait nécessaire pour atteindre cette composition. Afin de maximiser le taux de conversion de la réaction sans pour autant pénaliser le temps de cycle du procédé, la réaction est stoppée dès que la composition en produit P atteint la valeur de 0,8. Par ailleurs, compte tenu de la loi de vitesse considérée, la réaction (R) est d'autant plus rapide que la température T est élevée. La température retenue pour la réaction doit garantir une réaction rapide tout en maintenant les constituants dans l'état liquide. Une température de 383 K permet de satisfaire ces deux contraintes. La spécification de pureté imposée pour le produit P étant égale à 98%, une étape de distillation est donc nécessaire.

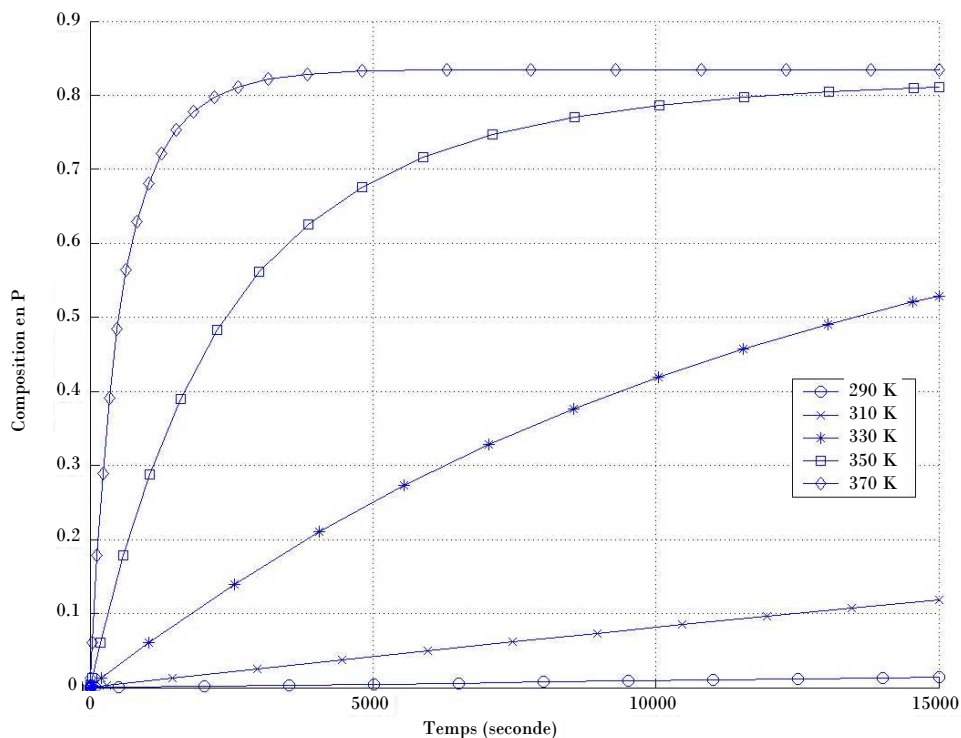


Figure 8.17 La vitesse de la réaction en fonction de la température

Celui-ci combine des opérations discontinues (synthèse, conditionnement) et des opérations continues (séparation). Chaque opération intègre les phases de transfert amont et aval avec les cuves tampon (par exemple l'opération de synthèse intègre les transferts depuis ST1 et ST2 en amont et ST5, ST6 en aval). Lorsque plusieurs cuves sont affectées au stockage, une opération spécifique de gestion des cuves de stockages doit être définie (dans notre cas, politique de gestion de ST5 et ST6).

8.3.2 La procédure

La recette générale du produit à fabriquer est représentée sur la figure 8.18 sous la forme d'un S.T.N. (State Tast Network).

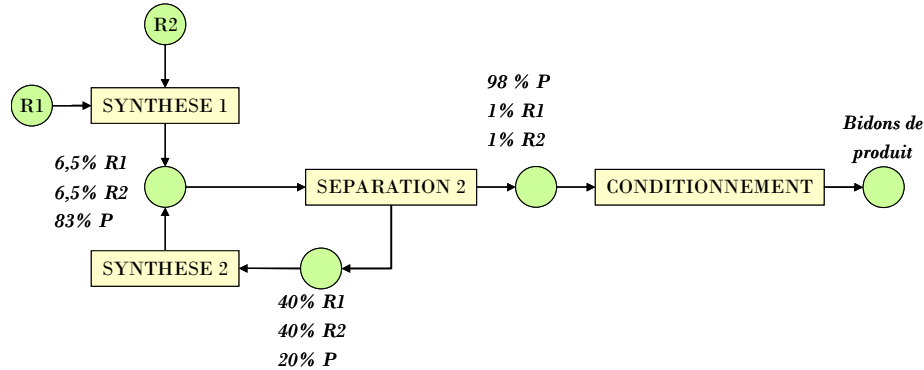


Figure 8.18 La recette de fabrication du produit P

Cinq opérations caractérisent la recette générale :

La fabrication du produit P selon la réaction (R) dans le réacteur BR1 ou BR2.

Cette opération comporte les phases suivantes :

- le chargement du réacteur avec $n/2$ moles de produit R1,
- le préchauffage à 383 K,
- le chargement du réacteur de $n/2$ moles de produit R2 tout en réglant la température à 383 K,
- la réaction jusqu'à ce que la composition en produit P atteigne la valeur de 0,8,
- et la vidange totale ou partielle du mélange réactionnel.

Lorsque cette opération est effectuée dans le réacteur BR2, les réactifs peuvent aussi provenir du bac de stockage ST3 qui contient un mélange de produit R1 et R2 proche des proportions stœchiométriques issu du recyclage provenant de l'étape de séparation.

Le stockage du mélange après réaction dans ST5 ou ST6 selon leur disponibilité

La purification du produit avec le séparateur SEP

Cette opération se divise en plusieurs phases :

- le chargement du rebouilleur avec une partie du contenu de la cuve ST5.
- le remplissage et démarrage de la colonne
- la mise en régime de la colonne comprenant le transfert en continu le mélange du réservoir ST5 sur le plateau d'alimentation de la colonne à un débit constant (pompe P4), ce qui provoque le remplissage du rebouilleur et la réalisation de l'opération de distillation à reflux infini pendant 1h.

- la marche à reflux fini qui consiste à ouvrir le diviseur de courant VPA1 afin de réaliser l'opération de distillation à reflux fini ($r=1,2$). Le produit P quasiment pur est récupéré dans le réservoir ST4. Le mélange de réactifs résiduels R1 et R2 est recyclé dans le réservoir ST3.

Le stockage des réactifs n'ayant pas réagi dans le réservoir ST3

Ceux-ci sont recyclés dans le procédé et peuvent servir de matières premières à la réaction mise en œuvre dans le réacteur BR2.

Le conditionnement du produit P

Le produit P, stocké dans le réservoir ST4 est ensuite conditionné dans des bidons de 10 litres.

8.3.3 La modélisation

Seuls les aspects « modélisation du procédé » sont développés ici.

8.3.3.1 La modélisation du procédé

8.3.3.1.1 Le procédé de référence

Le procédé est modélisé selon l'architecture représenté sur la figure 8.19. Le schéma donne, pour chaque élément, son type et la classe d'objet qui le décrit. Le procédé comporte :

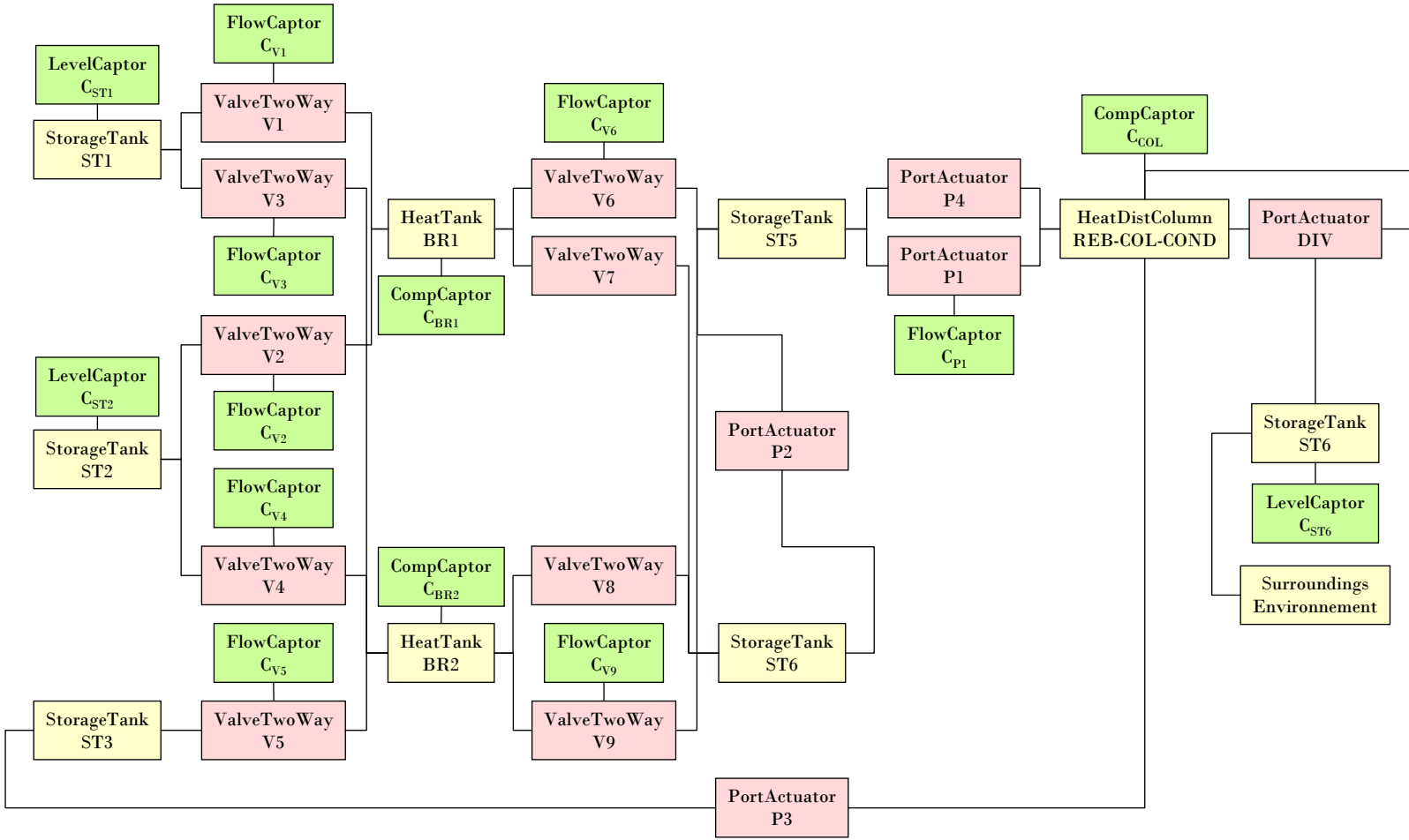
- six cuves ST_i de $i=1..6$ (de type StorageTank) ;
- deux réacteurs (de type HeatTank) ;
- dix vannes V_i de $i=1..10$ (de type TwoWayValve), dans lesquelles les inversions de flux sont possibles ;
- une colonne à distiller (de type HeatDistColumn) ;
- trois pompes (de type PortActuator) ;
- un diviseur de courant (de type PortActuator) ;
- un ensemble de capteurs de débit (de type FlowCaptor), de composition (de type CompCaptor) et de niveau (de type LevelCaptor) permettant de surveiller le système ;
- et l'environnement (de type Surroundings).

8.3.3.1.2 Le procédé réel

Nous supposons ici que seuls les capteurs et les vannes peuvent être défectueux. Le principe du procédé précédent est ainsi dupliqué et le modèle est composé alors de :

- six cuves ST_i de $i=1..6$ (de type StorageTank) ;
- deux réacteurs (de type FaultyHeatTank) ; ici on suppose que seule son alimentation d'énergie peut être défectueuse (de type FaultyEnergyFeed) ;
- dix vannes V_i de $i=1..10$ (de type FaultyTwoWayValve) ;
- une colonne à distiller (de type HeatDistColumn) ;

Figure 8.19 La modélisation du procédé [Joglekar et al., 1985]



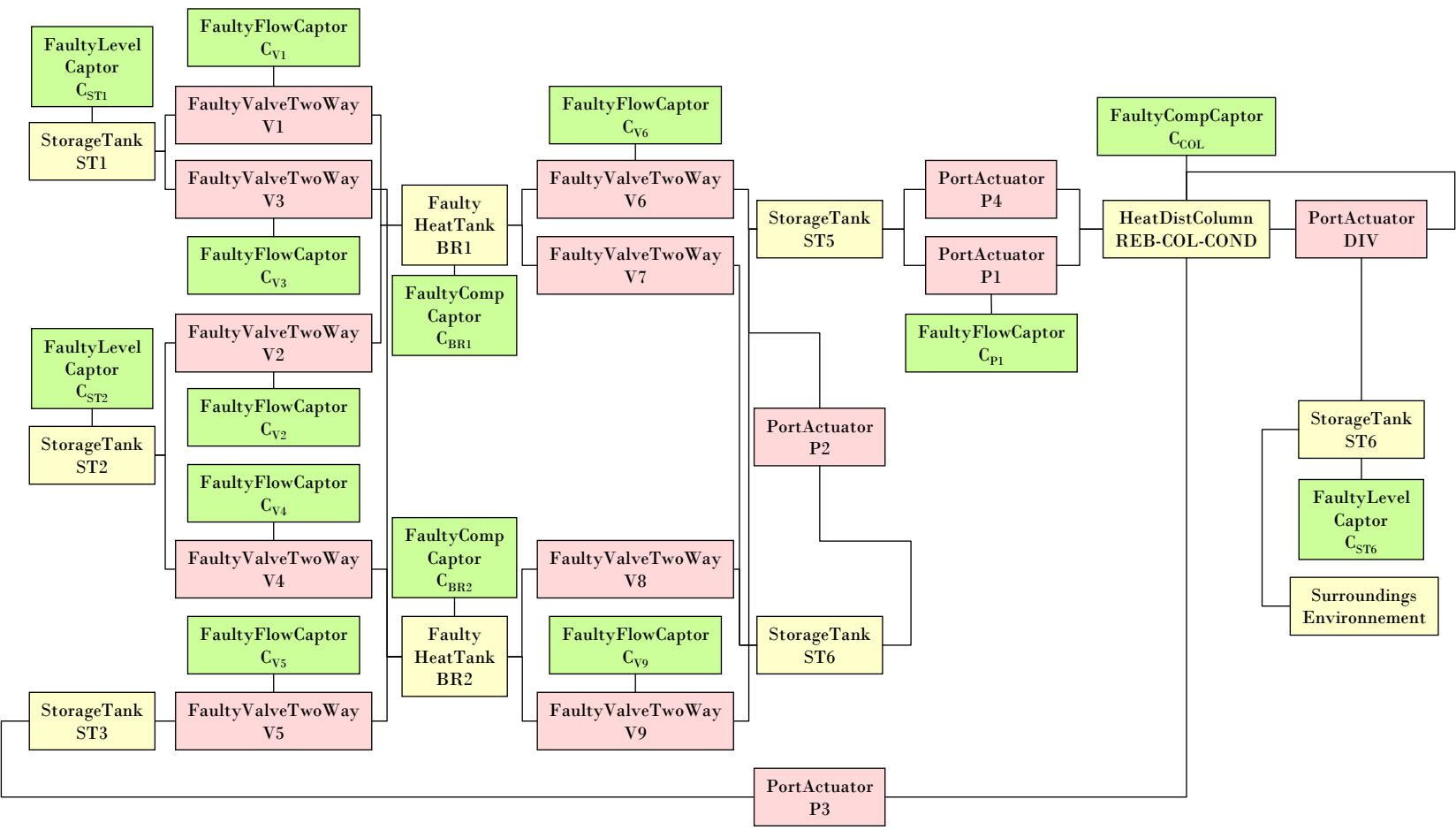


Figure 8.20 Le schema bloc associe au système réel du procédé [Joglekar et al., 1985]

- trois pompes (de type PortActuator) ;
- un diviseur de courant (de type PortActuator) ;
- un ensemble de capteurs de débit (de type FaultyFlowCaptor), de composition (de type FaultyCompCaptor) et de niveau (de type FaultyLevelCaptor) ;
- et l'environnement (de type Surroundings).

8.3.3.2 La modélisation de la recette

Seule la modélisation de la partie commande du procédé ainsi que sa connexion avec la partie opérative est détaillée. La décomposition de la procédure selon les différents niveaux décrits précédemment donne ainsi lieu à la définition de différents réseaux de Petri.

8.3.3.2.1 Le niveau procédure

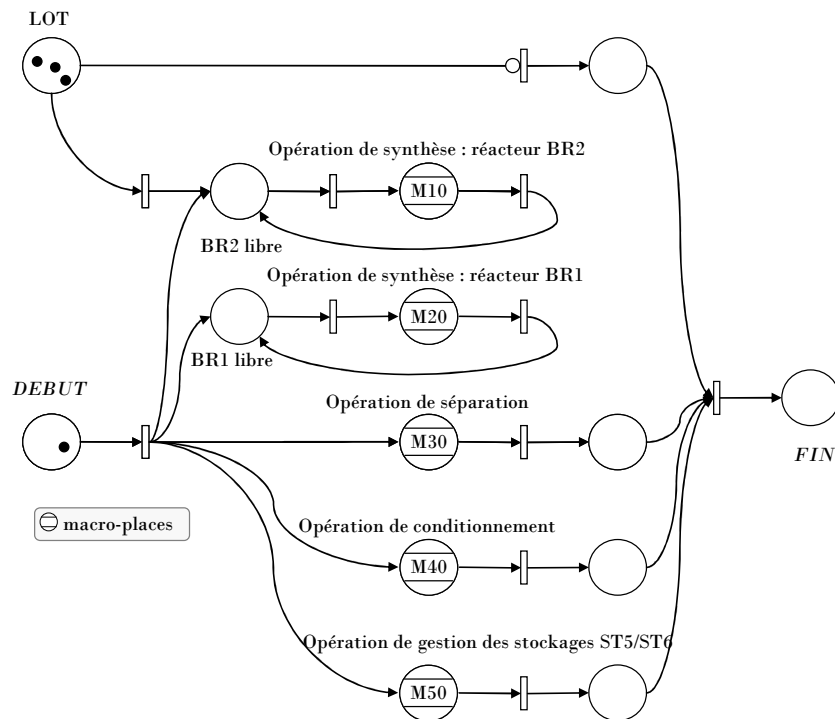


Figure 8.21 Le réseau de Petri du niveau Procédure

Dans le cas qui nous intéresse, la procédure (cf. figure 8.21) est constituée de cinq opérations représentées par des macro-places : synthèse dans BR1, synthèse dans BR2, séparation, conditionnement et gestion des cuves de stockage ST5 et ST6. Dans ce cadre, le lien prévu vers un objet de type MasterRecipe trouve ici sa justification car l'établissement de la procédure requiert la connaissance de l'ensemble des informations utiles pour la fabrication du produit (gammes de fabrication, équipements nécessaires).

8.3.3.2.2 Le niveau opération

Une opération définit un ensemble de phases exécutées en séquence ou en parallèle dans une unité de production. Ce niveau de description de la recette rejoint la notion d'opération unitaire couramment rencontrée en génie des procédés. La figure 8.22, qui représente l'opération de séparation permet de représenter l'étape 3 de la procédure. Elle se compose notamment d'une phase composée de distillation représentée par une macro-place, notée M300 et de phases élémentaires (ne comportant qu'un seul pas) représentées par une place (place nommée FILL_REB par exemple).

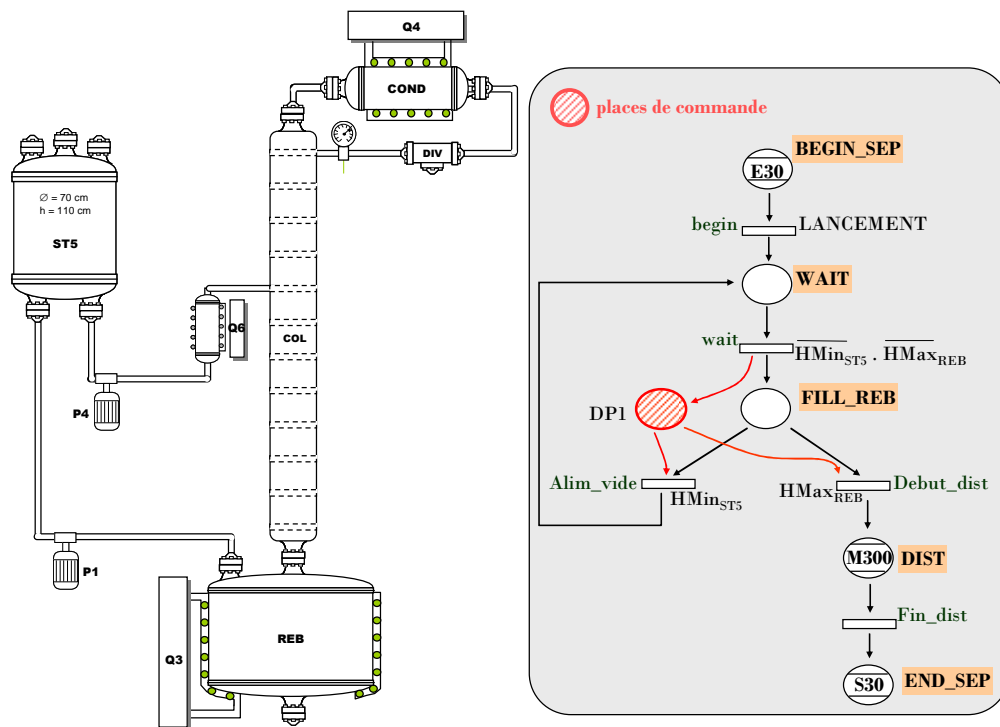


Figure 8.22 Le réseau de Petri du niveau Opération : opération séparation

8.3.3.2.3 Le niveau phase

Une phase est une succession de pas permettant de réaliser une fonction élémentaire. La figure 8.23 représente notamment le RdP de la phase de distillation (macro-place M300 sur la figure 8.22). Le RdP permettant de représenter le fonctionnement intrinsèque de la colonne est aussi représenté. Comme nous pouvons le constater les évolutions respectives de ces deux RdP sont intimement liées. Ainsi l'évolution du RdP de la phase active des places de commande permettant l'évolution du RdP intrinsèque de la colonne. Inversement, l'évolution du RdP intrinsèque de la colonne active des places d'information qui permettent l'évolution du RdP de la phase de distillation. Précisons par ailleurs, qu'au niveau de la phase, les RdP sont définis de manière générique afin de permettre leur réutilisation dans un autre contexte.

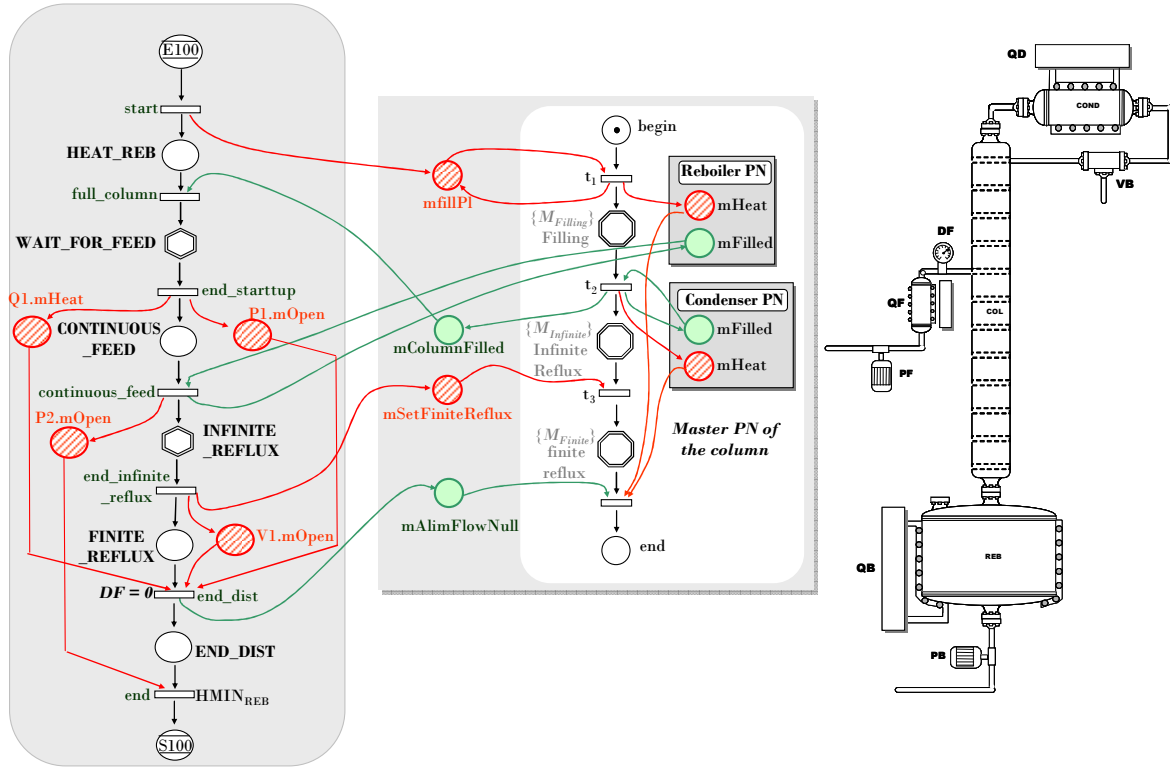


Figure 8.23 Le réseau de Petri du niveau Phase : phase distillation

8.3.4 Les résultats

8.3.4.1 La simulation du modèle de référence

Les modèles mis en œuvre dans cette simulation prennent en compte, entre autres, les bilans matière globaux et partiels, le bilan d'énergie, les équilibres liquide/vapeur, les vitesses de réaction, ainsi que les phénomènes hydrauliques. En effet, hormis les conduites munies d'une pompe, les transferts entre cuves sont effectués par gravité. Cela induit que les débits de sortie des cuves sont fonction de la pression hydraulique et donc du niveau de liquide dans les cuves source. Les durées de transfert dépendent donc de l'état du système à un instant donné.

Les principales contraintes se situent au niveau de la colonne. Le démarrage de la colonne nécessite du temps ainsi qu'une quantité importante d'énergie. Afin de ne pas pénaliser le temps de cycle et de minimiser la consommation d'énergie, il apparaît alors indispensable de garantir un flux continu de matière en sortie de cuve ST5 (seule cuve à pouvoir alimenter la colonne), une fois la colonne remplie. Ceci engendre aussi un débit continu de matière en sortie du rebouilleur et en sortie du diviseur de courant. Les réservoirs ST3 et ST4 font donc office de cuve tampon. Par ailleurs, le contenu de ST1 doit être suffisant pour alimenter sans rupture le conditionneur. À titre d'illustration, la figure 8.24 donne une partie de l'évolution de composition au sein du réacteur durant l'opération de réaction.

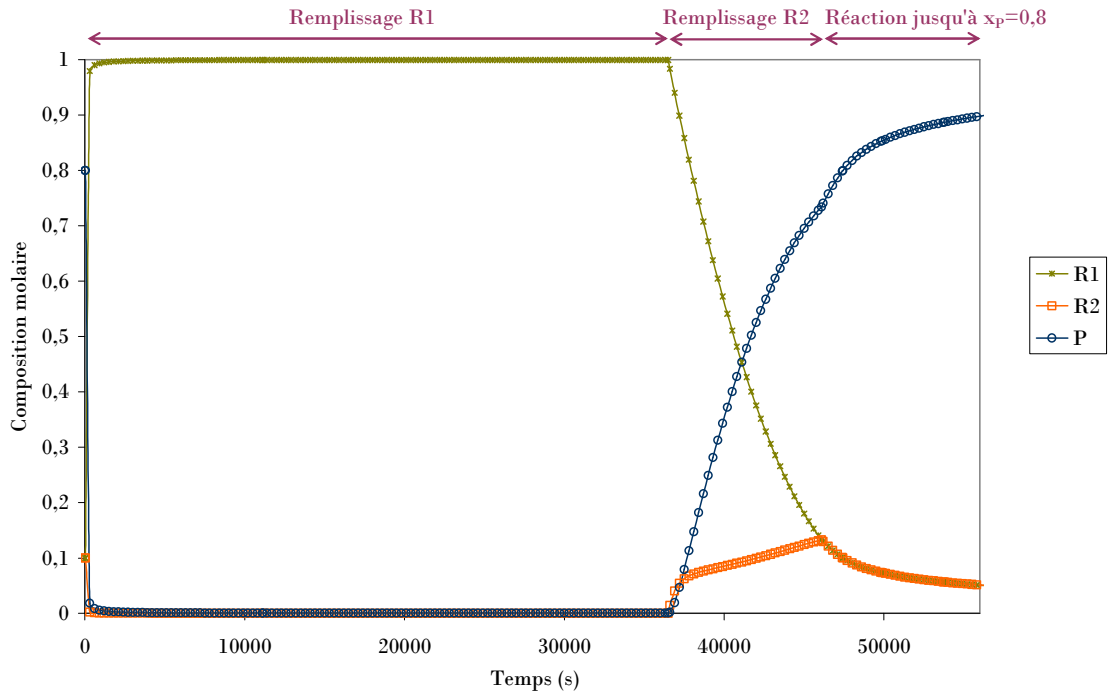


Figure 8.24 L'évolution de la composition dans BR1

8.3.4.2 La détection

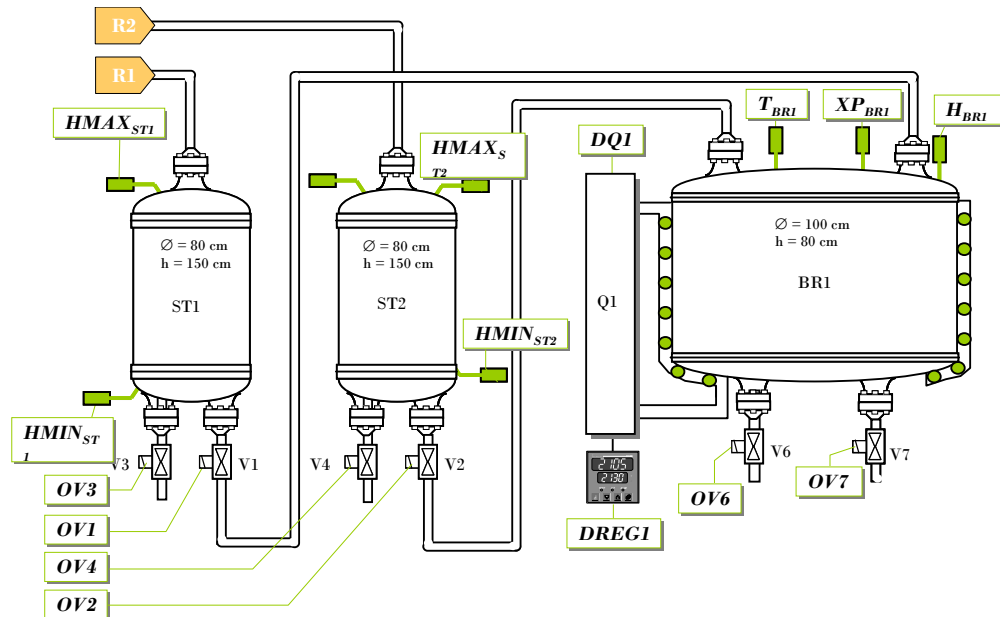


Figure 8.25 La réaction dans BR1

Le procédé simulé est un procédé complexe. Le problème de la surveillance de tel système est l'explosion combinatoire engendré par la considération de tous les défauts et modes de fonctionnement possibles du système. Nous sommes donc amenés à analyser des sous-systèmes, afin d'avoir une

représentation du système complet. Cette décomposition est d'autant plus appropriée pour un procédé batch. En effet, ce dernier peut être décomposé en opérations unitaires.

Notre étude ici porte en particulier sur la réaction effectuée dans le réacteur BR1. Ici, nous allons étudier la détection d'une dégradation : le débit traversant la vanne V2 est dégradée.

On a considéré 17 types de signatures chacune reliées à une grandeur physique :

- la signature s_1 au flux traversant la vanne V1,
- la signature s_2 au flux traversant la vanne V2,
- la signature s_3 à la composition en R1 dans la cuve ST1,
- la signature s_4 à la composition en R2 dans la cuve ST1,
- la signature s_5 à la composition en P dans la cuve ST1,
- la signature s_6 à la rétention liquide dans la cuve ST1,
- la signature s_7 à la composition en R1 dans la cuve ST2,
- la signature s_8 à la composition en R2 dans la cuve ST2,
- la signature s_9 à la composition en P dans la cuve ST2,
- la signature s_{10} à la rétention liquide dans la cuve ST2,
- la signature s_{11} à la hauteur de liquide dans la cuve BR1,
- la signature s_{12} à la composition en R1 dans la cuve BR1,
- la signature s_{13} à la composition en R2 dans la cuve BR1,
- la signature s_{14} à la composition en P dans la cuve BR1,
- la signature s_{15} à la température dans la cuve BR1,
- la signature s_{16} à la rétention liquide dans la cuve ST1,
- et la signature s_{17} à la chaleur fournie par l'alimentation d'énergie de la cuve BR1.

La figure 8.15 illustre l'étape de détection. Le résidu associé à la rétention liquide du réacteur BR1 est représentée. Une analyse statistique a permis de calculer les erreurs de prédiction du filtre de Kalman et de déterminer un seuil limite de 150 moles. Ce seuil correspond, selon la loi normale, à une probabilité de 98% : c'est-à-dire qu'il y a 98% de chance que l'intervalle défini par ce seuil englobe le comportement normal.

On constate que le résidu obtenu reste dans la zone de confiance. En d'autres termes ce seuil est mal adapté pour détecter une dégradation. Il faut donc abaisser ce seuil. Un compromis est effectué afin d'éviter l'émission de fausses alarmes. Pour cela on s'est basé sur la même hypothèse énoncée pour les indicateurs de défaut : on va abaisser le seuil à une probabilité de 0,68 (cf. figure 5.14) : on obtient alors le nouveau seuil à 50 moles.

À partir de $t \approx 2400$ s, le résidu associé à la rétention dans le réacteur BR1 sort de la zone de fonctionnement normal. Le diagnostic est lancé à $t \approx 3000$ s.

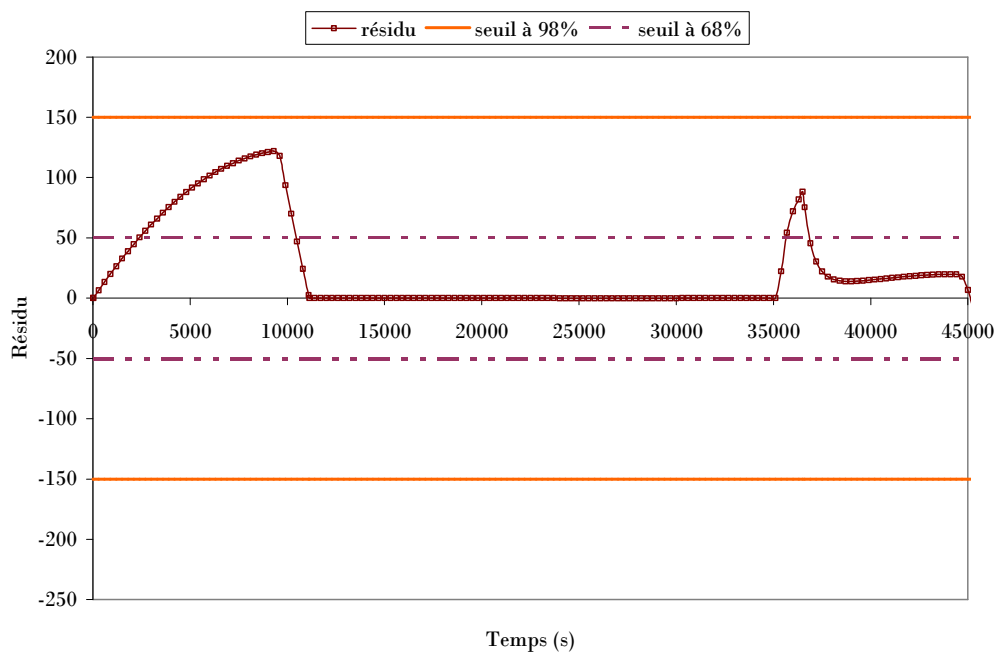


Figure 8.26 La détection d'une dégradation

On évalue alors le vecteur résidu et on obtient la signature de défaut instantanée correspondante :

| | Signature de défaut instantanée | | Signature de défaut instantanée |
|-------|---------------------------------|----------|---------------------------------|
| s_1 | 0,217753561 | s_{10} | 0 |
| s_2 | 0 | s_{11} | 0,219522665 |
| s_3 | 0 | s_{12} | 0,001668004 |
| s_4 | 0 | s_{13} | 0,165055734 |
| s_5 | 0 | s_{14} | 0,165053539 |
| s_6 | 0,011825474 | s_{15} | 2,29838E-05 |
| s_7 | 0 | s_{16} | 0,219098039 |
| s_8 | 0 | s_{17} | 0 |
| s_9 | 0 | | |

Tableau 8.7 La signature de défauts instantanée

8.3.4.3 Le diagnostic

Les défauts considérés dans cette étude sont les suivants :

- Le défaut 1 correspond à un défaut sur l'alimentation d'énergie du réacteur BR1 : cette dernière fournit une quantité d'énergie dégradée.

- De même, le défaut 2 représente un défaut du système de refroidissement du réacteur BR1 qui fournit une quantité d'énergie dégradée.
- Le défaut 3 est un défaut de composition sur la cuve ST1, qui contient normalement le constituant pur R1. Ici, on trouve des traces du constituant R2 dans cette cuve.
- Le défaut 4 caractérise le même défaut mais cette fois-ci ce sont des traces du constituant P qui sont présentes.
- On considère également le même type de défaut sur la cuve ST2 qui contient normalement le constituant pur R2. Ainsi, le défaut 5 représente le fait qu'il existe du constituant R1 dans la cuve ST1.
- Le défaut 6 est le même défaut mais avec le constituant P.
- Le défaut 7 représente un défaut de l'alimentation d'énergie du réacteur qui n'est pas à la bonne température.
- Ensuite on considère des défauts d'actionneurs avec le défaut 8 correspondant au blocage de la vanne V1 en position ouverte,
- Et le défaut 9 à un état dégradé de la vanne V1 : le flux traversant cette vanne est dégradé.
- Le défaut 10 est identique au défaut 8 mais pour la vanne V2.
- De même le défaut 11 désigne le même défaut que le défaut 8 mais s'applique sur la vanne V2.

La signature instantanée de défauts (tableau 8.7) est comparée à la matrice d'incidence en calculant les indicateurs de défauts relatifs aux distances de Manhattan relative (cf. définition 5-10) et améliorée (cf. définition 5-14). Nous obtenons alors les indicateurs suivants :

| | Indicateur de Manhattan relatif | Indicateur de Manhattan amélioré |
|-----------|---------------------------------|----------------------------------|
| Défaut 1 | 0,921189326 | 0,840521793 |
| Défaut 2 | 0,921189326 | 0,866690032 |
| Défaut 3 | 0,901967277 | 0,151772456 |
| Défaut 4 | 0,902774532 | 0,190063729 |
| Défaut 5 | 0,882381445 | 0,000377897 |
| Défaut 6 | 0,88236707 | 0,000187328 |
| Défaut 7 | 0,921189326 | 0,866690032 |
| Défaut 8 | 0,923352186 | 0,679442691 |
| Défaut 9 | 0,986708971 | 0,954939633 |
| Défaut 10 | 0,92383701 | 0,847166553 |
| Défaut 11 | 0,997854434 | 0,995476809 |

Tableau 8.8 Les indicateurs de défauts de l'exemple didactique

Les valeurs des indicateurs de défaut ne nous permettent pas d'écarter des défauts puisque tous les taux sont supérieurs à 0,68. Par contre l'indicateur de défaut amélioré permet d'écarter les défauts 3, 5, 6 et 8. Il nous reste donc 6 défauts possibles. Nous utilisons alors la seconde hypothèse que nous avons

formulée (cf. le point 5.5.3) : il ne peut y avoir plus de trois défauts simultanés. Ainsi on conserve uniquement les indicateurs de défauts ayant les plus fortes valeurs :

- Le défaut 11 avec pour les deux distances un taux de plus de 99%,
- Le défaut 9 avec un taux à 95%,
- Et les défauts 2 et 7 qui ont des valeurs d'indicateur identique à 98,7%.

En combinant les résultats des deux indicateurs, on constate que le défaut 11 est dans les deux cas à des taux de plus de 99%, et en particulier c'est le défaut qui fournit les indicateurs maximum. On peut donc conclure sur la cause la plus probable de la défaillance : le défaut 11, qui représente l'état dégradé de la vanne V2 : le flux la traversant a un débit plus faible.

La valeur du résidu nous renseigne alors sur la grandeur de la déviation soit environ 0,1. Une estimation paramétrique ici serait profitable afin de déterminer plus précisément le coefficient d'ouverture de la vanne. Toutefois en vue des résultats, le système est en mode dégradé. On peut envisager de le laisser dans cet état. Dans ce cas là, il est intéressant de prendre en compte cette dégradation dans le modèle de référence afin qu'il se rapproche de la réalité. Ainsi, la modélisation du procédé de référence évolue en faisant intervenir cette fois ci un type *FaultyTwoWayValve* pour la vanne V2. Enfin nous pouvons conclure, que le système de surveillance *FDIPro* est capable de détecter et diagnostiquer les dégradations.

8.4 LA CONCLUSION

Ce chapitre a permis, au travers de trois applications, d'éclairer le lecteur sur la démarche à suivre pour modéliser et simuler la surveillance d'un procédé au sein de la plate-forme *PrODHyS*. Elle se résume globalement en quatre opérations : la définition du procédé en termes de thermodynamique, appareils, configurations, connexions, la description de la recette à réaliser, le réglage du filtre de Kalman et l'établissement des seuils de détection et de la matrice d'incidence par expériences.

De plus, les applications ont permis de soulever trois points essentiels :

- Le premier repose sur la simplicité des modèles à établir par l'utilisateur pour représenter, finalement, des systèmes assez complexes. Les opérations à réaliser sont en effet peu nombreuses et suivent une procédure générale et systématique. Le caractère générique de la démarche à suivre est d'ailleurs un point positif pour l'élaboration d'interfaces graphiques. De plus, les capacités du modèle à gérer les phénomènes complexes tels que les phénomènes énergétiques, hydrauliques et les inversions de flux ont été soulignés.
- Le second point concerne la capacité du système à détecter et à localiser divers types de défauts (défauts d'actionneurs, de capteurs, défauts thermiques...), en particulier la détection et le diagnostic de défauts multiples.
- Enfin, le dernier point important concerne le caractère évolutif et extensible de la plate-forme. Le procédé de la section 8.3 en est un exemple d'application. Ainsi, le modèle de référence peut être modifié et/ou complexifié suite au changement ou dégradation observés sur le système.

CONCLUSION GÉNÉRALE

Un des principes de base de nombreux systèmes FDI est de comparer en temps réel l'évolution du système observé avec celle obtenue par la simulation synchrone d'un modèle de référence. Des structures sont alors générées et exploitées lors de la localisation du défaut. De nombreuses propriétés sont requises dans un système de surveillance : la rapidité de détection et de diagnostic (en termes de temps de réponse), la capacité de détecter de manière précoce les dégradations du système, son caractère évolutif et sa facilité de mise en œuvre ainsi que sa capacité à identifier plusieurs défauts.

Les objectifs étant établis, un premier travail a d'abord consisté à définir les spécificités du domaine d'application auquel on s'est intéressé (le Génie des procédés) et notamment son caractère hybride a été souligné.

Ensuite, nous nous sommes intéressés aux méthodes classiques de surveillance regroupant aussi bien les problèmes de détection et d'évaluation que de diagnostic. Dans ce cadre, un large état de l'art a été établi afin de justifier le choix de l'approche retenue. Ces méthodes se distinguent par le type de connaissances qu'elles exploitent. On peut les regrouper en deux grandes familles :

- les méthodes nécessitant une connaissance approfondie du système physique ; ce sont des méthodes à base de modèles phénoménologiques. Le modèle peut alors être *quantitatif*, *qualitatif* ou *semi qualitatif*, ou encore *flou* ;
- les méthodes ne nécessitant pas de connaissances accrues du système physique, mais utilisant une connaissance basée sur les expériences passées. On distingue alors les méthodes sans modèle (méthodes dites aussi à base de connaissance) et les méthodes à base de modèles comportementaux.

Ainsi, l'approche *SimAEM* développée dans ce mémoire s'inscrit dans la famille des méthodes à base de modèles phénoménologiques, plus particulièrement les méthodes dites de *FDI*. La méthodologie développée s'appuie sur la reconstruction d'état à partir d'un filtre de Kalman étendu, pour générer une signature structurée non binaire. Cette dernière est ensuite exploitée au cours d'une procédure d'appariement. Elle est comparée à une matrice d'incidence en termes de distances. L'utilisation d'une telle méthodologie s'avère intéressante pour la surveillance d'un large panel de procédés.

Les points forts de la méthodologie *SimAEM* mise en œuvre s'inscrivent à plusieurs niveaux, tant sur le plan théorique que sur le plan implémentation logicielle :

- S'appuyant sur un modèle de simulation dynamique hybride, elle permet une représentation générale et rigoureuse du procédé surveillé. En effet, le modèle de référence est construit sur la base des *RdPDO* qui permettent d'une part, une description efficace des contraintes de synchronisation, de parallélisme et de séquençement et d'autre part, une représentation précise et fiable de la dynamique continue grâce aux équations différentielles algébriques (à condition de disposer d'un solveur suffisamment puissant, notamment pour la recherche et la gestion des événements ainsi que pour le calcul des conditions initiales).
- De manière générale, l'utilisation des concepts objet engendre une description de haut niveau, favorise la modularité des modèles, permet la création d'entités génériques, extensibles et réutilisables (en exploitant notamment les mécanismes de spécialisation et de composition). Dans ce cadre, le chapitre 7

a développé un ensemble de modèles représentant les éléments fondamentaux servant de base à l'élaboration de modèles plus complexes. Cette approche favorise donc l'évolutivité du système par la création de nouvelles entités au fur et à mesure des besoins. Ceci permet alors d'enrichir la connaissance du système en générant une nouvelle matrice d'incidence.

- Le système de surveillance est robuste face aux bruits et aux incertitudes sur le procédé, par l'utilisation du filtre de Kalman étendu qui masque ces perturbations et évite ainsi la levée de fausses alarmes.
- La génération de signatures structurées relatives et normées confère au système la capacité de prendre en compte des défauts ayant les mêmes symptômes et de les différencier.
- Par ailleurs, les exemples du chapitre 8 ont permis de montrer que cette méthodologie permet de prendre en compte différents types de défaut, qu'ils concernent les actionneurs ou les capteurs.
- De même, la capacité de pouvoir traiter le cas de défauts multiples est aussi un atout car il permet d'élargir le champ des possibilités.
- Nous avons aussi montré que cette approche permet à la fois une détection précoce et une estimation de la criticité de la défaillance. Ce point est particulièrement important dans un contexte de reprise sur défaillance. En effet, cette connaissance permet d'avoir une action plus réactive sur le système et d'estimer au mieux les procédures correctives à mener pour remettre le système en fonctionnement normal ou au moins, en fonctionnement dégradé.
- Enfin, la méthodologie ayant été clairement définie, elle a ensuite été implémentée au sein de la plateforme de simulation dynamique, *PrODHyS*. Celle-ci propose un ensemble de composants logiciels permettant la mise en place de la méthodologie *SimAEM* et son exploitation au sein du système de surveillance *FDIPro*. Dans ce cadre, la dernière partie de ce manuscrit a permis d'illustrer, au travers de trois applications, l'ensemble des concepts évoqués tout au long de ces travaux. La première, concernant l'opération de changement de solvant, a montré la démarche générale que doit adopter un utilisateur pour modéliser son procédé. La deuxième, relative au système des vases communicants, a permis d'illustrer les phénomènes hydrauliques et d'inversion de flux, très importants dans le domaine de la simulation dynamique. Enfin, la dernière traitant de la modélisation d'un procédé batch, a mis l'accent sur le caractère extensible des modèles et la possibilité de décomposer un système complexe en sous-systèmes.

Ces travaux de recherche ouvrent de nombreuses perspectives pour l'avenir et nous donnons ici les quelques voies envisagées.

Sur un plan théorique :

- À court terme, un premier travail consistera à affiner la décision obtenue en cas d'indéterminisme (c'est à dire lorsqu'il n'a pas été possible de distinguer plusieurs défauts). Une première méthode est de coupler le système avec un système de classification. Cette approche a été explorée au travers de la thèse de [Moktari, 2007]. Une autre méthode serait de simuler les différents scénarii selon les défauts

restant à discriminer. L'initialisation de ces scénarios se baserait sur l'état du modèle de référence au moment de la détection du défaut. En revanche, se pose le problème du temps calcul induit, surtout dans un contexte temps réel souvent soumis à des temps de réponse fixés. La dynamique du système surveillé est alors un élément important pour la légitimité d'une telle méthode. Enfin, la simulation arrière du système est aussi une possibilité à approfondir. En effet, celle-ci a donné lieu à quelques études au début de ces travaux. Cependant, cette voie a été mise en suspend, suite à la mise en évidence d'un certain nombre de verrous. Notamment, la non-bijectivité des RdP peut générer des conflits au niveau de certaines transitions. Une solution envisagée pour limiter l'exploration arrière des marquages accessibles est d'attribuer des probabilités de franchissement compte tenu de la connaissance du système. Une autre difficulté provient aussi de la nature des modèles phénoménologiques mis en œuvre qui intègre la thermodynamique. En effet, selon le second principe, on ne peut garantir que le chemin arrière soit identique au chemin avant réel.

- Une hypothèse de ces travaux est que le système surveillé est observable. Or, dans un contexte industriel, ceci n'est pas toujours vérifié. De même, un travail sur la diagnosticabilité pourrait compléter ce travail. Il serait alors intéressant de se placer dans des conditions critiques afin d'évaluer le nombre de capteurs nécessaires et d'établir des règles de placement optimal de ces capteurs.
- Enfin, un élément qui nous apparaît crucial pour l'applicabilité de cette méthodologie dans un contexte industriel est l'établissement de stratégies de décomposition des systèmes complexes afin de pallier au problème d'explosion combinatoire des défauts pris en compte. Dans le cas des procédés batch, une des voies à exploiter est leur structure particulière qui conduit à un découplage « naturel » des opérations unitaires, l'interface étant constituée par les cuves de stockage intermédiaires. Le 3^{ème} exemple abordée dans le chapitre 8 va dans ce sens, mais il nous semble que cet aspect mérite d'être approfondi et surtout formalisé afin d'obtenir des procédures systématique de décomposition.

Sur un plan pratique :

- Un premier travail consistera en la validation de la méthodologie sur un cas réel (pilote).
- Afin d'étendre ces fonctionnalités au plan du pilotage, il serait intéressant d'approfondir les travaux initiés sur la criticité et de coupler ce système avec d'autres méthodes comme le raisonnement à base de cas afin d'aider l'opérateur dans le choix d'une configuration de reprise.
- Dans la même optique, l'intégration de cette méthodologie dans un système de supervision global intégrant notamment un module d'ordonnancement réactif paraît une extension intéressante. Dans ce cadre, une architecture système envisagée est montrée sur la figure suivante :

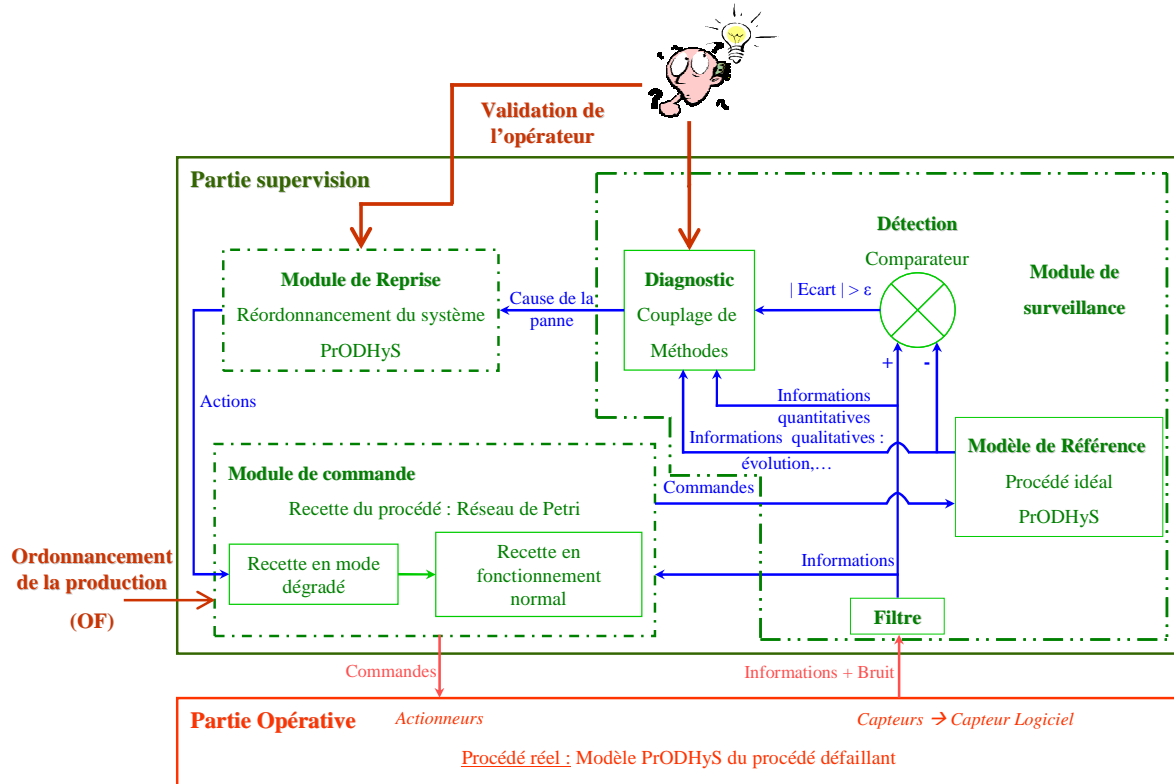


Figure L'architecture du système de supervision

- Un autre travail important sera de développer une interface graphique ouverte et conviviale afin de faciliter la conception et la mise en place sur site des procédures de surveillance basées sur la méthodologie *SimAEM*.
- Afin d'apporter une homogénéité des solutions aux problèmes d'ingénierie des procédés, il est primordial d'envisager une mise au standard CAPE-OPEN de l'ensemble des modules composant *PrODHyS*. Ceci permettra une interopérabilité et une intégration des composants logiciels de l'industrie des procédés. Dans ce cadre, une réflexion et des propositions pour la standardisation CAPE-OPEN concernant le traitement des systèmes discrets et hybrides seraient souhaitables.
- Enfin, un dernier champ d'intérêt important repose sur la manière avec laquelle le système a été utilisé jusqu'à présent. En effet, même si l'objectif ultime est l'implantation d'un tel système de surveillance sur un site de production industriel, la possibilité de simuler un procédé « réel » (c'est à dire, potentiellement défectueux) au moyen de *PrODHyS* (procédure mise en œuvre pour effectuer nos tests) pourrait conduire à l'utilisation de *FDIPro* pour des études de supervision, de surveillance et de diagnostic avec, comme objectif, des études de sécurité industrielle et le développement de simulateurs d'entraînement d'opérateurs.

Autant de sujets d'étude passionnants qui, nous l'espérons, viendront chacun apporter leur pierre à l'édifice, tout comme les travaux relatés dans ce manuscrit ont tenté de le faire.

ANNEXES

Annexe A : LES CONCEPTS ORIENTÉS OBJET

ET LE LANGAGE UML

Cette annexe est une brève introduction aux concepts de la technologie objet. Elle propose, pour chaque concept, la notation UML associée. Elle ne prétend en rien à l'exhaustivité et s'inspire très largement des ouvrages de D. Serain (2001) et de P.A. Muller et N. Gaertner (2001).

A.1 LE CONCEPT D'OBJET

Le concept d'*objet* nous est en fait très familier. Chaque jour, nous utilisons des objets ou nous leur faisons référence. Dans notre terminologie, un objet est identifié par son *nom* et est caractérisé par un ensemble d'*attributs* et d'*opérations*. Ainsi, un objet peut être une chose aussi réelle qu'une fleur ou un temps. Mais cela peut aussi représenter une chose immatérielle telle qu'un sentiment.

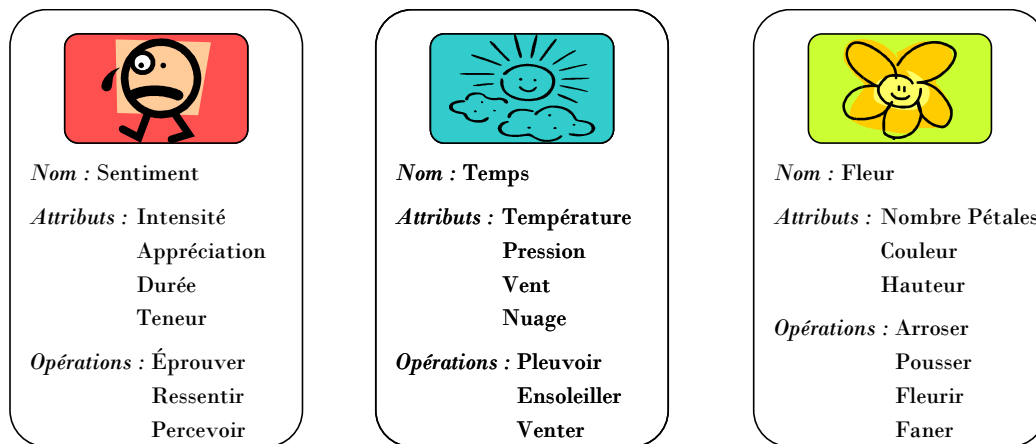


Figure A.1. Des exemples d'objets

Dans l'exemple de la Figure A.1, l'objet *Fleur* est référencé de façon unique par son espèce. Parmi ses *attributs*, on peut citer sa *couleur* et le *nombre de ses pétales* ; les actions (*opérations*) qui peuvent lui être associées sont par exemple : *Arroser*, *Pousser*, *Fleurir* et éventuellement *Faner*.

Selon Jacobson I., « *Un objet est caractérisé par un nombre d'opérations et un état qui mémorise l'effet de ces opérations* ». Dans cette définition, l'état d'un objet est représenté par l'ensemble des valeurs des attributs associés à cet objet.

Du point de vue informatique, un objet réunit en une seule entité les concepts de traitement (*opérations* ou *méthodes*) et les données (*attributs*). Ainsi, lorsque l'on représente un processus industriel sous la forme d'une collection d'objets, on analyse en même temps l'aspect « *traitement* » et l'aspect « *données* ». Ceci est en complète opposition avec l'approche traditionnelle de modélisation qui s'ingénie à séparer la partie traitement de la partie donnée.

A.2 LES CLASSES

A.2.1 La notion de classe

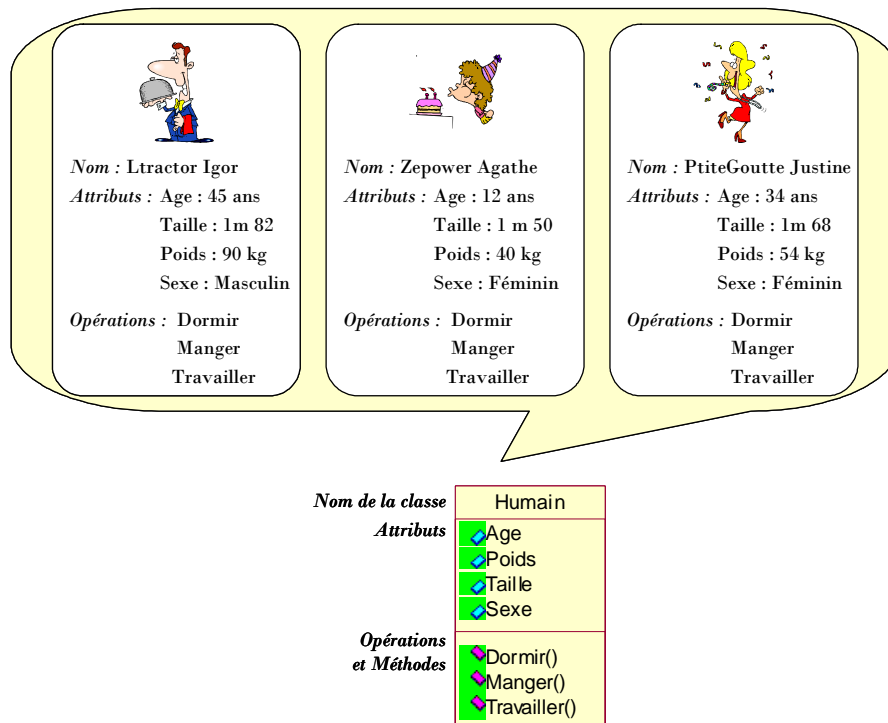


Figure A.2. Les classes et les instances de classe

Lorsque l'on construit un modèle à base d'objets, on constate que certains d'entre eux ont des caractéristiques communes en termes d'attributs et d'opérations. Il est alors intéressant de les regrouper sous une même entité. La Figure A.2 représente un exemple dans lequel les trois objets `Ltractor_Igor`, `Zepower_Agathe` et `PtiteGoutte_Justine` identifient trois humains distincts mais avec des caractéristiques communes en termes d'attributs et d'opérations. Au lieu de décrire chaque objet de façon individuelle, le concept de *classe* est alors utilisé.

Une classe contient tous les attributs et opérations communs aux objets qu'elle représente. Jacobson I. définit la notion de classe de la façon suivante : « Une classe représente un moule pour objets qui décrit leur structure interne. Les objets d'une même classe possèdent la même définition de leurs opérations et attributs ». Ainsi, dans l'exemple ci-dessus, la classe `Humain` regroupe les attributs et opérations communs aux objets `Ltractor_Igor`, `Zepower_Agathe` et `PtiteGoutte_Justine`.

En langage UML, la classe est représentée par un rectangle solide à trois compartiments (cf. Figure A.3) :

- Le premier contient le nom de la classe ;
- Celui du milieu contient la liste des attributs ;
- Le dernier contient les méthodes.

De plus, il existe trois niveaux de visibilité pour les attributs et les méthodes

- *Public* : visible par tous les clients de la classe (représenté par un rectangle sans clé ni cadenas et parfois par le signe « + ») ;
- *Protégé* : visible par les sous-classes de la classe (représenté par une clé et parfois par le signe « # ») ;
- *Privé* : visible par la classe seule (représenté par un cadenas et parfois par le signe « - »).
- *Statique* : accessible par toutes les instances de la classe (notion définie dans le point suivant) mais ces dernières n'en possèdent pas une copie (représenté par un « t » et parfois par le signe « \$ »).

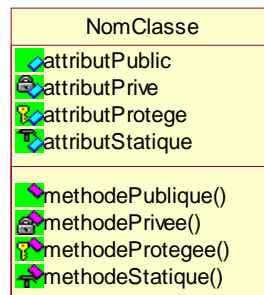


Figure A.3. La représentation d'une classe en langage UML

A.2.2 Une instance de classe

La classe apparaît donc comme une abstraction qui décrit toutes les caractéristiques communes aux objets membres de cette classe. Dans la technologie objet, tout objet appartient à une classe et est appelé *instance* de cette classe. Jacobson I. décrit une instance comme suit : « Une instance est un objet créé à partir d'une classe. La classe décrit la structure de l'instance (opérations et attributs), tandis que l'état courant de l'instance est défini par les opérations exécutées sur l'instance ». Ainsi, dans l'exemple de la Figure A.2, Ltractor_Igor, Zepower_Agathe et PtiteGoutte_Justine sont des instances de la classe Humain.

A.2.3 L'abstraction

Lors de la construction d'un objet, la question de sa complétude se pose. Il ne faut pourtant pas identifier et inclure tous les attributs et toutes les opérations possibles associés à un objet. En effet, construire un modèle d'objet signifie construire une représentation abstraite d'une entité matérielle et immatérielle. Cette abstraction doit incorporer dans le modèle, *les aspects de l'objet qui sont importants dans le contexte du domaine à traiter plutôt que d'essayer de représenter l'objet en entier.*

L'abstraction se concentre sur les caractéristiques importantes d'un objet selon le point de vue de l'observateur. Le domaine dans lequel se situe le problème à résoudre doit être alors utilisé comme un facteur discriminant. Ainsi, si l'on doit construire l'objet *Voiture*, les attributs et les opérations choisis seront très différents suivant que l'on construit un logiciel destiné à un utilisateur, à un concessionnaire ou à un garagiste. La Figure A.4 explicite cette approche :

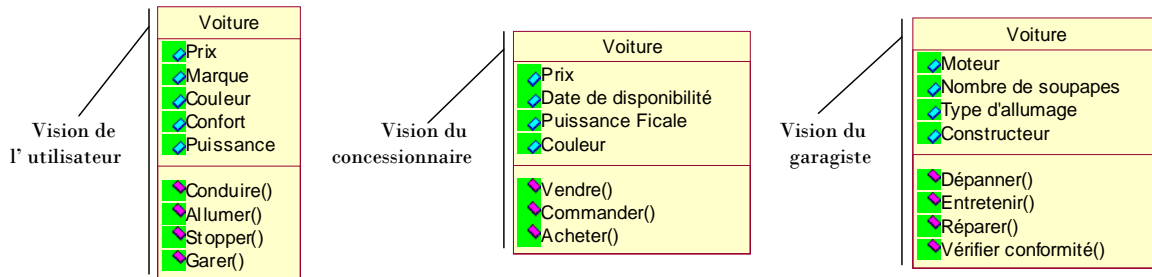


Figure A.4. Deux exemples d'abstraction de l'objet *Voiture*

A.2.4 L'encapsulation

L'encapsulation a pour but de séparer l'aspect externe d'un objet (c'est à dire ce qui est accessible par tous les objets) des détails d'implémentation. Elle permet ainsi d'occulter les détails de l'implémentation d'un objet. On parle alors de *masquage de l'information*.

A.2.4.1 L'encapsulation et les opérations

Les opérations associées à un objet représentent les actions possibles qui peuvent être demandées par d'autres objets. La structure externe de ces opérations doit être parfaitement définie.

La structure externe d'une opération est appelée sa *signature*. Elle est publique. Par contre, la partie de code associée est cachée. Ainsi, l'utilisateur d'un objet sait comment appeler une opération associée à cet objet, mais ne sait pas comment celle-ci est implémentée.

A.2.4.2 L'encapsulation et les attributs

Les attributs d'un objet permettent de sauvegarder son état. La valeur des attributs dépend de l'action des opérations effectuées sur l'objet. Cela signifie qu'ils ne sont pas directement accessibles de l'extérieur. La seule façon de les atteindre est d'utiliser une opération préalablement définie. Par convention, la plupart des langages possèdent deux opérations standards pour atteindre les attributs que l'on souhaite rendre publics. Elles peuvent s'écrire :

- `set_Nom_attribut` (paramètre)
- `get_Nom_attribut` (paramètre)

Par exemple, l'opération `set_Age` (46) de la classe *Humain* donne la valeur 46 à l'attribut `Age` de l'instance `Ltractor_Igor` (cf. Figure A.2). De ce fait, le nom et le type de chaque attribut sont publics mais la façon dont chaque attribut est codé est cachée.

Le concept d'encapsulation permet, en séparant l'aspect extérieur d'un objet (appelé *interface*) de son implémentation, d'atteindre deux objectifs qui sont :

- offrir au niveau de la modélisation le concept d'objet sans avoir à se soucier des problèmes d'implémentation ;
- offrir au niveau de l'implémentation, la flexibilité de choisir le langage de programmation ad hoc pour coder les méthodes et associer à une opération la ou les méthodes appropriées.

A.3 LES HIÉRARCHIES DE CLASSES

Les *hiérarchies de classes* ou *classifications* permettent de gérer la complexité en ordonnant les objets au sein d'arborences de classes d'abstraction croissante.

A.3.1 La généralisation et la spécialisation

La *généralisation* et la *spécialisation* permettent de construire des hiérarchies de classes. La généralisation décrit une relation entre une classe générale (classe de base ou classe parent) et une classe spécialisée (sous-classe). Elle consiste donc en la factorisation des éléments communs (attributs et opérations) d'un ensemble de classes dans une classe plus générale.

La *spécialisation* permet de capturer les particularités d'un ensemble d'objets non discriminés par les classes déjà identifiées. Une *classe spécialisée* est ainsi créée, intégralement cohérente avec la classe de base, mais comportant des caractéristiques nouvelles (attributs, opérations, associations).

L'exemple suivant montre une classification partielle du règne animal.

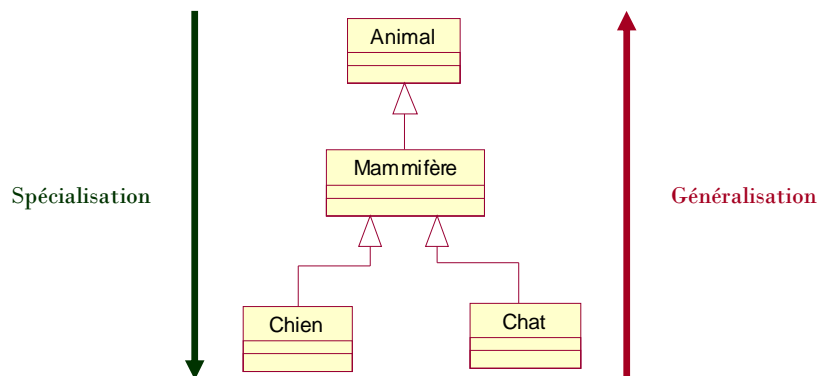


Figure A.5. Les notions de Généralisation / Spécialisation

A.3.2 L'héritage

Il existe de nombreuses manières de réaliser la classification. En programmation objet, la technique la plus utilisée repose sur l'héritage entre classes. L'héritage est une technique offerte par les langages de programmation objet pour construire une classe à partir d'une ou plusieurs autres classes. Elle consiste à partager des attributs et opérations au sein d'une hiérarchie de classes. Les classes *enfants*

héritent des caractéristiques de leurs classes *mères* ; les attributs et opérations déclarés dans la classe mère sont accessibles dans la classe enfant comme s'ils avaient été déclarés localement.

Par exemple, dans la Figure A.6.a, les classes Chat et Chien sont très similaires. Ces similitudes peuvent être extraites et regroupées dans une nouvelle classe appelée Mammifère. Une relation entre cette nouvelle classe et les classes existantes (Chien et Chat), appelée relation d'héritage, est alors définie. Par cette relation, les types Chien et Chat héritent des opérations et attributs de la classe Mammifère et puisque la classe Mammifère est une classe fille de la classe Animal, ils héritent également des opérations et attributs de la classe Animal. Ceci conduit à n'introduire, au niveau des classes Chien et Chat, que les éléments qui les différencient. Les éléments communs sont reportés dans la classe Mammifère ou Animal (cf. Figure A.6.b).

La notion d'héritage conduit à l'obtention d'un modèle très concis et sans redondance. Elle facilite également les mises à jour ; en effet, une modification réalisée au niveau de la super-classe est automatiquement reportée au niveau des sous-classes.

Le symbole utilisé pour la relation d'héritage ou de généralisation est une flèche avec un trait plein dont la pointe est un triangle fermé désignant le cas le plus général (cf. Figure A.6).

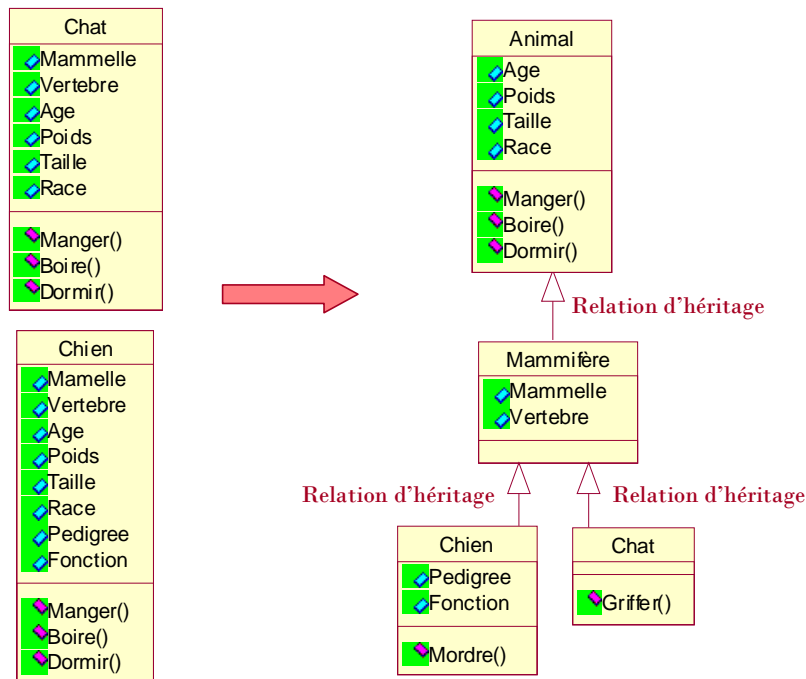


Figure A.6. L'héritage

A.3.3 La surcharge

La surcharge de méthodes est utilisée en permanence pour réaliser quatre types de surcharge :

- Ajout de comportement : l'objectif est de définir du code exécutable par rapport à une fonction "vide" du principe plus général. L'ajout de comportement est un cas particulier de la "redéfinition de méthode".
- Inhibition de comportement : l'objectif dans ce cas est au contraire, de *bypasser* le code exécutable que la généralisation fournissait. La méthode originale est surchargée par une méthode vide. C'est également un cas particulier de la "redéfinition de méthode".
- Redéfinition de méthode : le cas généralisé des précédents, dans lesquels le code de la méthode surchargée effectue un traitement complètement différent de la méthode originale.
- Complétion de méthode : généralement, des méthodes d'une classe mère doivent être "adaptées" pour tenir compte des caractéristiques ajoutées à l'objet. Le traitement original doit être effectué, puis un traitement additionnel.

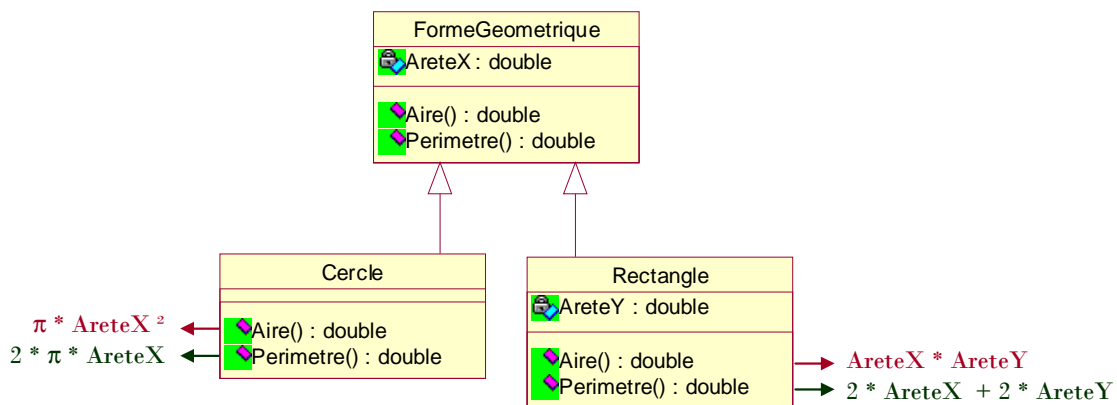


Figure A.7. La surcharge

Dans l'exemple de la Figure A.7, la classe *FormeGeometrique* possède les opérations *Calculer_Aire()* et *Perimetre()*. Ces opérations retournent réciproquement l'aire et le périmètre de la figure géométrique. Le mécanisme d'héritage transmet telle quelle ces opérations aux deux sous-classes *Cercle* et *Rectangle*.

Néanmoins, les calculs de l'aire et du périmètre de la forme géométrique dépendent de sa nature. Ainsi, l'aire d'un cercle se définit comme étant égal au produit du nombre PI et de son rayon au carré (représenté par l'attribut *AreteX*), quant à son périmètre il est défini comme étant égal à deux fois le produit du nombre pi et de son rayon (représenté par l'attribut *AreteX*). Les définitions des opérations héritées *Calculer_Aire()* et *Perimetre()* associées au type *Cercle* sont remplacées par ces nouvelles définitions. C'est le mécanisme de surcharge.

Les calculs de l'aire et du périmètre d'un rectangle sont aussi différents de ceux hérités du type *FormeGeometrique*. Dans l'exemple ci-dessus, ils sont égaux respectivement au produit de son grand côté

(représenté par l'attribut *AreteX*) et de son petit côté (représenté par l'attribut *AreteY*) pour l'un et pour l'autre à deux fois la somme de son grand côté (représenté par l'attribut *AreteX*) et de son petit côté (représenté par l'attribut *AreteY*). La substitution de ces nouvelles définitions à celles héritées repose ici aussi sur le mécanisme de surcharge.

A.3.4 Le polymorphisme

Le concept de surcharge défini précédemment montre qu'une opération héritée peut avoir un comportement différent suivant le type sur lequel elle s'applique dans la hiérarchie. Ce concept peut être généralisé à tous les types, *indépendamment des relations d'héritage*. Le polymorphisme consiste en « la capacité donnée à différentes classes de réagir à la même opération chacune à sa façon. »

En informatique par exemple, imprimer un fichier est une opération très courante. Cependant l'opération mise en œuvre est différente suivant que le type du fichier est « ANSI texte » ou « Postscript ». Ainsi, le même nom d'opération *imprimer()* correspond à deux comportements différents, ceux-ci étant directement liés aux classes sur lesquelles elle s'applique.

Le principe de polymorphisme est très utile en phase de modélisation. A ce niveau, les concepteurs sont intéressés principalement par la signature des opérations et non par la description de leur contenu. Ainsi, dans l'exemple ci-dessus lié aux fichiers informatiques, le concepteur devra identifier l'existence de l'opération *imprimer()* sur les fichiers du type « ANSI texte » ou « Postscript ». Il n'a pas à se soucier de savoir si ces opérations sont différentes ou non en termes de réalisation. Ce qui importe pour lui est l'utilisation du nom le plus approprié à l'opération qu'il souhaite inclure dans son modèle.

A.3.5 La généricité

En programmation objet, la généricité est mise en œuvre par le biais de *classes génériques*, appelées aussi *classes paramétrables* ou *classes « templates »*. Elles offrent un mécanisme de réutilisation basé sur la définition de *types génériques*. Ces classes sont définies sans préciser le type qu'elles utilisent. Ces types non spécifiés sont des paramètres précisés lors de la définition de classes particulières (processus appelé *instanciation*).

En langage UML, une classe générique est représentée par un rectangle (comme la classe) avec une deuxième zone en pointillés contenant le paramètre générique.

Par exemple, il est possible de paramétrer une liste générique avec le type d'éléments qu'elle contient (cf. Figure A.8). Le comportement commun à toutes les listes est synthétisé dans la classe paramétrable *Liste* et réutilisé lors de la création des classes *Liste_d_Entiers* et *Liste_d_Animaux*. Des objets de type *Liste_d_Entiers* ou *Liste_d_Animaux* peuvent ensuite être instanciés.

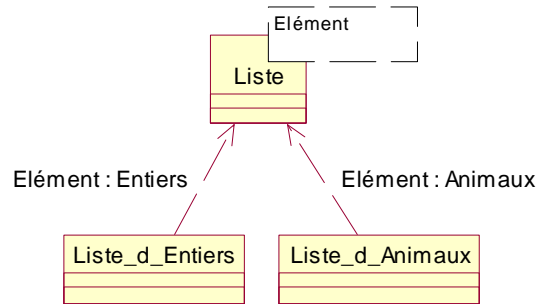


Figure A.8. La classe générique

A.4 LES RELATIONS ENTRE CLASSES

Les liens particuliers qui relient les objets peuvent être vus de manière abstraite dans le monde des classes : à chaque famille de liens entre objets, correspond une relation entre les classes de ces mêmes objets. De même que les objets sont des instances des classes, les *liens entre objets* sont des instances des *relations entre classes*.

A.4.1 L'association

Une association est une abstraction des liens qui existent entre les objets instances des classes associées. Par exemple, on peut créer, à partir de l'association définie entre la classe Voiture et la classe Roue (cf. Figure A.9), une instance de cette association entre *la Clio* et *ses roues*. Cette instance est également appelée *lien*. En langage UML, l'association se représente par une ligne entre les classes associées.

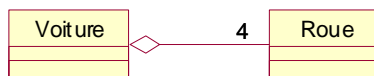


Figure A.9. La relation d'association

Plusieurs éléments caractérisent l'association :

- La plupart des associations sont *binaires*, c'est à dire qu'elles connectent deux classes.
- L'extrémité d'une association possède un nom. Ce nom, appelé aussi *rôle*, décrit comment une classe source voit une classe de destination au travers de l'association. Le rôle représente un attribut de la classe source. Par exemple, sur la Figure A.9, Roue est un attribut de la classe Voiture.
- Une indication de *visibilité* peut être précisée. Elle est placée avant le rôle pour préciser la visibilité du rôle à l'extérieur de l'association.

| | |
|------|---------------------|
| 1..1 | Un et un seul |
| 0..1 | Zéro ou un |
| 0..* | De zéro à plusieurs |
| 1..* | De un à plusieurs |

Tableau A. 1 Les valeurs des multiplicités

- Chaque extrémité d'une association peut porter une indication de *multiplicité* qui montre combien d'objets de la classe considérée peuvent être liés à un objet de l'autre classe. La multiplicité est une information portée par l'extrémité de l'association, sous la forme d'une expression entière. Le Tableau A. 1 représente les valeurs des multiplicités utilisées dans le cadre de manuscrit. Par exemple, la Figure A.9 rend compte du fait qu'une voiture possède quatre roues.
- Enfin, par défaut, les associations sont *navigables* dans les deux directions. Dans certains cas, une seule direction de navigation est utile ; l'extrémité de l'association vers laquelle la navigation est possible porte une flèche. L'absence de flèche signifie que la relation est navigable dans les deux sens. Dans l'exemple suivant, les objets instances de A voient les objets instances de B, mais les objets instances de B ne voient pas les objets instances de A.



Figure A.10. La navigabilité

A.4.2 L'agrégation

De façon générale, une relation exprime une forme de couplage entre abstractions. La force de ce couplage dépend de la nature de la relation dans le domaine du problème. Par défaut, l'*association* exprime une relation à couplage faible, les classes associées restant relativement indépendantes l'une de l'autre. L'*agrégation* est une forme particulière d'association qui exprime un couplage plus fort entre classes. Une des classes joue un rôle plus important que l'autre. L'agrégation permet d'exprimer des relations du type maître et esclaves. Elle représente une connexion bidirectionnelle dissymétrique.

L'agrégation ne peut concerner qu'un rôle de l'association. Elle se représente en ajoutant un petit losange du côté de l'agregat.

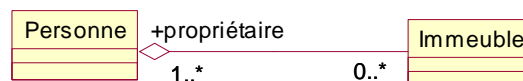


Figure A.11. La relation d'agrégation

Le diagramme de la Figure A.11 montre que des personnes peuvent être copropriétaires des mêmes immeubles sur lesquels elles possèdent des droits conjoints.

A.4.3 La composition

La composition est un cas particulier d'agrégation avec un couplage plus important. Ce couplage de composition indique que les composants ne sont pas partageables et que la destruction de l'agrégat entraîne la destruction des composants agrégés. Les termes « *conteneur* » et « *composite* » sont parfois utilisés pour désigner l'agrégat. La valeur maximale de multiplicité du côté de l'agrégat ne doit pas excéder *un* puisque les objets, instances de la classe des composants, doivent tous appartenir au même objet conteneur.

En langage UML, la composition est représentée par un petit losange plein du côté de l'agrégat. L'exemple suivant présente le cas des voitures. Chaque voiture possède un moteur qui ne peut être partagé entre plusieurs voitures. La destruction de la voiture entraîne nécessairement la destruction du moteur.



Figure A.12. La relation de composition

A.5 LE DIAGRAMME DE SÉQUENCE

Le diagramme de séquence est l'une des vues dynamiques les plus importantes de la formulation UML. Il permet de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Cette représentation se concentre sur l'expression des interactions. Elle peut servir à illustrer un cas d'utilisation. La représentation graphique d'un message est une flèche pleine directionnelle. Cette représentation est similaire à celle d'une association sur un diagramme de classes ; cependant, les messages servent à représenter la communication entre les objets, et non la relation structurelle présente entre les classes. L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule « de haut en bas » de cet axe. La disposition des objets sur l'axe horizontal n'a pas de conséquence pour la sémantique du diagramme. La Figure A.13 montre la structure générale d'un diagramme de séquence contenant les composants primaires suivants :

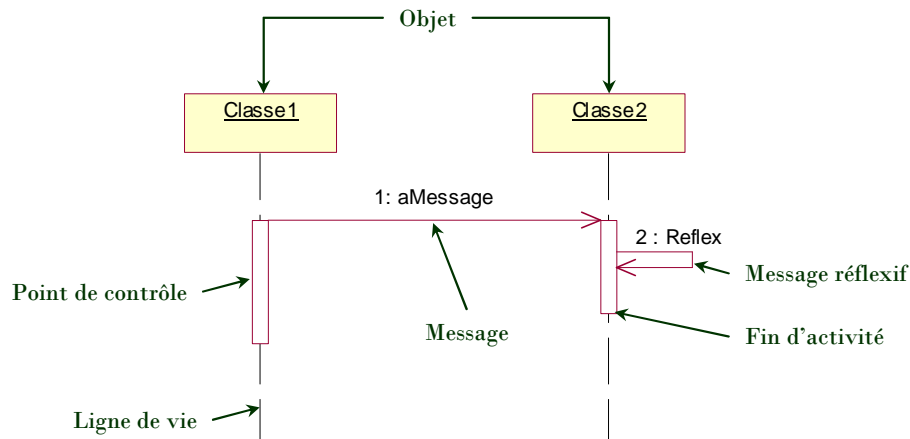


Figure A.13. La structure du diagramme de séquence

A.5.1 les objets

Sur un diagramme de séquence, les objets apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction (cf. Figure A.13).

A.5.1.1 La ligne de vie de l'objet

Ce composant représente la vie d'un objet dans le contexte de la séquence d'événements (cf. Figure A.13). Les objets qui sont créés vers la fin de la séquence n'apparaissent pas toujours en haut du diagramme, mais peuvent apparaître à l'endroit où ils sont créés. En outre, la ligne de vie peut se terminer à l'endroit où l'objet est détruit ou une référence à l'objet est perdue. De même, la création d'un objet peut être représentée simplement par l'envoi d'un nouveau message à un objet.

A.5.1.2 L'activation d'un objet

Sur un diagramme de séquence, il est aussi possible de représenter de manière explicite les différentes périodes d'activité d'un objet. Le *point de contrôle* illustre ainsi la période pendant laquelle un objet effectue une action. Dans ce contexte, la seule chose qui nous intéresse est de savoir pendant quelle période une méthode donnée détient le contrôle sur la séquence des événements. Le point de contrôle est représenté au moyen d'une bande rectangulaire superposée à la ligne de vie de l'objet (cf. Figure A.13).

A.5.2 Les messages

A.5.2.1 La représentation

Les *messages* sont représentés par des flèches directionnelles. Dans le diagramme de la Figure A.13, Classe1 envoie aMessage à Classe2. Au-dessus des flèches directionnelles figure un texte informatif du message envoyé entre les objets. L'ordre des messages est toujours déterminé par l'ordre dans lequel ils apparaissent dans le diagramme, c'est à dire du haut vers le bas. De plus, il n'y a aucune restriction quant au sens des messages : les messages peuvent être dirigés de droite à gauche ou de gauche à droite;

Le *message réflexif* représente un message qu'envoie un objet à une instance de lui-même (cf. Figure A.13).

A.5.2.2 Les stéréotypes

Comme vous pouvez le voir dans l'exemple ci-dessous (cf. Figure A.14), la formulation UML propose un certain nombre de stéréotypes graphiques pour décrire la nature du message :

- *message simple* : message dont on ne spécifie aucune caractéristique d'envoi ou de réception particulière,
- *message minuté* : message bloqué par l'expéditeur pendant un temps donné (qui peut être spécifié dans une contrainte), en attendant la prise en compte du message par le récepteur (l'expéditeur est libéré si la prise en compte n'a pas eu lieu pendant le délai spécifié),
- *message synchrone* : action ne se produisant qu'à partir de l'acceptation du message par le destinataire,

- *message asynchrone* : message n'interrompant pas l'exécution de l'expéditeur (le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré (jamais traité)),
- *message déroband* : message n'interrompant pas l'exécution de l'expéditeur et ne déclenche une opération chez le récepteur que s'il s'est préalablement mis en attente de ce message.

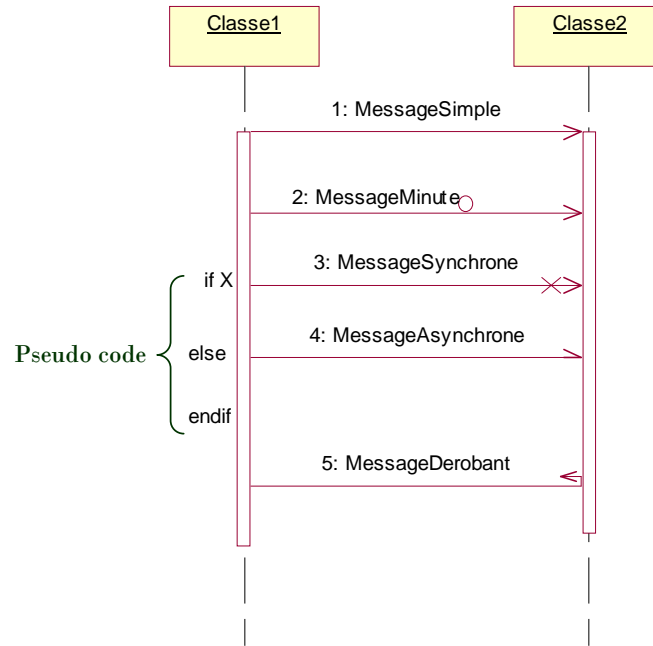


Figure A.14. Les différents messages

Pour représenter de manière graphique une exécution conditionnelle d'un message, on peut documenter un diagramme de séquence avec du pseudo-code (cf. Figure A.14).

Annexe B : LA DÉFINITION FORMELLE D'UN RdPDO

*Cette annexe expose la définition formelle d'un Réseau de Petri Différentiel
Objet (RdPDO). Elle s'appuie sur les travaux de Perret J. (2001).*

B.1 LA DÉFINITION FORMELLE

Formellement, un *réseau de Petri différentiel* à objets initialement marqué est un n-uplet :
 $N = \langle C, V, P, T, A, A_{AC}, Pre, Post, X, F, A_T, M_0 \rangle$

Où :

- $C = \{C_1, C_2, \dots, C_{n_c}\}$ est un ensemble fini de classes d'objets (types) ; $card(C) = n_c$. De plus, chaque élément C_k de l'ensemble C est un doublet $\langle Att_{C_k}, Meth_{C_k} \rangle$ où :
 - Att_{C_k} est un ensemble fini d'attributs de la classe C_k ;
 - $Meth_{C_k}$ est un ensemble fini de méthodes de la classe C_k .
- $V = \{V_1, V_2, \dots, V_{n_v}\}$ est un ensemble fini de variables formelles typées par des éléments de C ; $card(V) = n_v$.
- $P = \{P_1, P_2, \dots, P_{n_p}\}$ est un ensemble fini de places ; $card(P) = n_p$. L'ensemble P peut être décomposé en deux sous-ensembles disjoints P_H et P_D où P_H est un ensemble fini de places différentielles et P_D , un ensemble fini de places discrètes.
- $T = \{T_1, T_2, \dots, T_{n_t}\}$ est un ensemble fini de transitions ; $card(T) = n_t$.
- $A = \{A_1, A_2, \dots, A_{n_a}\}$ est un ensemble fini d'arcs ; $card(A) = n_a$.
- $A_{AC} : A \rightarrow V^{n_v}$ est une application qui associe à chaque arc, une somme formelle de n-uplets d'éléments de V .
- $Pre : P \times T \rightarrow V^{n_v}$ est l'application d'incidence avant qui définit les arcs entrants des transitions.
- $Post : P \times T \rightarrow V^{n_v}$ est l'application d'incidence arrière qui définit les arcs sortants des transitions.
- $X = \{X_1, X_2, \dots, X_{n_x}\}$ est un ensemble fini de variables ; $card(X) = n_x$. L'ensemble X peut être décomposé en sous-ensembles non nécessairement disjoints X_{P_k} de variables associées aux places

différentielles P_k . De plus, chaque ensemble X_{P_k} peut être décomposé en sous-ensembles non nécessairement disjoints $X_{P_k}^G, X_{P_k}^L, X_{P_k}^F$ tels que :

$$X_{P_k} = X_{P_k}^G \cup X_{P_k}^L \cup X_{P_k}^F$$

Où :

- $X_{P_k}^G$ est l'ensemble des variables globales au réseau de Petri tel que :

$$\left| \forall i = 1, \text{card}(P), \forall j = 1, \text{card}(P) : X_{P_i}^G = X_{P_j}^G = X^G \right.$$
- $X_{P_k}^L$ est l'ensemble des variables locales à la place P_k ;
- $X_{P_k}^F$ est l'ensemble des variables d'état définies comme attributs des variables formelles V_{P_k} , où $V_{P_k} = \text{Post}(P_k, \bullet)$.

- $F = \{F_1, F_2, \dots, F_{n_f}\}$ est un ensemble fini de fonctions ; $\text{card}(F) = n_f$. L'ensemble F peut être décomposé en sous-ensemble disjoints F_{P_k} de fonctions associées aux places différentielles P_k :

$$F(\dot{X}, X, p, t) = \begin{pmatrix} F_{P_1}(\dot{X}_{P_1}, X_{P_1}, p_{P_1}, t) \\ \vdots \\ F_{P_{n_f}}(\dot{X}_{P_{n_f}}, X_{P_{n_f}}, p_{P_{n_f}}, t) \end{pmatrix}$$

avec $F_{P_k}(\dot{X}_{P_k}, X_{P_k}, p_{P_k}, t) = \begin{bmatrix} f_{P_k 1}(\dot{X}_{P_k}, X_{P_k}, p_{P_k}, t) \\ \vdots \\ f_{P_k n_f}(\dot{X}_{P_k}, X_{P_k}, p_{P_k}, t) \end{bmatrix}$

- A_T est l'annotation de N : $A_T = \langle \text{Meth}^G, \text{Att}^G, \text{Meth}_{in}^{T_j}, \text{Att}_{in}^{T_j}, A_{TE}, A_{TA} \rangle$, où :
 - Meth^G est un ensemble fini de méthodes globales au réseau de Petri.
 - Att^G est un ensemble fini d'attributs globaux au réseau de Petri.
 - $\text{Meth}_{in}^{T_j}$ est l'union des méthodes des classes associées aux variables formelles de l'ensemble $V_{in}^{T_j}$ où $V_{in}^{T_j} = \text{Pre}(\bullet, T_j)$.
 - $\text{Att}_{in}^{T_j}$ est l'union des attributs des classes associées aux variables formelles de l'ensemble $V_{in}^{T_j}$.
 - $A_{TE} : T \rightarrow \text{Att}_{in}^{T_j} \times \text{Meth}_{in}^{T_j} \times \text{Att}^G \times \text{Meth}^G$ est une application qui associe à une transition, un événement sous la forme d'une conjonction de conditions en utilisant les éléments des ensembles $\text{Meth}_{in}^{T_j}, \text{Meth}^G, \text{Att}_{in}^{T_j}$ et Att^G ;
 - $A_{TA} : T \rightarrow \text{Att}_{in}^{T_j} \times \text{Meth}_{in}^{T_j} \times \text{Att}^G \times \text{Meth}^G$ est une application qui associe à une transition, une action sous la forme d'une suite de traitements réalisés par les méthodes des ensembles $\text{Meth}_{in}^{T_j}$ et Meth^G et appliqués sur les éléments des ensembles $\text{Att}_{in}^{T_j}$ et Att^G .
- M_0 est le marquage initial.

B.2 UN EXEMPLE ILLUSTRATIF

Afin d'illustrer cette définition formelle, les ensembles définis ci-dessus sont identifiés au travers d'un exemple. Celui-ci apparaît sur la Figure B.1. Notons simplement qu'il comporte trois classes, chacune possédant un réseau de Petri spécifique.

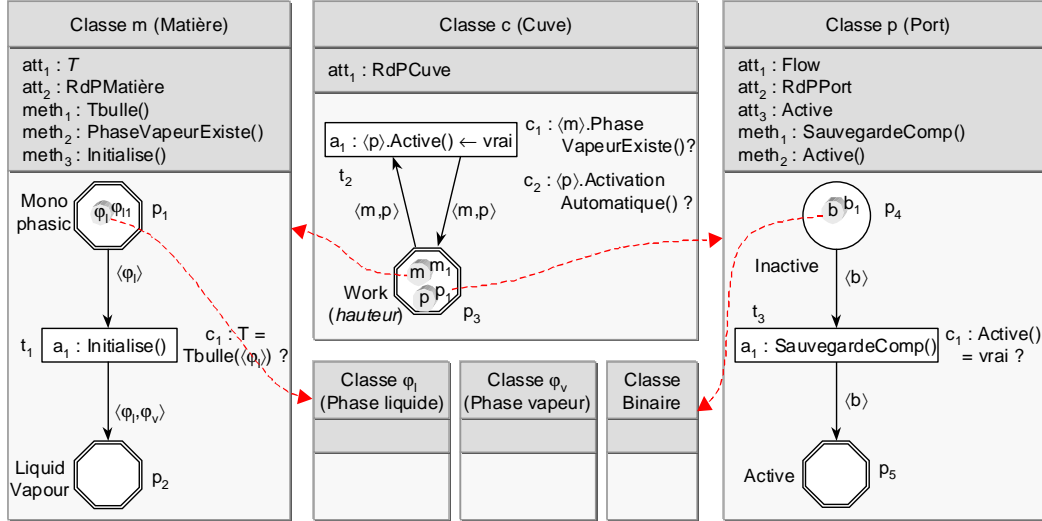


Figure B.1 Un exemple illustratif

- $C = \{C_1, C_2, \dots, C_{n_c}\}$ est un ensemble fini de classes d'objets (types) ; $card(C) = n_c$. De plus, chaque élément C_k de l'ensemble C est un doublet $\langle Att_{C_k}, Meth_{C_k} \rangle$ où :
 - Att_{C_k} est un ensemble fini d'attributs de la classe C_k ;
 - ↳ L'ensemble C associé au RdP de la cuve est le suivant : $C = \{m, p, c\}$. L'ensemble des attributs Att_{C_k} de la classe m est le suivant : $Att_{C_k} = \{T, RdPMatière\}$;
 - $Meth_{C_k}$ est un ensemble fini de méthodes de la classe C_k .
 - ↳ L'ensemble des méthodes $Meth_{C_k}$ de la classe m est le suivant :
 $Meth_{C_k} = \{Tbulle(), PhaseVapeurExiste(), Initialise()\}$.
- $V = \{V_1, V_2, \dots, V_{n_v}\}$ est un ensemble fini de variables formelles typées par des éléments de C ; $card(V) = n_v$.
 - ↳ L'ensemble V associé au RdP de la cuve est le suivant :
 $V = \{\langle m \rangle, \langle p \rangle\}$;
 $\langle m \rangle$ est de type m ;
 $\langle p \rangle$ est de type p .
- $P = \{P_1, P_2, \dots, P_{n_p}\}$ est un ensemble fini de places ; $card(P) = n_p$. L'ensemble P peut être décomposé en deux sous-ensembles disjoints P_H et P_D où P_H est un ensemble fini de places différentielles et P_D , un ensemble fini de places discrètes.
 - ↳ Les ensembles P, P_D et P_H associés au réseau de Petri du port sont les suivants :
 $P = \{p_4, p_5\}$;

- $$\left| \begin{array}{l} P_D = \{p_4\} ; \\ P_H = \{p_5\} . \end{array} \right.$$
- $T = \{T_1, T_2, \dots, T_{n_T}\}$ est un ensemble fini de transitions ; $\text{card}(T) = n_T$.
 - $\left| \text{L'ensemble } T \text{ associé au réseau de Petri du port est le suivant : } T = \{t_3\} . \right.$
 - $A = \{A_1, A_2, \dots, A_{n_A}\}$ est un ensemble fini d'arcs ; $\text{card}(A) = n_A$.
 - $\left| \text{L'ensemble } A \text{ associé au réseau de Petri du port est le suivant :} \right.$

$$A = \{a(p_4, t_3), a(t_3, p_5)\} .$$
- $A_{AC} : A \rightarrow V^{n_v}$ est une application qui associe à chaque arc, une somme formelle de n-uplets d'éléments de V .
 - $\left| \begin{array}{l} \text{Les n-uplets de variables formelles associés aux arcs des réseaux de Petri de} \\ \text{l'exemple sont les suivants :} \\ a(p_1, t_1) \rightarrow \langle \phi \rangle ; \\ a(t_1, p_2) \rightarrow \langle \phi_l, \phi_v \rangle ; \\ a(p_3, t_2) \rightarrow \langle m, p \rangle ; \\ a(t_2, p_3) \rightarrow \langle m, p \rangle ; \\ a(p_4, t_3) \rightarrow \langle b \rangle ; \\ a(t_3, p_5) \rightarrow \langle b \rangle . \end{array} \right.$
- $Pre : P \times T \rightarrow V^{n_v}$ est l'application d'incidence avant qui définit les arcs entrants des transitions.
 - $\left| \begin{array}{l} \text{La matrice Pre associée au réseau de Petri de la matière est la suivante :} \\ \\ Pre : p_1 \begin{pmatrix} t_1 \\ \langle \phi_l \rangle \end{pmatrix} \\ p_2 \begin{pmatrix} 0 \end{pmatrix} \end{array} \right.$
- $Post : P \times T \rightarrow V^{n_v}$ est l'application d'incidence arrière qui définit les arcs sortants des transitions.
 - $\left| \begin{array}{l} \text{La matrice Post associée au réseau de Petri de la matière est la suivante :} \\ \\ Post : p_1 \begin{pmatrix} t_1 \\ 0 \end{pmatrix} \\ p_2 \begin{pmatrix} \langle \phi_l, \phi_v \rangle \end{pmatrix} \end{array} \right.$
- $X = \{X_1, X_2, \dots, X_{n_X}\}$ est un ensemble fini de variables ; $\text{card}(X) = n_X$. L'ensemble X peut être décomposé en sous-ensembles non nécessairement disjoints X_{P_k} de variables associées aux places différentielles P_k . De plus, chaque ensemble X_{P_k} peut être décomposé en sous-ensembles non nécessairement disjoints $X_{P_k}^G, X_{P_k}^L, X_{P_k}^F$ tels que :

$$X_{P_k} = X_{P_k}^G \cup X_{P_k}^L \cup X_{P_k}^F$$
 - Où :
 - $X_{P_k}^G$ est l'ensemble des variables globales au réseau de Petri tel que :
 - $\left| \forall i=1, \text{card}(P), \forall j=1, \text{card}(P) : X_{P_i}^G = X_{P_j}^G = X^G \right.$
 - $X_{P_k}^L$ est l'ensemble des variables locales à la place P_k ;
 - $X_{P_k}^F$ est l'ensemble des variables d'état définies comme attributs des variables formelles V_{P_k} , où $V_{P_k} = Post(P_k, \bullet)$.

L'ensemble des variables X associé au RdP de la cuve est le suivant :

$$X = \{\text{Flow}, T, \text{hauteur}\};$$

Les ensembles des variables $X_{p_3}, X_{p_3}^G, X_{p_3}^L$ et $X_{p_3}^F$ associés à la place p_3 sont les suivants :

$$X_{p_3} = \{\text{Flow}, T, \text{hauteur}\}; X_{p_3}^G = \emptyset; X_{p_3}^L = \{\text{hauteur}\}; X_{p_3}^F = \{\text{Flow}, T\}.$$

- $F = \{F_1, F_2, \dots, F_{n_f}\}$ est un ensemble fini de fonctions ; $\text{card}(F) = n_f$. L'ensemble F peut être décomposé en sous-ensemble disjoints F_{p_k} de fonctions associées aux places différentielles P_k :

$$F(\dot{X}, X, p, \theta) = \begin{pmatrix} F_{p_1}(\dot{X}_{p_1}, X_{p_1}, p_{p_1}, \theta) \\ \vdots \\ F_{p_{n_p}}(\dot{X}_{p_{n_p}}, X_{p_{n_p}}, p_{p_{n_p}}, \theta) \end{pmatrix}$$

$$\text{avec } F_{p_k}(\dot{X}_{p_k}, X_{p_k}, p_{p_k}, \theta) = \begin{bmatrix} f_{p_{k,1}}(\dot{X}_{p_k}, X_{p_k}, p_{p_k}, \theta) \\ \vdots \\ f_{p_{k,n_f}}(\dot{X}_{p_k}, X_{p_k}, p_{p_k}, \theta) \end{bmatrix}$$

Les ensembles de fonctions définis au niveau des places p_1 et p_2 du réseau de Petri de la matière sont les suivants :

$$F_{p_1}(\dot{X}_{p_1}, X_{p_1}, p_{p_1}, \theta) = \emptyset$$

$$F_{p_2}(\dot{X}_{p_2}, X_{p_2}, p_{p_2}, \theta) = \begin{pmatrix} y_i - K_i \cdot x_i = 0 \\ \vdots \\ y_{n_c} - K_{n_c} \cdot x_{n_c} = 0 \\ \sum_{i=1}^{n_c} (y_i - x_i) = 0 \\ K_i - mK_i \cdot (T, P, x, y) = 0 \\ \vdots \\ K_{n_c} - mK_{n_c} \cdot (T, P, x, y) = 0 \end{pmatrix}$$

- A_T est l'annotation de N : $A_T = \langle \text{Meth}^G, \text{Att}^G, \text{Meth}_{in}^{T_j}, \text{Att}_{in}^{T_j}, A_{TE}, A_{TA} \rangle$ où :

- Meth^G est un ensemble fini de méthodes globales au réseau de Petri.

L'ensemble des méthodes globales Meth_G associé au réseau de Petri de la matière est le suivant :

$$\text{Meth}_G = \{\text{Tbulle}(); \text{PhaseVapeurExiste}(); \text{Initialise}()\}.$$

- Att^G est un ensemble fini d'attributs globaux au réseau de Petri.

L'ensemble des attributs globaux Att_G associé au réseau de Petri de la matière est le suivant :

$$\text{Att}_G = \{T; \text{RdPMatière}\}.$$

- $\text{Meth}_{in}^{T_j}$ est l'union des méthodes des classes associées aux variables formelles de l'ensemble $V_{in}^{T_j}$ où $V_{in}^{T_j} = \text{Pre}(\bullet, T_j)$.

L'ensemble $\text{Meth}_{in}^{t_2}$ associé au RdP de la cuve est le suivant :

$$\text{Meth}_{in}^{t_2} = \{\text{Tbulle}(); \text{PhaseVapeurExiste}(); \text{Initialise}(); \text{SauvegardeComp}(); \text{Active}()\}.$$

- $Att_{in}^{T_j}$ est l'union des attributs des classes associées aux variables formelles de l'ensemble $V_{in}^{T_j}$.

L'ensemble $Att_{in}^{t_2}$ associé au RdP de la cuve est le suivant :

$$Att_{in}^{t_2} = \{T; RdPMatière; Flow; RdPPort; Active\}.$$

- $A_{TE} : T \rightarrow Att_{in}^{T_j} \times Meth_{in}^{T_j} \times Att^G \times Meth^G$ est une application qui associe à une transition, un événement sous la forme d'une conjonction de conditions en utilisant les éléments des ensembles $Meth_{in}^{T_j}$, $Meth^G$, $Att_{in}^{T_j}$ et Att^G ;

La condition c_1 associée à la transition t_2 de la cuve utilise, par exemple, la méthode PhaseVapeurExiste() de l'ensemble $Meth_{in}^{t_2}$ associé au RdP de la cuve. De même, la condition c_1 associée à la transition t_1 du système de phases utilise, par exemple, la méthode Tbulle() de l'ensemble $Meth^G$ associé au RdP de la matière.

- $A_{TA} : T \rightarrow Att_{in}^{T_j} \times Meth_{in}^{T_j} \times Att^G \times Meth^G$ est une application qui associe à une transition, une action sous la forme d'une suite de traitements réalisés par les méthodes des ensembles $Meth_{in}^{T_j}$ et $Meth^G$ et appliqués sur les éléments des ensembles $Att_{in}^{T_j}$ et Att^G .

L'action a_1 associée à la transition t_2 de la cuve utilise, par exemple, la méthode Active() de l'ensemble $Meth_{in}^{t_2}$ associé au RdP de la cuve. De même, l'action a_1 associée à la transition t_1 du système de phases utilise, par exemple, la méthode Initialise() de l'ensemble $Meth^G$ associé au RdP de la matière.

- M_0 est le marquage initial.

Les marquages initiaux des RdP sont les suivants :

$$\text{Réseau de Petri de la matière : } M_0 = \begin{pmatrix} \Phi_{11} \\ 0 \end{pmatrix};$$

$$\text{Réseau de Petri du port : } M_0 = \begin{pmatrix} b_1 \\ 0 \end{pmatrix};$$

$$\text{Réseau de Petri de la cuve : } M_0 = (p_1 + m_1).$$

Annexe C : LE NOYAU NUMÉRIQUE : *DISCo*

Cette annexe présente brièvement le module DISCo, qui constitue le noyau numérique de la plate-forme PrODHyS.

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 2.2, *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.

C.1 LA 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 :

$$\begin{aligned} F(x^{(n)}, x, u, p, t) &= 0 \\ x_{t=0} &= x_0 \\ x_{t=0}^{(n)} &= x_0^{(n)} \end{aligned} \tag{C-3}$$

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 ;

u : ensemble des paramètres opératoires du système ;

p : ensemble des variables de commande du système ;

t : variable indépendante (généralement le temps).

C.2 LA REPRÉSENTATION OBJET DES MODÈLES MATHÉMATIQUES

C.2.1 La structure générale

Un ensemble de classes a été défini afin de représenter efficacement la plupart des systèmes d'équations rencontrés en simulation continue à savoir les équations linéaires, non linéaires et différentielles à un ordre quelconque (cf. Figure C.1).

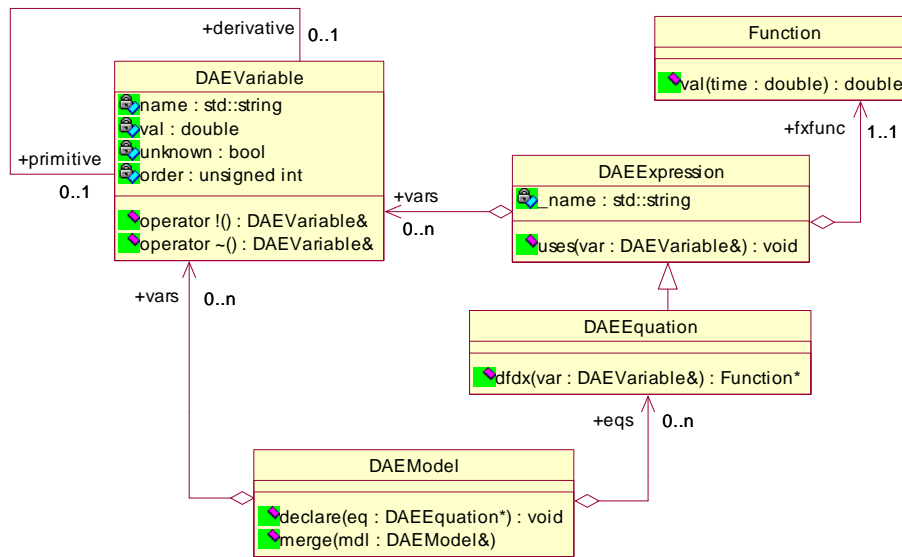


Figure C.1 La structure générale du module DISCo

La représentation mathématique continue d'un système se fait en termes de variables, de fonctions, d'équations et de modèles. Les classes définies dans cette section seront illustrées au travers de l'exemple suivant :

Soit une cuve M reliée à deux alimentations liquides A et B . Les produits sont purs et délivrés avec les débits molaires respectifs F_A et F_B .

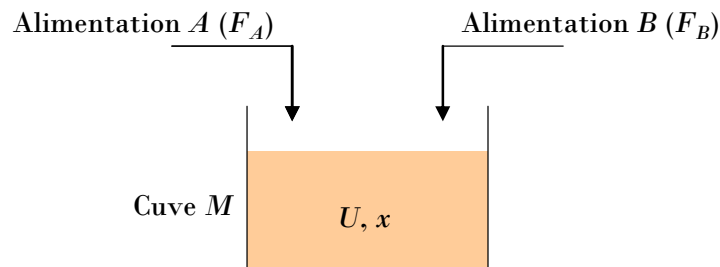


Figure C.2 L'alimentation d'un bac

La rétention U et la composition x (x_A , x_B) du mélange au sein de la cuve évoluent au cours du temps selon le modèle continu suivant :

$$F_A + F_B - \frac{dU}{dt} = 0$$

$$F_A - \frac{dUx_A}{dt} = 0$$

$$F_B - \frac{dUx_B}{dt} = 0$$

La première équation correspond au bilan matière global et les deux suivantes représentent les bilans matière partiels. Le système global est ainsi constitué de trois équations et de trois inconnues (la rétention molaire U et les fractions molaires des constituants A et B , x_A et x_B).

C.2.2 Les variables

La classe `DAEVariable` est la plus petite entité manipulable dans la bibliothèque `DISCo` et fixe le niveau de granularité du modèle. Elle est caractérisée par son nom, sa valeur, un booléen qui indique s'il s'agit d'un paramètre ou d'une inconnue et de son degré de dérivation. Des opérateurs spécifiques `!` et `~` ont été surchargés pour accéder respectivement à la dérivée et à la primitive de la variable. De plus, une classe nommée `DAEVariableVector` a été définie comme un vecteur d'éléments de type `DAEVariable` ; elle permet, par exemple, de représenter efficacement des variables de type fractions molaires. Afin de clarifier les idées, l'exemple suivant représente l'instanciation d'une variable et montre de quelle façon elle est définie.

Exemple d'instanciation de la variable de rétention U :

```
DAEVariable U( "rétention", vrai, 1 );
                ↳ nom           ↳ inconnue   ↳ degré
```

La variable de rétention est de type `DAEVariable`. Elle est caractérisée par son nom "rétention" et définie comme une inconnue. Son degré de dérivation est de 1 : une deuxième variable correspondant à sa dérivée (`!U`) est alors automatiquement créée.

C.2.3 Les équations

Selon le type de résolution utilisé, les informations requises à propos de l'équation ne sont pas les mêmes. Afin de tenir compte de ces différences, deux classes ont été conceptualisées pour définir une équation : il s'agit des classes `DAEExpression` et `DAEEquation`.

La classe `DAEExpression` contient la fonction à étudier `fxfunc`. Cet attribut, de type `Fonction`, n'est pas défini à ce niveau mais le devient dans les classes plus spécialisées. Il permet d'évaluer, grâce à la méthode `val()`, l'expression mathématique du résidu de la fonction. La classe `DAEExpression` contient également la liste de toutes les variables de l'expression mathématique.

La classe `DAEEquation` hérite de la classe `DAEExpression` et permet d'évaluer l'expression des dérivées par rapport aux variables. Ceci est réalisé grâce à la méthode virtuelle `dfdx()` surchargée dans les classes filles. De plus, une classe nommée `DAEEquationVector` a été définie comme un vecteur

d'éléments de type DAEEquation ; elle permet, par exemple, de représenter efficacement les bilans matière partiels.

Reprenons l'exemple précédent (cf. Figure C.2) et modélisons le bilan matière global avec la structure décrite sur la Figure C.1. Cinq nouvelles classes doivent alors être créées. Le diagramme de classe de la figure suivante en propose la synthèse :

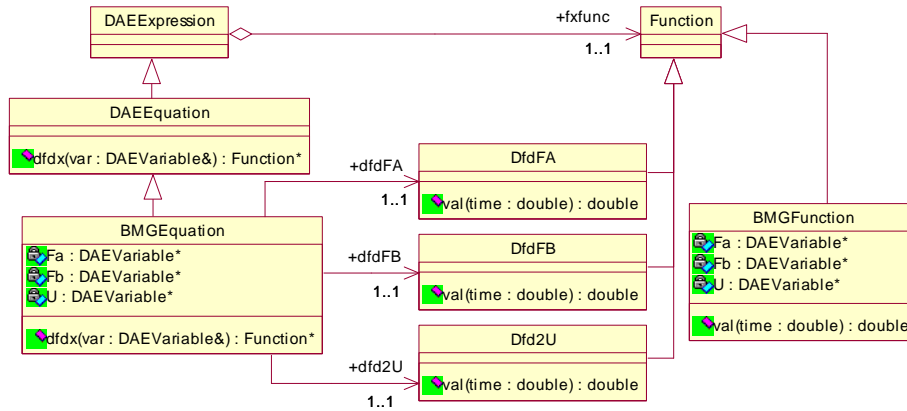


Figure C.3 Le diagramme de classes associé au bilan matière global

Il faut, dans un premier temps, définir la fonction à évaluer fxfunc. Dans ce contexte, la classe BMGFunction est créée comme une classe fille de la classe Function. Elle contient des références sur l'ensemble des variables de la fonction et une méthode val() qui retourne la valeur numérique de l'expression :

$$F_A + F_B - !U$$

Ensuite, trois autres classes doivent être définies : DfdFA, DfdFB, Dfd2U ; elles sont définies de la même manière que la classe BMGFunction et correspondent respectivement :

- à la dérivée de la fonction par rapport à la variable F_A (la méthode val() retourne la valeur 1) ;
- à la dérivée de la fonction par rapport à la variable F_B (la méthode val() retourne la valeur 1) ;
- à la dérivée de la fonction par rapport à la dérivée de la variable U (la méthode val() retourne la valeur -1) ;

Enfin, une dernière classe, nommée BMGEquation, est créée et correspond à l'équation de bilan proprement dite. Elle contient des références sur l'ensemble des variables de l'équation et sur l'ensemble des fonctions décrites précédemment. Elle surcharge la méthode dfdx(), initialement déclarée dans la classe mère, qui retourne la dérivée de la fonction par rapport à la variable transmise en argument ; elle fait alors appel aux fonctions « dérivées » définies précédemment.

C.2.4 Le modèle

La classe DAEModel n'est rien d'autre qu'un conteneur d'équations et de variables. Lorsqu'un nouvel objet DAEEquation est déclaré dans un modèle (declare()), ce dernier vérifie si les variables utilisées dans l'équation ont déjà été stockées ; si ce n'est pas le cas, il les rajoute à sa liste. Ce processus permet d'assurer l'unicité des variables au sein du modèle. D'autre part, deux modèles M_1 et M_2 peuvent également fusionner pour n'en former plus qu'un (merge()). Cela consiste à agréger l'ensemble des équations du modèle M_1 au modèle M_2 . Ce processus garantit également l'unicité des variables. La fusion de modèles offre une grande souplesse de modélisation. En effet, la description du comportement d'un appareil résulte d'un ensemble de petits modèles très ciblés. Cela favorise bien évidemment la modularité du modèle global.

À titre d'exemple, reprenons le système algébro-différentiel précédent et décrivons comment se déroule le remplissage des vecteurs d'équations et de variables associés au modèle.

| Equations classées selon leur ordre de déclaration | Variables associées à l'équation | Variables prises en compte |
|--|---|-------------------------------|
| $F_A + F_B - \frac{dU}{dt} = 0$ | F_A F_B U $!U$ | F_A F_B U $!U$ |
| $F_A - \frac{dUx_A}{dt} = 0$ | F_A U $!U$ x_A $!x_A$ | x_A $!x_A$ |
| $F_B - \frac{dUx_B}{d\tau} = 0$ | F_B U $!U$ x_B $!x_B$ | x_B $!x_B$ |

La déclaration du bilan matière global implique l'ajout, dans le vecteur de variables d'état, des variables F_A , F_B , U et $!U$; de la même manière, la déclaration des bilans matière partiels impliquent l'ajout des variables x_A , $!x_A$, x_B et $!x_B$. Toutefois, les variables appartenant à ces deux dernières équations ne sont pas toutes introduites puisque certaines d'entre elles ont déjà été ajoutées au niveau du bilan matière global.

C.3 LA RÉOLUTION

C.3.1 Les solveurs

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 l'environnement *DISCO* :

- Un solveur linéaire pour résoudre les systèmes de type $A \cdot X = B$;
- Un solveur de systèmes d'équations algébriques non linéaires de la forme $F(X) = 0$;
- Un solveur de systèmes algébro-différentiels de la forme $F\left(\frac{dx}{dt}, x, t\right) = 0$

Ces trois types de solveurs permettent de résoudre un grand nombre de problèmes de simulation rencontrés en génie chimique.

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

C.3.2 Le solveur d'équations différentielles algébriques

Ce solveur a été conçu dans le but de résoudre des systèmes algébro-différentiels, creux et de grande taille, en vue d'une simulation dynamique robuste. Il 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 principe de résolution se base sur un algorithme de type prédicteur / correcteur basé sur la différentiation rétrograde :

Prédicteur

D'après Gear, on pose :

$$x_{n+1} = \psi(x_n, x_{n-1}, \dots) + h \cdot \beta \cdot \frac{dx_{n+1}}{dt} \quad (\text{C-4})$$

avec $x_{n+1} = x(t+h)$

β : le paramètre leader qui correspond à la constante dépendant de l'ordre de la méthode ;

h : le pas d'intégration ;

ψ : le polynôme fonction des variables aux pas précédents.

À l'ordre 1, on obtient la relation d'Euler avec $\beta = 1$ et $\psi = x_k$.

Correcteur

On peut alors approximer la dérivée :

$$\frac{dx_{n+1}}{dt} = \frac{x_{n+1} - \Psi(x_n, x_{n-1}, \dots)}{h \cdot \beta} \quad (\text{C- 5})$$

Le système (C- 3) s'écrit alors :

$$F\left(\frac{x_{n+1} - \Psi(x_n, x_{n-1}, \dots)}{h \cdot \beta}, x_{n+1}, u, p, t\right) = 0 \quad (\text{C- 6})$$

Le système est alors un système d'équations différentielles algébriques non linéaires pouvant être résolu par une méthode type Newton-Raphson, c'est-à-dire dont le principe est basée sur l'équation suivante :

$$x_{n+1} = x_n + \Delta x_n \quad (\text{C- 7})$$

Afin d'établir cette relation de redondance, on écrit alors le développement limité de la fonction F au voisinage de x_n :

$$\begin{aligned} 0 &= F\left(\frac{dx_{n+1}}{dt}, x_{n+1}, u, p, t\right) \\ &= F\left(\frac{dx_n}{dt}, x_n, u, p, t\right) + \dot{F}\left(\frac{dx_n}{dt}, x_n, u, p, t\right) \cdot (x_{n+1} - x_n) \end{aligned} \quad (\text{C- 8})$$

On obtient alors :

$$x_{n+1} = -\left[\dot{F}\left(\frac{dx_n}{dt}, x_n, u, p, t\right)\right]^{-1} \cdot F\left(\frac{dx_n}{dt}, x_n, u, p, t\right) + x_n \quad (\text{C- 9})$$

De plus, la dérivée du système F s'écrit :

$$\dot{F}\left(\frac{dx_n}{dt}, x_n, u, p, t\right) = \frac{\partial F\left(\frac{dx_n}{dt}, x_n, u, p, t\right)}{\partial x} + \frac{\partial \dot{x}}{\partial x} \cdot \frac{\partial F\left(\frac{dx_n}{dt}, x_n, u, p, t\right)}{\partial \dot{x}} \quad (\text{C- 10})$$

$$\text{avec } \dot{x} = \left(\frac{dx_n}{dt}\right)$$

À l'ordre 1, l'expression (C- 10) combinée l'expression (C- 4) devient :

$$M = \dot{F}\left(\frac{dx_n}{dt}, x_n, u, p, t\right) = \frac{\partial F\left(\frac{dx_n}{dt}, x_n, u, p, t\right)}{\partial x} + \frac{1}{h \cdot \beta} \cdot \frac{\partial F\left(\frac{dx_n}{dt}, x_n, u, p, t\right)}{\partial \dot{x}} \quad (\text{C- 11})$$

M est appelé opérateur dynamique.

Δx_n de l'expression (C- 7) correspond donc à la résolution du système suivant :

$$\Delta x_n = -M^{-1} \cdot F\left(\frac{dx_n}{dt}, x_n, u, p, t\right) \quad (\text{C- 12})$$

Ceci implique que la résolution du système d'équations différentielles algébriques par la méthode de Gear nécessite la connaissance des gradients $\frac{\partial F}{\partial x}$ et $\frac{\partial F}{\partial \dot{x}}$.

C.4 LE LIEN ENTRE MODÈLES ET SOLVEURS

La conception de cette bibliothèque visait à découpler totalement modélisation et résolution. En effet, les modèles sont composés d'ensembles inorganisés de variables et d'équations. Or, les solveurs ont besoin de structuration. Pour cette raison, les objets de type `DAEMap` se chargent de l'adéquation et de l'adaptation des modèles au format des solveurs. Un objet `DAEMap` ne peut se construire qu'à partir d'un objet `DAEModel`. Il alimente son vecteur d'état à partir des *inconnues* du modèle (de type `DAEVariable`) ; lors du remplissage, les `DAEVariable` ayant le statut de *paramètre* sont écartées. De la même manière, il alimente son vecteur d'équations avec celles qui lui ont été spécifiées ; les équations qui ne comportent que des paramètres sont écartées du système.

Les solveurs ne sont ainsi liés au modèle qu'au travers d'un objet `DAEMap`, ce qui assure l'intégrité des modèles et de leur résolution.

Annexe D : LES DISTANCES DE MANHATTAN RELATIVE ET AMÉLIORÉE

Cette annexe contient les démonstrations de la définition 5-10 et de la définition 5-14.

Afin de démontrer que l'application \mathcal{D} est une distance il faut vérifier que \mathcal{D} possède les trois propriétés citées dans la propriété 5-7. Rappelons ces propriétés :

Soient $X \in \mathcal{S}, Y \in \mathcal{T}$,

- $\mathcal{D}(X, Y) = 0 \Rightarrow X = Y$ (Séparation)
- $\mathcal{D}(X, Y) = \mathcal{D}(Y, X)$ (Symétrie)
- Soit $Z \in \mathcal{S}, \mathcal{D}(X, Z) \leq \mathcal{D}(X, Y) + \mathcal{D}(Y, Z)$ (Inégalité triangulaire)

D.1 LA DISTANCE DE MANHATTAN RELATIVE

Rappelons dans un premier temps la définition de cette distance et démontrons les trois propriétés qu'elle doit vérifier.

Soient \mathcal{S} l'espace des signatures instantanées relatives normées et \mathcal{T} l'espace borné des signatures de défaut théoriques ($\text{card}(\mathcal{T})=m$, m étant le nombre de défauts considérés).

La distance de Manhattan relative entre une signature instantanée $\mathcal{S}^{rN}(t)$ et une signature de défaut particulière $\mathcal{T}_{\cdot,j}$ (toutes deux de dimension n) est définie par la relation suivante :

$$\mathcal{D}_j^{Mr}(t) = \frac{\sum_{i=1}^n |\mathcal{S}_i^{rN}(t) - \mathcal{T}_{ij}|}{n}$$

D.1.1 La propriété de séparation

Soient $\mathcal{S}^{rN}(t) \in \mathcal{S}$ et $\mathcal{T}_{\cdot,j} \in \mathcal{T}$.

On suppose que la distance de Manhattan relative entre $\mathcal{S}^{rN}(t)$ et $\mathcal{T}_{\cdot,j}$ est nulle :

$$\mathcal{D}_j^{Mr}(t) = 0$$

$$\Leftrightarrow \frac{\sum_{i=1}^n |s_i^{rN}(t) - \tau_{ij}|}{n} = 0$$

$$\Leftrightarrow \sum_{i=1}^n |s_i^{rN}(t) - \tau_{ij}| = 0$$

$$\text{Or, } |s_i^{rN}(t) - \tau_{ij}| \geq 0, \forall i \in [1; n].$$

$$\text{On en déduit que : } |s_i^{rN}(t) - \tau_{ij}| = 0, \forall i \in [1; n].$$

$$\Leftrightarrow \text{On en déduit que : } s_i^{rN}(t) = \tau_{ij}, \forall i \in [1; n].$$

$$\Leftrightarrow s^{rN}(t) = \tau_{\cdot, j}$$

La propriété de séparation est donc démontrée.

D.1.2 La propriété de symétrie

Soient $s^{rN}(t) \in S$ et $\tau_{\cdot, j} \in T$.

$$\text{On a : } \mathcal{D}_j^{Mr}(t) = \mathcal{D}_j^{Mr}(s^{rN}(t), \tau_{\cdot, j})$$

On en déduit :

$$\begin{aligned} \mathcal{D}_j^{Mr}(s^{rN}(t), \tau_{\cdot, j}) &= \frac{\sum_{i=1}^n |s_i^{rN}(t) - \tau_{ij}|}{n} \\ &= \frac{\sum_{i=1}^n |s_i^{rN}(t) - \tau_{ij}|}{n} \\ &= \frac{\sum_{i=1}^n |\tau_{ij} - s_i^{rN}(t)|}{n} \\ &= \mathcal{D}_j^{Mr}(\tau_{\cdot, j}, s^{rN}(t)) \end{aligned}$$

CQFD.

D.1.3 L'inégalité triangulaire

Soient $x, y, z \in S$. Notons que $T \subset S$.

Nous avons :

$$\begin{aligned} \mathcal{D}^{Mr}(x, y) &= \frac{\sum_{i=1}^n |x_i - y_i|}{n} \\ &= \frac{\sum_{i=1}^n (|x_i - z_i| + |z_i - y_i|)}{n} \end{aligned}$$

Or la valeur absolue vérifie l'inégalité triangulaire : $|a + b| \leq |a| + |b|$.

On en déduit que : $\left| (x_i - z_i) + (z_i - y_i) \right| \leq |x_i - z_i| + |z_i - y_i|, \forall i \in [1; n]$.

On obtient finalement :

$$\begin{aligned} \mathcal{D}^{\text{Mr}}(\mathbf{x}, \mathbf{y}) &\leq \frac{\sum_{i=1}^n \left[|x_i - z_i| + |z_i - y_i| \right]}{n} \\ &\leq \frac{\sum_{i=1}^n |x_i - z_i| + \sum_{i=1}^n |z_i - y_i|}{n} \\ &\leq \mathcal{D}^{\text{Mr}}(\mathbf{x}, \mathbf{z}) + \mathcal{D}^{\text{Mr}}(\mathbf{z}, \mathbf{y}) \end{aligned}$$

L'inégalité triangulaire est vérifiée, on en déduit que la distance de Manhattan relative est bien une distance mathématique.

D.2 LA DISTANCE DE MANHATTAN AMÉLIORÉE

Nous avons souligné dans le chapitre 5 que la distance de Manhattan améliorée n'est pas une distance mathématiquement parlant. C'est par abus de langage que nous la nommons « distance ». Nous allons démontrer ici que cette application n'est pas une distance. Pour cela, il suffit de montrer qu'une des trois propriétés (séparation, symétrie et inégalité triangulaire) n'est pas vérifiée.

Rappelons tout d'abord la définition de cette relation :

La distance de Manhattan améliorée est définie par la relation suivante :

$$\mathcal{D}_j^{\text{Ma}}(\mathbf{t}) = \frac{\sum_{i=1}^n \left| \mathcal{S}_i^{\text{rN}}(\mathbf{t}) \times m' - \tau_{ij} \times n' \right| \cdot \tau_{ij}}{n'}$$

Avec n' le nombre d'éléments non nuls de la signature de défauts théoriques $\tau_{\cdot, j}$,

m' le nombre d'éléments non nuls de la signature de défauts instantanée \mathcal{S}^{rN} .

Soient $\mathcal{S}^{\text{rN}}(\mathbf{t}) \in \mathcal{S}$ et $\tau_{\cdot, j} \in \mathcal{T}$.

On suppose que la distance de Manhattan améliorée entre $\mathcal{S}^{\text{rN}}(\mathbf{t})$ et $\tau_{\cdot, j}$ est nulle :

$$\mathcal{D}_j^{\text{Ma}}(\mathbf{t}) = 0$$

$$\Leftrightarrow \frac{\sum_{i=1}^n \left| \mathcal{S}_i^{\text{rN}}(\mathbf{t}) \times m' - \tau_{ij} \times n' \right| \cdot \tau_{ij}}{n'} = 0$$

$$\Leftrightarrow \sum_{i=1}^n \left| \mathcal{S}_i^{\text{rN}}(\mathbf{t}) \times m' - \tau_{ij} \times n' \right| \cdot \tau_{ij} = 0$$

Or, $\tau_{ij} \in [0; 1], \forall i \in [1; n]$ et $\left| \mathcal{S}_i^{\text{rN}}(\mathbf{t}) - \tau_{ij} \right| \geq 0, \forall i \in [1; n]$.

On en déduit donc que

$$\Leftrightarrow \left| s_i^{rN}(\mathbf{t}) \times \mathbf{m}' - \tau_{ij} \times \mathbf{n}' \right| \cdot \tau_{ij} = 0, \quad \forall i \in [1; n]$$

$$\Leftrightarrow \text{Soit } \left| s_i^{rN}(\mathbf{t}) \times \mathbf{m}' - \tau_{ij} \times \mathbf{n}' \right| = 0 \quad \Rightarrow s_i^{rN}(\mathbf{t}) = \frac{\tau_{ij} \times \mathbf{n}'}{\mathbf{m}'}$$

Soit $\tau_{ij} = 0$

On en déduit que : $s_i^{rN}(\mathbf{t})$ n'est pas forcément égal à $\tau_{i,j}$. On n'a donc pas l'équivalence de séparation et on en conclut que la distance de Manhattan améliorée n'est pas une distance mathématique.

Annexe E : LES STRUCTURES DAEMATRIX ET DAEVECTOR

Cette annexe présente brièvement les structures de DAEMatrix et DAEVector utilisés pour le calcul matriciel au sein du filtre de Kalman étendu.

Le diagramme de classes de la Figure D.1 donne la structure générale pour représenter les vecteurs et les matrices utilisés pour le calcul matriciel au sein du module *PrODHySAEM*.

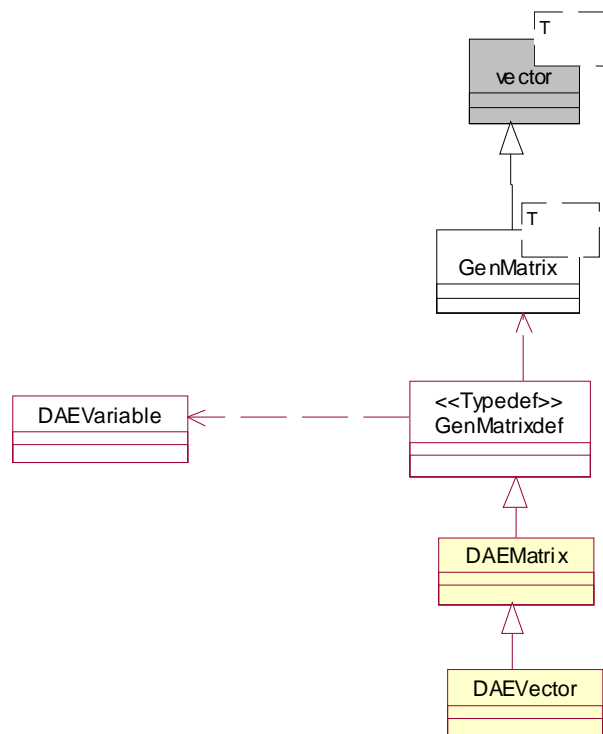


Figure D.1 Le diagramme de classes associé à la structure de matrice et vecteur

La classe `GenMatrix` représente tout type de matrice. Elle hérite de la classe `vector`, classe la plus élevée dans la hiérarchie, provenant de la bibliothèque STL et permettant de mémoriser tout type d'objet séquentiellement. Cette structure permet d'exploiter les méthodes et opérateurs, développés au sein de la bibliothèque STL, de la classe `vector`.

Au sein du module *PrODHySAEM*, les variables utilisées sont de type `DAEVariable`. Ainsi la classe `DAEMatrix` représente une matrice de `DAEVariable`. Elle hérite donc d'une classe instanciée de la

classe GenMatrix (T = DAEVariable). De cette structure découle la classe DAEVector qui représente un vecteur de DAEVariable.

E.1 LA STRUCTURE DAEMATRIX

La Figure D.2 représente le diagramme de classe de la structure DAEMatrix :

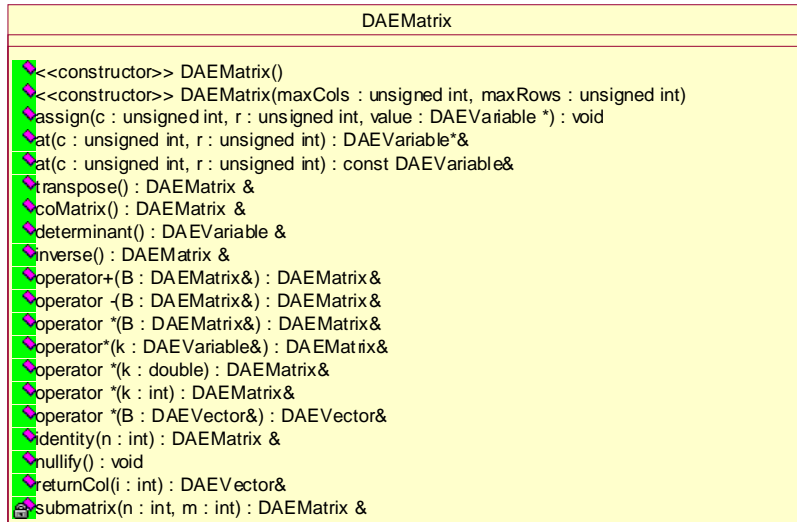


Figure D.2 Le diagramme de classe de la matrice de DAEVariable

La classe DAEMatrix exploite les méthodes développées au sein de la bibliothèque STL. Elle est composée de nombreuses méthodes, fonctions et opérateurs :

- Elle possède deux constructeurs : l'un sans arguments et l'autre avec deux arguments d'entrée : le nombre de colonnes et le nombre de lignes ;
- La méthode assign permet de modifier la valeur de l'élément de colonne c et de ligne r (r et c sont fournies en entrée) ; cette méthode est un accesseur en écriture ;
- La fonction at qui retourne l'élément de colonne c et de ligne r ; cette méthode est un accesseur en lecture. Notons qu'un élément de la matrice est repéré d'abord par le rang de sa colonne puis par le rang de sa ligne, malgré la logique qui voudrait l'inverse.
- La fonction transpose retourne la transposée de la matrice ;
- La fonction determinant calcule le déterminant de la matrice ;
- La fonction inverse évalue l'inverse de la matrice ;
- L'opérateur + additionne deux matrices si elles sont de même dimension ;
- L'opérateur - soustrait deux matrices si elles ont la même dimension ;
- L'opérateur * est surchargé afin d'effectuer différents calculs :
 - La multiplication de deux matrices si le nombre de colonnes de la première est égale au nombre de lignes de la seconde ;

- La multiplication d'une matrice par un vecteur si le nombre de colonnes de la matrice est égale au nombre de lignes du vecteur ;
 - La multiplication d'une matrice par un scalaire de type DAEVariable ;
 - La multiplication d'une matrice par un scalaire de type double ;
 - La multiplication d'une matrice par un scalaire de type int.
- La fonction identity retourne la matrice identité de dimension n (dimension fournie en entrée) ;
 - La méthode nullify permet de transformer la matrice en la matrice nulle ;
 - La fonction returnCol retourne la $i^{\text{ème}}$ colonne de la matrice, avec i paramètre d'entrée (de type DAEVector).

E.2 LA STRUCTURE DAEVECTOR

La Figure D.3 représente le diagramme de classe de la structure DAEVector :

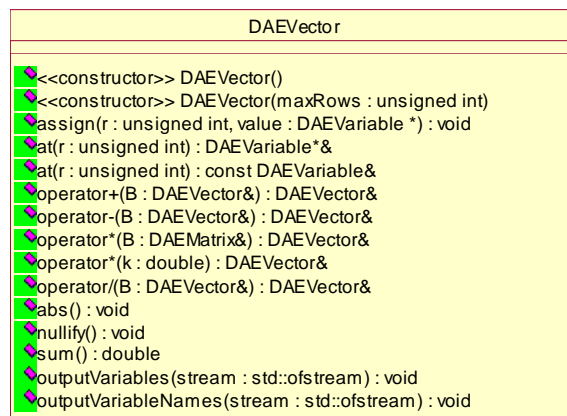


Figure D.3 Le diagramme de classe du vecteur de DAEVariable

La classe DAEVector hérite de la classe DAEMatrix et nécessite donc la redéfinition de certaines méthodes. La structure DAEVector représente un vecteur colonne. Elle est donc constituée des méthodes, fonctions et opérateurs suivants :

- Elle possède deux constructeurs : l'un sans arguments et l'autre avec un seul argument d'entrée : le nombre de lignes du vecteur ;
- La méthode assign permet de modifier la valeur du $r^{\text{ème}}$ élément du vecteur, r valeur de la ligne fournie en entrée (le nombre de colonne est égale à 1) ; cette méthode est un accesseur en écriture ;
- La fonction at qui retourne le $r^{\text{ème}}$ élément du vecteur ; cette méthode est un accesseur en lecture ;
- L'opérateur + additionne deux vecteurs s'ils sont de même dimension ;
- L'opérateur – soustrait deux vecteurs s'ils ont la même dimension ;
- L'opérateur * est surchargé afin d'effectuer différents calculs :

- La multiplication d'un vecteur par une matrice si le nombre de lignes du vecteur est égale au nombre de ligne de la matrice. Ici, le calcul matriciel correspond à la multiplication d'un vecteur ligne par une matrice.
- La multiplication d'un vecteur par un scalaire de type double ;
- La fonction `abs` retourne la valeur absolue de chaque élément du vecteur ;
- La méthode `nullify` permet de transformer le vecteur en vecteur nul ;
- La fonction `sum` retourne la somme de tous les éléments du vecteur.

NOMENCLATURE

Lettres romaines

| | | |
|---------------|---|---------------------------|
| C_k | <i>classe</i> | |
| Cov | <i>matrice de covariance</i> | |
| d | <i>nombre de défauts</i> | |
| D | <i>distance</i> | - |
| D_e | <i>vecteur de défaillances</i> | |
| Dir | <i>matrice de direction des défaillances</i> | |
| e | <i>état discret</i> | |
| f | <i>modèle non linéaire du système</i> | |
| f_i | <i>défaut</i> | |
| F | <i>modèle du système</i> | |
| F | <i>ensemble des défauts</i> | |
| F_{in}^a | <i>flux molaire d'entrée de l'appareil a</i> | <i>mol.s⁻¹</i> |
| F_{out}^a | <i>flux molaire de sortie de l'appareil a</i> | <i>mol.s⁻¹</i> |
| g | <i>constante de gravité</i> | <i>m.s⁻²</i> |
| G | <i>matrice des entrées du système</i> | |
| \mathcal{G} | <i>gain</i> | |
| h | <i>hauteur</i> | |
| h_l | <i>hauteur de liquide</i> | <i>m</i> |
| h_L | <i>enthalpie molaire liquide</i> | <i>J.mol⁻¹</i> |
| H | <i>matrice caractéristique des mesures</i> | |
| I | <i>indicateur de défaut</i> | |
| K_k | <i>matrice gain de Kalman</i> | |
| m | <i>nombre de mesures</i> | |
| mh | <i>modèle d'enthalpie molaire liquide</i> | <i>J.mol⁻¹</i> |
| mH | <i>modèle d'enthalpie molaire vapeur</i> | <i>J.mol⁻¹</i> |
| mP | <i>modèle de pression</i> | |
| n | <i>nombre d'états</i> | |
| \mathcal{O} | <i>observabilité</i> | |
| Op | <i>coefficient d'ouverture</i> | - |
| p | <i>paramètres physiques</i> | |
| p_{in} | <i>port d'entrée</i> | |
| P | <i>pression</i> | <i>Pa</i> |
| $P_{i j}$ | <i>matrice de covariance de l'erreur</i> | |
| P_r | <i>matrice de projection</i> | |
| Q_k | <i>variance des bruits de dynamique</i> | |
| r | <i>résidu</i> | |
| \mathcal{R} | <i>ensemble des résidus</i> | |
| R_k | <i>variance des bruits d'observation</i> | |
| S | <i>signature</i> | |
| S | <i>ensemble des signatures</i> | |
| $S_{t,c}$ | <i>surface (ou section)</i> | <i>m²</i> |
| t | <i>temps</i> | <i>s</i> |

| | | |
|---------------|--|----------------------|
| \mathcal{T} | <i>matrice d'incidence</i> | |
| \mathcal{T} | <i>ensemble des signatures de défauts théoriques</i> | |
| T | <i>température</i> | K |
| v | <i>bruit du système</i> | |
| U | <i>matrice caractéristique des entrées</i> | |
| U_d | <i>univers de discours</i> | |
| U_l | <i>rétention molaire</i> | mol |
| V | <i>vecteur propre</i> | |
| V_k | <i>vecteur de l'erreur de modèle</i> | |
| V_{ml} | <i>volume molaire</i> | $m^3 \cdot mol^{-1}$ |
| w | <i>bruit des mesures</i> | |
| W_k | <i>vecteur des bruits sur les mesures</i> | |
| \hat{x} | <i>estimation</i> | |
| \tilde{x} | <i>erreur d'estimation</i> | |
| \bar{x} | <i>trajectoire nominale</i> | |
| x_i | <i>fraction molaire liquide</i> | - |
| X | <i>vecteur d'état</i> | |

Lettres grecques

| | |
|---------------|--|
| α | <i>résultat du test de seuillage</i> |
| Δ | <i>variation</i> |
| ε | <i>seuil</i> |
| Γ_p | <i>générateur d'évènements</i> |
| λ | <i>valeur propre</i> |
| μ | <i>fonction d'appartenance</i> |
| θ | <i>paramètre du modèle</i> |
| σ | <i>écart type</i> |
| Σ_e | <i>évènements de contrôle du processus</i> |
| Σ_p | <i>évènements physiques</i> |
| ω | <i>poids</i> |

Indices

| | |
|---------|--|
| Ha | <i>Hamming amélioré</i> |
| Hr | <i>Hamming relatif</i> |
| in | <i>entrée</i> |
| $k+1 k$ | <i>prédiction faite à l'instant $k+1$ connaissant seulement l'état à l'instant k ($k+1$ sachant k)</i> |
| min | <i>minimum</i> |
| Ma | <i>Manhattan amélioré</i> |
| max | <i>maximum</i> |
| Mr | <i>Manhattan relatif</i> |
| $order$ | <i>consigne</i> |
| out | <i>sortie</i> |

Exposants

| | |
|------|----------------------|
| rN | <i>relatif normé</i> |
| r | <i>relatif</i> |

Abréviations

| | |
|-------------|---|
| <i>EANL</i> | <i>Équation Algébrique Non Linéaire</i> |
| <i>EDA</i> | <i>Équation Différentielle Algébrique</i> |
| <i>FDI</i> | <i>Fault Detection and Isolation</i> |
| <i>OO</i> | <i>Orienté Objet</i> |
| <i>RdPO</i> | <i>Réseau de Petri Différentiel objet</i> |
| <i>RdP</i> | <i>Réseau de Petri</i> |
| <i>SDH</i> | <i>Système Dynamique Hybride</i> |
| <i>SED</i> | <i>Système à Événements Discrets</i> |

Sigles

| | |
|-------------------|--|
| <i>DISCo</i> | <i>Do Integrate by Software Component</i> |
| <i>FDIPro</i> | <i>Fault Detection and Isolation for Process</i> |
| <i>PrODHyS</i> | <i>Process Object Dynamic Hybrid Simulator</i> |
| <i>PrODHySAEM</i> | <i>Process Object Dynamic Hybrid Simulator</i> |
| <i>SimAEM</i> | <i>Simulation Abnormal Event Management</i> |

BIBLIOGRAPHIE

A

- Abuhadrous I. (2005). *Système embarqué temps réel de localisation et de modélisation 3D par fusion multi-capteur*. Doctorat Informatique temps réel - Robotique - Automatique, Centre de Robotique, ENSMP
- Adjallah K. (1993). *Contribution au diagnostic de systèmes par observateur d'état*. Thèse de doctorat, Institut National Polytechnique de Lorraine, Nancy, France
- AFNOR (1994). *Norme expérimentale X60-010. Maintenance – Concepts et définition des activités et maintenance*. AFNOR, premier tirage, Paris, France
- Aguilar-Martin J., Lopez de Mantaras R. (1982). *The process of classification and learning the meaning of linguistic descriptors of concepts*. Approximate Reasoning in Decision Analysis, North Holland, pp. 165-175
- Albet J. (1992). *Simulation rigoureuse de colonnes de distillation discontinues à séquences opératoires multiples*. Thèse de doctorat, INP, Toulouse, France
- Alla H., Cavaille J.B., Le Bail J., Bel G. (1992). *Les systèmes de production par lot : une approche discret-continu utilisant les réseaux de Petri hybrides*. Automation of Mixed Processes (ADPM 92), Janvier, Paris, France
- Alur R., Courcoubetis C., Halbwachs N., Henzinger T.A., Ho P.H., Nicollin X., A. Olivero, Sifakis J. and Yovine S. (1995). *The algorithmic analysis of hybrid systems*. Theoretical Computer Science, vol. 138, pp. 3-34
- Alur R., Dill D.L. (1994). *A Theory of Timed Automata*. Theoretical Computer Science, vol. 126, n°2, pp. 183-225
- Andersson M. (1994). *Object-Oriented Modelling and Simulation of Hybrid Systems*. Ph.D thesis, Lund Institute of Technology, Sweden

B

- Babuska R. (1998). *Fuzzy Modeling for control*. Kluwer Academic Publishers, Massachusetts, USA
- Balluchi A., Benvenuti L., Sangiovanni-Vincentelli A.L. (2002). *Observers for hybrid systems with continuous state resets*. Proceedings of 10th Mediterranean conference on control and automation – Med'2002, Lisbon, Portugal
- Barton P.I. and Pantelides C.C. (1994). *The Modelling of Combined Discrete/Continuous Processes*. AIChE Journal, 40, pp. 966-979
- Basseville M. (1988). *Detecting changes in signals and systems – A survey*. Automatica, vol. 24 (3), pp. 309-326
- Basseville M. (1997). *Information criteria for residual generation and fault detection and isolation*. Automatica, vol. 33 (5), pp. 783-803
- Bastin G. et Gevers M. (1998). *Stable adaptive observers for nonlinear time varying systems*. IEEE Trans. On Automatic Control 33(7), pp. 650-658
- 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 (ESCAPE 12), 26-29 mai, The Hague, The Netherlands
- Bezdek J.C. (1981). *Pattern recognition with fuzzy objective function algorithms*, Plenum Publishing Corporation, New York, USA
- Birouche A. (2006). *Contribution sur la synthèse d'observateurs pour les systèmes dynamiques hybrides*, Thèse de doctorat, Institut National Polytechnique de Lorraine, Nancy, France
- Bishop C. (1995). *Neural networks for pattern recognition*. Oxford University Press, New York, USA
- Biswas G., Cordier M.O., Lunze J., Travé-Messuyès L., Staroswiecki M. (2004). *Diagnosis of complex systems: Bridging the methodologies of the FDI and DX communities*. IEEE Trans. On Systems, man, and cybernetics – Part B: Cybernetics, vol. 34 (5), pp. 2159-2162

Blanke M., Kinnaert M., Lunze J., Staroswiecki M. (2003). *Diagnosis and fault-tolerant control*. Springer-Verlag

Boillereaux L., Flaus J.M. (2003). *Commande et supervision. Les procédés agroalimentaires 2*. Lavoisier ISBN 2-7462-0755-9

Boufaied A. (2003). *Contribution à la surveillance distribuée des systèmes à événements discrets complexes*. Thèse de doctorat, Université Paul Sabatier - Toulouse III, France

Bourseau P. (1993). *Modélisation des connaissances de l'ingénieur de procédé : application de l'intelligence artificielle*. Doctorat d'État, Université Paris IV, France

Branicky M.S. (1995). *Studies in hybrid systems: Modeling, Analysis and Control*. PhD thesis, MIT, Massachusetts, USA

Buisson J., Cormerais H. (1998). *Descriptor Systems for the Knowledge Modelling and Simulation of Hybrid Physical Systems*. Journal Européen des systèmes automatisés, vol. 32, n°9-10, pp. 1047-1072

C

Casimir R. (2003). *Diagnostic des défauts des machines asynchrones par reconnaissance des formes*. Thèse de doctorat, École Centrale de Lyon, France

Cassar J.P., Litwak R-G., Cocquemot V., Staroswiecki M., (1994). *Approche structurale de la conception de systèmes de surveillance pour les procédés industriels*. Diagnostic et Sécurité de Fonctionnement, vol. 4, n° 2, pp. 179-202

Chantler M.J., Coghil G.M., Shen Q., Leitch R.R. (1998). *Selecting tools and techniques for model-based diagnosis*. Artificial Intelligence in Engineering, 12, pp. 81-98

Champagnat R., Esteban P., Pingaud H., Valette R. (1998). *Modeling and simulation of a hybrid system through Pr-Tr PN-DAE model*. ADPM'98, Reims, France, pp. 131-137

Chen J., Patton R.J., Zhang H.Y. (1995). *Design of robust structured and directional residuals for fault isolation via unknown input observers*. Proceedings of 3rd European Control Conference, vol. 1, pp. 248-353

Chester D., Lamb D., Dhurjati P. (1984). *Rule-based computer alarm analysis in chemical process plants*. Proceedings of 7th Micro-Delcon, pp. 22-29

Cheung J.T., Stephanopoulos G. (1990). *Representation of process trends Part I. A formal representation framework*. Computers & Chemical Engineering, vol. 14 (4-5), pp. 495-510

Chin H., Danai K. (1991). *A method of fault signature extraction for improved diagnosis*. IEEE ACC conference, Boston, USA

Chombart A., Flaus J.M., Valentin-Roubinet C. (1996). *Hybrid Systems Modelling: a comparison of three methods applied to an example*. Congrès IFAC'96, 30 Juin-5 Juillet, San Francisco, USA

Chow E.Y.(1980). *Failure detection system design methodology*. Thèse de doctorat, Laboratoire Information and decision system, M.I.T., Cambridge, USA

Chow E.Y., Willsky A.S. (1984). *Analytical redundancy and the design of robust failure detection system*. in IEEE Trans. Automatic Control, vol. 29(7), pp. 603-614

Clark R.N. (1979). *The dedicated observer approach to instrument fault detection*. Proceedings of the 18th IEEE-CDC, pp. 237-241, Fort Lauderdale, Florida, USA

Clark R.N. (1989). *State estimation schemes for instrument fault detection*. Fault Diagnosis in Dynamic Systems: Theory and application, éd. R. Patton, P. Frank and R. Clark, Prentice Hall

Clark R.N., Fosth D.C., Walton V.M. (1975). *Detection instrument malfunctions in control systems*. IEEE Transactions on Aerospace and Electronic Systems, vol. AES-11, pp. 465-473

Combacau M., Courvoisier M. (1990). *A hierarchical and modular structure for F.M.S. control and monitoring. 1rst conference on Artificial Intelligence, Simulation and Planning in High Autonomy Systems*, Tuscon, Arizona, USA

D

- Dash S., Venkatasubramanian V. (2000). *Challenges in the industrial applications of fault diagnostic systems*. Proceedings of the conference on Process Systems Engineering Comput. & Chem.
- Dash S., Rengaswamy R., Venkatasubramanian V. (2003). *Fuzzy logic based trend classification for fault diagnosis of chemical processes*. Computers & Chemical Engineering, vol. 27, pp. 347-362
- De Kleer J. (1986). *An assumption-based TMS*. Artificial Intelligence, vol. 28, pp. 127-162
- Denoeux T., Zouhal L. M. (2001). *Handling possibilistic labels in pattern classification using evidential reasoning*. Fuzzy Sets and Systems 1(22)., pp. 409-424
- De Persis C., Isidori A. (2001). *A geometric approach to nonlinear fault detection and isolation*. IEEE Transactions on Automatic Control, vol. 46 (6), pp. 853-865
- Deshpande A., Gollu A., Semenzato L. (1998). *The shift programming language for dynamic networks of hybrid systems*. IEEE Trans. Automatic Control special issue on Hybrid Systems
- Desroches P. (1987). *Syclare: Système de classification avec apprentissage et reconnaissance de formes, Manuel d'utilisation*. Rapport de recherche, Centre d'études avancés de Blanes, Espagne
- Diamantaras K.L., Kung S.Y. (1996). *Principal component neural networks*. Theory and applications, John Wiley & Sons
- Ding X., Frank P.M. (1991). *Frequency domain approach and threshold selector for robust model-based fault detection and isolation*. in Proceedings of IFAC Symposium SAFEPROCESS'91, Baden-Baden, Germany, vol. 1, pp. 307-312
- Ding X., Frank P.M. (1994). *Comparison of observer-based fault detection approaches*. SAFEPROCESS'94, IFAC Symposium on Fault Diagnostic, Supervision and Safety for Technical Processes, Espoo, Finland, vol. 2, pp. 556-561
- Dubois D., Gentil S. (1990). *Intelligence Artificielle : un outil pour l'automatique*. Journées annuelles du GR automatique, Octobre
- Dubuisson B. (1990). *Diagnostic et reconnaissance des formes*. Éditions Hermès, France
- Dubuisson B. (2001). *Diagnostic, intelligence artificielle et reconnaissance des formes*. Productique (IC2), Éditions Hermès, Lavoisier ISBN 2-7462-0249-2, Paris
- Duda R.O., Hart P.E. (1973). *Pattern classification and scene classification*. New York, Wiley Interscience

E

- Einicke G.A., White L.B. (1999). *Robust extended Kalman filtering*. IEEE Trans. on Signal Processing, vol. 47 (9), pp. 2596-2599
- Engell S. (1997). *Modelling and analysis of hybrid systems*. 2nd IMACS MATHMOD Conference, pp. 17-31, Vienne, Autriche
- Escobet T., Travé-Massuyès L., Tornill S., Quevedo J. (2001). *Fault detection of a gas turbine fuel actuator based on qualitative causal models*. European Control Conference (ECC'01)., Porto, Portugal, pp. 2741-2746
- Evensen G., (2003). *The ensemble Kalman filter: theoretical formulation and practical implementation*. Ocean Dynamics, vol. 53, pp. 343-367

F

- Fabián G., van Beek D.A. and Rooda J.E. (1998). *Integration of the Discrete and the Continuous Behaviour in the Hybrid Chi Simulator*. European Simulation Multiconference, Manchester
- Fang C. Z., Ge W. (1998). *Failure isolation in linear systems*. IMACS 12th world congress, Paris, France, pp. 442-446

- Flandrin P. (1998). *Temps – Fréquence*. Traité de Traitement du Signal, Hermès, Paris
- Flaus J.M. (1998). *Modeling and Analysis of Hybrid Dynamical Systems: an Introduction*. Journal Européen des Systèmes Automatisés (JESA), vol. 32, n°7-8, pp. 797-830
- Florin G., Fraize C., Natkin S. (1991). *Stochastic Petri Nets: Properties, Applications and Tools*. Microelectronics and Reliability, vol. 31, n°4, pp. 669-697
- Frank P.M. (1987). *Fault diagnosis in dynamic systems via state estimation – a survey*. Tzafestas S., Singh M., Schmidt G. (Eds.), Systems fault diagnostics, reliability and related knowledge-based approaches, vol. 1, pp. 35-98
- Frank P.M. (1996). *Analytical and qualitative model-based fault diagnosis – A survey and some new results*. European Journal of Control (EUCA), vol. 2, pp. 6-28
- Frank P.M. (1998). *The application of fuzzy logic to fault diagnosis and supervision*. LFA, pp. 59-87
- Frank P.M., Köppen-Seliger B. (1997). *New developments using AI in fault diagnosis*, Engineering. Applic. Artif. Intell., vol. 10(1), pp. 3-14
- Frank P.M., Wünnenberg J. (1989). *Robust diagnosis using unknown input observer schemes*. Fault Diagnosis in dynamic systems: theory and application, Patton R.J., Frank P.M., Clark R.N., Prentice Hall, Englewood, Cliffs, pp. 47-98
- Fussel D., Isermann R. (1998). *Model-based fault detection and diagnosis methods*. Part A: methods, 4th International Conference on Probabilistic Safety Assessment and Management, PSAM 4, New York, USA

G

- Gani R., Braunschweig B.L. (2002). *Software architectures and tools for computer aided process Engineering*. Elsevier, ISBN :0-444-50827-9
- Gauthier J.P., Hammouri H., Othman S. (1992). *A simple observer for nonlinear systems applications to bioreactors*. IEEE Trans. Automat. Contr., vol. 37, pp. 875-880
- Gauthier J.P., Kupka I. (2001). *Deterministic observation theory and applications*. Cambridge university press
- Gear C.W. (1971). *The Simultaneous Numerical Solution of Differential-Algebraic Equations*. IEEE Transaction on Corcuit Theory, CT 18 (1), Ed. Academic Press
- Gehin A., Assas M., Staroswiecki M. (2000). *Structural Analysis of System Reconfigurability*. 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS'00), Budapest, Hungary, pp. 292-297
- Gentil S. (1997). *Filtrage d'alarmes en temps réel*. Rapport de Recherche convention DRET
- Gentil S. (2003). *Commande et supervision – les procédés agroalimentaires 2*. Traité IC2 Systèmes Automatisés, Éditions Hermès, Chapitre 4 : Supervision et diagnostic, pp. 101-135
- Gentil S., Montmain J., Combastel C. (2004). *Combining FDI and AI approaches within causal-model-based diagnosis*. IEEE Trans. on Systems, man, and Cybernetics, vol. 34(5), pp. 2207-2221
- Gertler J. (1991). *Analytical redundancy methods in failure detection and isolation*. SAFEPROCESS'91, Baden-Baden, Germany, pp. 9-21
- Gertler J. (1992). *Analytical redundancy methods in fault detection and isolation-survey and synthesis*. IFAC symposium on online fault detection and supervision in the chemical process industries
- Gertler J., Singer D. (1990). *A new structural framework for parity equation based failure detection and isolation*. Automatica, vol. 26(2), pp. 381-388
- Guéguen, H. and Lefebvre, M. A. (2001). *A comparison of mixed specification formalisms*. Journal Européen des Systèmes Automatisés (APII JESA), vol. 35, n°4, pp. 381-394
- Guerraz, B. (2005). *Construction de chroniques à partir d'une modélisation du système. Application au diagnostic de réseaux de télécommunications*. Thèse de Doctorat, Université de Rennes 1, France

Gustafson D.E. (1979). *Fuzzy clustering with a fuzzy covariance matrix*. Proceedings of IEEE CDC, San Diego, USA, pp. 761-766

H

Harkat M.F. (2003). *Détection et Localisation de Défauts par Analyse en Composantes Principales*. Thèse de doctorat, Institut National Polytechnique de Lorraine, Nancy, France

Hengy D., Frank P.M. (1986). *Component failure detection via nonlinear state observers*. Proceedings of the IFAC Workshop on Fault Detection and Safety in Chemical Plants, Kyoto, Japan, pp. 153-157

Henley E.J. (1984). *Application of expert systems to fault diagnosis*. in AIChE annual meeting, San Francisco, CA

Hétreux G., Perret J., Le Lann J.M. *Bibliothèque orientée objet pour la conception de simulateurs dynamiques hybrides*. Congrès Français de Génie des Procédés (CFGP'2003), 9-11 Septembre, Saint-Nazaire, 2003

Hétreux G., Théry R., Olivier N., Le Lann J.M. (2007). *Exploitation de la simulation dynamique hybride pour la conduite de procédés semi-continus*. Journal Européen des Systèmes Automatisés JESA, Supplément MOSIM, vol. 41, n°5, pp. 585-616

I

IEEE (1988). *Standard Dictionary of electrical and electronics terms*. 4th edition, F. Jay (ed.), The Institute of Electrical and Electronics Engineers, New York, IEEE, 1988

Iri M., Aoki K., O'Shima E., Matsuyama H. (1979). *An algorithm for diagnosis of system failures in chemical processes*. Computers and Chemical Engineering, vol. 3, pp. 489-493.

Isermann R. (1984). *Process Fault diagnosis based on modelling and estimation methods – A survey –*. Automatica, vol. 20, pp. 387-404

Isermann R. (1993). *Fault diagnosis on machines via parameter estimation and knowledge processing*. Automatica, vol. 29, pp. 815-836

Isermann R. (1994). *Integration of fault detection and diagnosis methods*. Proceedings of the IFAC Symposium on Fault Detection, Supervision and Safety for Technical Industrial Processes, SAFEPROCESS'94, Helsinki, Finland, pp. 597-612

Isermann R. (1997). *Supervision, fault detection and fault diagnosis methods – an introduction*. Control Eng. Practice, vol. 5 (5), Elsevier, pp. 639-652

Isermann R., Freyermuth B. (1991). *Process fault diagnosis based on process model knowledge*. Parts I (Principles for fault diagnosis with parameter estimation), and Parts II (Case study experiments), ASME J, Dynamic Systems, Measurement Control, pp. 620-633

Isermann R., Phalle P. (2000). *Applied terminology of fault detection, supervision and safety for technical processes*. Site Internet de IFAC Symposium on Fault Detection Supervision and Safety for Technical Process

Iwasaki Y., Simon H.A. (1986). *Causality in device behaviour*. Artificial Intelligence, vol. 29 (1), pp. 3-33

J

Jazwinski A.H. (1970). *Stochastic Processes and Filtering Theory*. New York, Academic Press, Mathematics in Science and Engineering, vol. 64

Joglekar G.S., Reklaitis G.V. (1985). *A simulator for batch and semi-continuous processes*. Computers and Chemical Engineering, vol. 8, n°6, pp. 315-327

Jolliffe I.T. (1986). *Principal components analysis*. Springer-Verlag Press

Jourda, L. (1996). *Composants Logiciels Orientés Objets pour la Modélisation et la Simulation des Procédés Chimiques*. Thèse de doctorat, INP, Toulouse, France

Jourda L. Joulia X. and Koehret B. (1996). *Introducing ATOM, the Applied Thermodynamic Object-Oriented Model*. Computer & Chemical Engineering, 20A, S157-S164

K

Kalman R.E. (1960). *A new approach to linear filtering and prediction problems*. Trans. ASME, Journal of basic Engineering, vol. 82, pp. 34-45

Kalman R.E. (1963). *Mathematical description of linear dynamical systems*. SIAM Journal of Control and Optimization 1, pp. 152-192

Kalman R.E., Bucy R. (1961). *A new approach to linear filtering and prediction theory*, in Trans. ASME, Journal of basic Engineering, vol. 83, pp. 95-108

Kaufmann A., (1977). *Introduction à la théorie des sous-ensembles flous à l'usage des ingénieurs*. Tomes I et II, Masson

Kempowski T. (2004). *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

Kesten Y., Pnueli A. (1992). *Timed and hybrid statecharts and their textual representation*. Lecture Notes in Computer Science (LNCS), vol. 571

Koscielny J.M., (1993). *Method of fault isolation for industrial processes*. Diagnostic et sûreté de fonctionnement, vol. 3, n° 2, pp. 205-220

Krener A.J. (1984). *Approximate linearization by state feedback and coordinate changes*. in System Control Letter, vol. 5, pp. 181-185

Kuipers B. (1985). *The limits of qualitative simulation*. in Proceedings of ninth joint international conference on artificial intelligence

Kuipers B. (1986). *Qualitative simulations*. in Artificial Intelligence, vol. 29(3), pp. 289-338

L

Landau I.D. (2001). *Identification des systèmes. Les bases de l'identification des systèmes*. 3. Partie d'ouvrage

Laser M. (2000). *Recent safety and environmental legislation*, Trans IchemE. vol. 78 (B), pp. 419-422

Le Bail J., Alla H. and David R. (1991). *Hybrid Petri Nets*. Proceedings of the European Control Conference, pp. 1472-1477, Grenoble

Lefebvre D. (2000). *Contribution à la modélisation des systèmes dynamiques à événements discrets pour la commande et la surveillance*. Habilitation à Diriger des Recherches, Université de Franche Comté/IUT Belfort – Montbéliard

Le Lann J.M. (1999). *Des mathématiques à la simulation dynamique robuste des procédés : le traitement algèbro-différentiel des équations EDA*. Habilitation à Diriger les Recherches, INP, Toulouse, France

Ljung L. (1987). *System identification: theory for the users*. Prentice Hall Information and System Sciences Series, 2nd edition 1999

Lou X.C., Willsky A.S., Verghese G.C. (1986). *Optimally robust redundancy relations for failure detection in uncertain systems*. in Automatica, vol. 22 (3), pp. 333-344

Luenberger D.G. (1971). *An introduction to observers*. IEEE Trans. on Automatic Control, vol. 16 (6), pp. 596-602

M

- MacGregor J.F., Kourti T. (1995). *Statistical process control of multivariate processes*. Control Engineering Practice 3 (3), pp. 403-414
- MacQueen J. (1967). *Some methods for classification and analysis of multivariate observations*. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California, Berkeley, USA, pp. 281-297
- Maget V. (2007). *Développement et comparaison de méthodes d'assimilation de données appliqués à la restitution de la dynamique des ceintures de radiation de la Terre*. Thèse de doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France
- Magni J.F., Mouyon P. (1991). *A generalized approach to observers for fault diagnosis*. Proceedings of the 30th IEEE-CDC, vol. 3, pp. 2236-2241, December 11-13, Brighton, UK
- Mamdani E. (1977). *Application of fuzzy logic to approximate reasoning using linguistic system*. in Fuzzy sets and systems, vol. 26, pp. 1182-1191
- Maquin D., Cocquemot V., Cassar J.P., Staroswiecki M., Ragot J. (1997). *Generation of analytical redundancy relations for FDI purposes*. IEEE Int. Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives, SDEMPED'97, Carry-le-Rouet, France, pp. 270-276
- Maquin D., Ragot J. (1993). *Diagnostic de fonctionnement des systèmes à partir de modèles*. 4^{ème} Conférence Internationale de Génie des Systèmes Industriels, GSI 4, France
- Marie-Joseph I. (2003). *Méthodologie de diagnostic appliquée à la maintenance préventive d'unités de production d'électricité en sites isolés*. Thèse de doctorat de l'Université des Antilles et de la Guyane
- Marino R., Tomei P. (1995). *Nonlinear control design.*, London, New York, Prentice Hall, Information and system sciences
- Massoumnia M. (1986). *A geometric approach to the synthesis of failure detection filters*. IEEE Transaction on Automatic Control, vol. 31 (9), pp. 839-846
- Mattsson S.E., Andersson M., Åström K.J. (1993). *Object-Oriented Modelling and Simulation*. Linkens, Ed., CAD for Control Systems, Chapitre 2, pp. 31-69, Marcel Dekker Inc, New York.
- McCulloch W.S., Pitts W. (1943). *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Math. Biophysics, vol. 5
- Merlin P. M., Segall A. (1976). *Recoverability of Communication Protocols-Implications of a Theoretical Study*. IEEE Transactions on Communications
- Mohinder SG, Lawrence RW, Angus PA (2001). *Global positioning systems, inertial navigation, and integration*. Wiley Interscience, Canada
- Mokhtari A. (2007). *Diagnostic des systèmes hybrides : développement d'une méthode associant la détection par classification et la simulation dynamique*. Thèse de doctorat, Institut National des Sciences Appliquées, Toulouse, France
- Moler C. (1998). *Are we there yet? Zero crossing and event handling for differential equations*. Simulink 2 special edition, Matlab, News and Notes, pp. 16-17
- Moyse, A. (2000). *Odysseo : plate-forme orientée-objet pour la simulation dynamique des procédés*. Thèse de Doctorat, INP de Toulouse, France
- Muller P.A., Gaertner N. (2000). *Modélisation objet avec UML*. Éditions Eyrolles

N

- Nakoula Y. (1997). *Apprentissage des modèles linguistiques flous, par jeu de règles pondérées*. Thèse de Doctorat, Université de Savoie, France
- Neves F., Martin J.A. (2000). *Heterogeneous control and qualitative supervision, application to a distillation column*. Engineering Applications of Artificial Intelligence, vol. 13, pp. 179-197
- Niida K. (1985). *Expert system experiments in processing engineering*. Institution of chemical engineering symposium series, pp. 529-583.

Nikoukhah R. (1998). *A new methodology for observer design and implementation*. IEEE Trans. on Automatic Control, vol. 43(2), pp.229-231

Nilsson B. (1993). *Dynamic modeling of chemical processes using Omola*. IchEmE Symposium Series, vol. 133, pp. 103-110

Nimmo I. (1995). *Adequately address abnormal situation operations*. Chemical Engineering Progress, vol. 91, n°9, pp. 36-45

Nomikos P., MacGregor J. (1994). *Monitoring batch processes using multiway principal component analysis*. American Institute of Chemical Engineers Journal 40 (8), pp. 1361-1375

O

Oja E., Ogawa H., Wangviwattana J. (1992). *Principal component analysis by homogeneous neural networks, part I & part II: the weighted subspace criterion*. IEICE Transactions INF & Syst., vol. E75-D (3), pp. 366-381

Olivier N., Hétreux G., Le Lann J.M. (2005). *Formal modelling and simulation for control of batch processes*. Conference on Conceptual Modelling and Simulation CMS'05, Marseille, France

Olivier N., Hétreux G., Le Lann J.M., Le Lann M.V. (2006). *Use of an Object Oriented Dynamic Hybrid Simulator for the Monitoring of Industrial Processes*. Proceedings of ADHS'06, Alghero, Italia, pp. 235-240

Olivier-Maget N., Hétreux G., Le Lann J.M., Le Lann M.V. (2007). *Fault detection using a hybrid dynamic simulator: Application to a hydraulic system*. International Modeling and Simulation Multiconference CMS'07, Buenos Aires, Argentina, 8-10 Février

Orantes Molina A. (2005). *Méthodologie pour le placement de capteurs à base de méthodes de classification en vue de son diagnostic*. Thèse de doctorat, INSA, Toulouse, France

P

Patton R.J. (1994). *Robust model-based fault diagnosis: the state of the art*. Proceedings of IFAC Symp. On Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'94, Espoo, Finland, pp. 1-24

Patton R.J., Chen J. (1991a). *A review of parity space approaches to fault diagnosis*. Proceedings of IFAC conference on Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, pp. 65-81

Patton R.J., Chen J. (1991b). *A re-examination of the relationship between parity space and observers-based approaches in fault diagnosis*. Revue Européenne Diagnostic et Sûreté de Fonctionnement, vol. 1, pp. 183-200

Patton R.J., Frank P.M., Clark R.N. (1989). *Fault diagnosis in dynamic systems – Theory and applications*. Prentice Hall

Patton R.J., Frank P.M., Clark R.N. (2000). *Issues of fault diagnosis for dynamical systems*. Springer, England

Perret J. (2003). *Intégration des Réseaux de Petri Différentiels à Objets dans une plate-forme de simulation dynamique hybride : application aux procédés industriels*. Thèse de Doctorat, INP, Toulouse, France

Perret J., Hétreux G., Le Lann J.M. (2004). *Integration of an object formalism within a hybrid dynamic simulation environment*. Control Engineering Practice (Elsevier), 12 10, pp. 1211-1223

Piela P.C. (1989). *ASCEND : an object-oriented computer environment for modelling and analysis*. PhD Thesis, Carnegie-Mellon University, USA

Piera N., Aguilar-Martin J. (1989). *LAMDA: An incremental conceptual clustering method*. Rapport technique LAAS-CNRS n°89420, Décembre

Ploix S. (1998). *Diagnostic des systèmes incertains. Approche bornante*. Thèse de l'Université Henri Poincaré, CRAN, Nancy 1, France

Pons R., Travé-Massuyès L. (1997). *Causal Ordering for multiple mode system*. 11th International Workshop on Qualitative Reasoning, Cortona, Italie

Potter J.E., Suman M.C. (1977). *Thresholdless redundancy management with arrays of skewed instruments*. Electronic flight control systems, Agardograph 224, pp. 2115-2121

R

Ragot J., Maquin D. (1996). *Techniques récentes de diagnostic de systèmes et rôle de l'instrumentation*. Congrès de la Société de l'Industrie Minérale, Montpellier, France

Ragot J., Maquin D. (1998). *Techniques récentes de diagnostic de systèmes et rôle de l'instrumentation, Séminaire de l'École Doctorale PROMEN : Quelle métrologie pour le XXIème siècle ?* Nancy, France, in Récents Progrès en Génie des Procédés, vol. 12 (60), pp. 35-52

Rahmani A., Sueur C., Dauphin-Tanguy G. (1997). *Approche des bond graphs pour l'analyse structurelle des systèmes linéaires*. Linear Algebra and its Applications, vol. 259, pp. 101-131

Recht J.L. (1966). *Failure mode and effect*. National Safety Council.

Reif K., Unbehauen R. (1999). *The extended Kalman Filter as an exponential observer for nonlinear systems*. IEEE Trans. on Signal Processing, vol. 47 (8), pp. 2324-2328

Rengaswamy R., Hagglund T., Venkatasubramanian V. (2001). *A qualitative shape analysis formalism for monitoring control loop performance*. Engineering Applications of Artificial Intelligence, vol. 14, pp.23-33

Ribes J.C., Delaunay J., Delvaux J., Merie E., Mouillet M. (2002). *Diagnostic par reconnaissance de formes de l'état de fonctionnement de l'accélérateur AIRIX*. Journal Européen des Systèmes Automatisés, APII-JESA

Ripoll P. (1999). *Conception d'un système de diagnostic flou appliqué au moteur automobile*. Thèse de doctorat, Université de Savoie, France

Rocha-Loures E. (2006). *Surveillance et diagnostic des phases transitoires des systèmes hybrides basés sur l'abstraction des dynamiques continues par réseau de pétri temporel flou*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France

S

Saporta G. (1990). *Probabilités, analyse des données et statistique*. Éditions technip

Sargousse, A.(1999). *Noyau numérique orientée-objet dédié à la simulation des systèmes dynamiques hybrides*. Thèse de Doctorat, INPT, France

Sauter D., Hamelin F., Ghetie M.A., Fonte C. (1996). *Adaptative residual evaluation of model uncertainties: some methods and results*. Engineering Simulation, vol. 13, pp.773-788

Schmidt S. F. (1966). *Applications of state space methods to navigation problems*. C. T. Leondes, editor, Advanced Control Systems, vol. 3, pp. 293-340, Academic Press

Serain D. (2001). *Enterprise Application Integration – L'architecture des solutions e-business*. Dunod

Sifakis, J. (1977). *Use of Petri Nets for Performance Evaluation, Measuring, Modelling and Evaluating Computer Systems*. pp. 75-93. Amsterdam, North Holland

Söderström Y. et Stoica P. (1989). *System identification*. Prentice Hall International Series in Systems and Control Engineering

Stephanopoulos G., Henning G., Leone H. (1990). *MODEL.LA: a modelling language for process engineering. I- the formal framework*. Computers and Chemical Engineering, vol. 14, 8, pp. 813-846

Sundarraman A., Srinivasan R. (2003). *Monitoring transitions in chemical plants using enhanced trend analysis*. Computers and Chemical Engineering, vol. 27, pp. 1455-1472

T

Takagi T., Sugeno M. (1985). *Fuzzy identification of systems and its application to modelling and control*. IEEE Trans. on systems, man and cybernetics, vol. 15 (1), pp. 116-132

Theillol D., Weber P., Ghetie M., Noura H. (1995). *A hierarchical fault diagnosis method using a decision support system applied to a chemical plant*. IEEE International Conf. on Systems Man and Cybernetics, Canada

Thévenon L. (2000). *Représentation des Systèmes Hybrides Complexes par Flux de Données : Développement d'un Outil de Modélisation et de Simulation des Procédés Batch*. Thèse de doctorat, INP, Grenoble, France

Tittus, M. (1995). *Control Synthesis for Batch Processes*. Thèse de doctorat, Chalmers Univ. of Technology, Sweden

Touaf S. (2005). *Diagnostic logique des systèmes complexes dynamiques dans un contexte multi-agent*. Thèse de doctorat de l'Université Joseph Fourier, Grenoble I, France

Travé-Massuyès L., Dague P., Guerrin F. (1997). *Le raisonnement qualitatif pour les sciences de l'ingénieur*. Éditions Hermès, Paris

Travé-Massuyès L., Milne R. (1997). *TIGERTM: gas turbines condition monitoring using qualitative model based diagnosis, in IEEE Expert*. Intelligent systems & their applications, vol. 12 (3), pp. 22-31

V

Vanbergen J.P. (1994). *Vue générale des problèmes de l'automatisation des batches, Ibra éditeur, Automatisation des processus mixtes : les systèmes dynamiques hybrides*. ADPM'94, pp. 265 – 270, Bruxelles, Belgique, Novembre

Vedam H., Venkatasubramanian V. (1997a). *Signed digraph based multiple fault diagnosis*. Computers and Chemical Engineering, vol. 21, pp. S655-S660

Vedam H., Venkatasubramanian V. (1997b). *A wavelet theory-based adaptive trend analysis system for process monitoring and diagnosis*. American Control Conference, pp. 309-313

Venkatasubramanian V., Rengaswamy R., Yin K., Kavuri S. N., (2003). *A review of process fault detection and diagnosis*. Computers & Chemical Engineering (Elsevier), vol. 27, pp. 293-346

Vilain L., Petit B., Cassar J.P., (1996). *Surveillance du serrage hydraulique d'une cage de laminoir*. AGI'96 Forum des Doctorants Automatique Génie informatique Image, pp. 87-90, Tours, France

Villemeur A. (1988). *Sûreté de fonctionnement des Systèmes Industriels – Fiabilité, Facteurs Humains, Informatisation*. Collection de la Direction des études et recherches d'Électricité de France, Eyrolles, Paris

W

Walter E., Pronzato L. (1997). *Identification of parametric models from experimental data*. Masson

Watanabe K., Himmelblau D.M. (1984). *Incipient fault diagnosis of nonlinear processes with multiple causes of faults*. Chemical Engineering Science vol. 39 (3), pp. 491-508

Weiss J.L. (1988). *Threshold computations for detection of failures in SISO systems with transfer function errors*. Proceedings of the American Control Conference, pp. 2213-2218

Willsky A.S. (1976). *A survey of design methods for failure detection in dynamic systems*. Automatica, vol. 12, pp. 601-611

Wöllhaf K., Fritz M., Schulz C. and Engell S. (1996). *BaSiP – Batch Process Simulation With Dynamically Reconfigured Process Dynamics*. Supplement to Computers and Chemical Engineering, 20 (972), pp. 1281-1286

X

Xu A. (2002). *Observateurs adaptatifs non-linéaires et diagnostic de pannes*. Thèse de doctorat, Université de Rennes 1, Rennes, France

Xu A., Zhang Q. (2004). *Nonlinear system fault diagnosis based on adaptive estimation*.

Z

Zadeh L. (1965). *Fuzzy sets*. Information Control, vol. 8, pp. 338-353

Zaytoon J. (2001). *Systèmes dynamiques hybrides*. Hermès Sciences publications

Zhang, Q. (1999). *Identification et surveillance de systèmes dynamiques*. Habilitation à diriger des recherches, université de Rennes 1, January

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

**LISTE DES PUBLICATIONS ET
COMMUNICATIONS**

Le travail réalisé au cours de ces trois années a donné lieu à plusieurs communications dans des congrès nationaux et internationaux ainsi qu'à une publication parue et à une publication en deuxième revue.

- REVUE INTERNATIONALE (AVEC COMITÉ DE LECTURE)
 - [1] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.
Integration of a failure monitoring within a hybrid dynamic simulation environment
Chemical engineering and processing, 2008, doi:10.1016/j.cep.2007.12.009 (à paraître)
 - [2] HETREUX G., HETREUX-THERY R., OLIVIER N., LE LANN J.M.
Exploitation de la simulation dynamique hybride pour la conduite de procédés semi-continus
Journal Européen des Systèmes Automatisés JESA, 2007, Supplément MOSIM, Vol.41 N°5, pp. 585-616

- CONFÉRENCE INTERNATIONALE AVEC ACTES AVEC COMITÉ DE SÉLECTION ET ACCEPTATION SUR TEXTE COMPLET
 - [3] HETREUX G., HETREUX-THERY R., OLIVIER N., LE LANN J.M.
Environnement de modélisation et simulation dynamique hybride : application à la supervision des procédés batch
6ème congrès international de Génie Industriel ; Besançon (France) ; 7-10 Juin 2005
 - [4] OLIVIER N., HETREUX G., LE LANN J.M.
Formal modelling and simulation for control of batch processes
Conceptual Modeling and Simulation Conference CMS'05 ; Marseille (France), 20-22 Octobre 2005
 - [5] OLIVIER N., HETREUX-THERY R., HETREUX G., LE LANN J.M.
Decision-Making tool for scheduling of batch processes: the dynamic hybrid simulation kernel
ESCAPE ; Garmisch-Partenkirchen (Germany) ; 9-13 Juillet, 2006
 - [6] OLIVIER N., HETREUX G., LE LANN J.M.
Use of an object oriented dynamic hybrid simulator for a model based diagnosis system
2nd IFAC Conference on Analysis and Design of Hybrid Systems ADHS'06 ; Alghero, Sardinia (Italy) ; 7-9 Juin 2006
 - [7] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.
Fault detection using a hybrid dynamic simulator: Application to a hydraulic system
International Modeling and Simulation Multiconference CMS'07 ; Buenos Aires (Argentina) ; 8-10 Février 2007
 - [8] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.
Fault Detection Filter Applied To Hybrid Dynamic System Monitoring
Conference on Systems and Control CSC'07 ; Marrakech (Maroc) ; 16-18 Mai 2007
 - [9] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.
Robust fault detection for the hybrid dynamic systems

4th International Federation of Automatic Control Conference IPAC MCPL 2007 ; Sibiu (Romania) ; 27-30
Septembre 2007

[10] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.

Failure monitoring of a Hybrid Dynamic System Monitoring using the extended Kalman Filter

5th International Conference on Integrated Design and Production CPI'07 ; Rabat (Maroc) ; 22-24 Octobre
2007

[11] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.

Fault detection and isolation based on the model-based approach: Application on chemical processes

ESCAPE ; Lyon (France) ; 1-4 Juin, 2008

• CONFÉRENCE NATIONALE AVEC ACTES AVEC COMITÉ DE SÉLECTION ET
ACCEPTATION SUR TEXTE COMPLET

[12] OLIVIER N., HETREUX G., LE LANN J.M

Supervision d'un procédé batch avec un simulateur dynamique hybride

Journées Doctorales et Nationales du GDRMACS, JDMACS-JNMACS 2005; Lyon (France) ; 5-7
Septembre 2005

[13] HETREUX G., OLIVIER N., HETREUX-THERY R., LE LANN J.M.

Exploitation de la simulation dynamique hybride pour la conduite des procédés semi-continus

6ème Conférence Francophone de Modélisation et Simulation MOSIM'06 ; Rabat (Maroc) ; 3-5 Avril 2006

[14] OLIVIER N., HETREUX G., LE LANN J.M., LE LANN M.V.

Conduite des Systèmes Dynamiques Hybrides : Application à la supervision des procédés industriels

Systèmes d'Information, Modélisation, Optimisation SIMO'06 ; Toulouse (France) ; 11-12 Octobre 2006

[15] OLIVIER-MAGET N., HETREUX G., LE LANN J.M., LE LANN M.V.

Modélisation et simulation de procédés défectueux pour des études de surveillance (article soumis

7ème Conférence Francophone de Modélisation et Simulation MOSIM'08 ; Paris (France) ; 31 Mars - 2
Avril 2008

• DIVERS

[16] OLIVIER N., LE LANN J.M., LE LANN M.V.

Vers une approche de diagnostic des systèmes dynamiques Hybrides

7ème congrès des doctorants EDSYS ; Pau (France) ; 11 Mai 2006

SURVEILLANCE DES SYSTÈMES DYNAMIQUES HYBRIDES : APPLICATION AUX PROCÉDÉS

Résumé :

Ces travaux présentent une méthodologie de détection et localisation de défauts pour la surveillance des Systèmes Dynamiques Hybrides. La méthodologie développée repose sur une approche mixte qui combine une méthode à base de modèles pour la détection de fautes et une approche à partir de données de reconnaissance de formes pour l'identification de la ou les fautes(s). Elle se décompose en trois grandes étapes :

- La première étape consiste en la reconstruction de l'état à partir d'un filtre de Kalman étendu et en la génération de résidus par comparaison de l'état reconstruit à celui obtenu par la simulation en parallèle du modèle de référence ;
- La seconde étape réside dans l'exploitation des résidus générés précédemment pour la construction d'une structure plus synthétique, les signatures (non binaires);
- La troisième et dernière étape s'apparente à la résolution d'un problème de reconnaissance de formes dans lequel la signature générée est comparée à une matrice d'incidence (signatures de défauts théoriques) au sens d'une distance.

Cette approche a été intégrée au sein de la plate-forme de simulation PrODHyS, au travers du développement du module PrODHySAEM. Son utilisation est illustrée par l'étude de problèmes de diagnostic dans le domaine des procédés.

Mots clés :

Détection et diagnostic de défauts, Systèmes Dynamiques Hybrides, Génération de signatures non binaires, Distance de Manhattan, Filtre de Kalman étendu, Réseau de Petri Différentiels Objets

MONITORING OF HYBRID DYNAMIC SYSTEMS: APPLICATION TO PROCESSES

Abstract:

These works present a fault detection and isolation methodology for the monitoring of Hybrid Dynamic Systems. The developed methodology rests on a mixed approach which combines a model-based method for the fault detection and an approach based on data (pattern matching) for the identification of fault(s). It is divided into three parts:

The first part concerns the reconstruction of the state of the system, thanks to the Extended Kalman Filter and the generation of the residuals by comparison between the predicted behavior (obtained thanks to the simulation of the reference model) and the real observed behavior (estimated by the Extended Kalman Filter).

The second part exploits these residuals for the generation of a synthetic structure: the non binary signatures.

The last part deals with the diagnosis of the fault and is based on a problem of pattern matching: the signature obtained in the previous part is compared with the theoretical fault signatures by means of distance.

This methodology is integrated within the simulation platform PrODHyS, through the development of the module PrODHySAEM. Its use is illustrated by the studies of diagnosis problems in the field of Chemical Process System Engineering.

Keywords:

Fault detection and diagnosis, Hybrid Dynamic Systems, Generation of non binary signatures, Manhattan distance, Extended Kalman filter, Object Differential Petri nets