DIAGNOSTIC LOGIQUE DES SYSTEMES COMPLEXES ET DYNAMIQUES DANS UN CONTEXTE MULTI-AGENT

Samir Touaf

UNIVERSITE JOSEPH FOURIER - GRENOBLE 1

UNIVERSITE JOSEPH FOURIER
SCIENCES. TECHNOLOGIE. MEDECINE. GRENOBLE - ALPES

N° attribué par la bibliothèque

|__|__|__|__|__|__|__|__|__|__|

THESE

Présentée par

Samir TOUAF

Pour obtenir le titre de

Docteur de L'université Joseph Fourier – Grenoble 1

Spécialité : Automatique – Productique

DIAGNOSTIC LOGIQUE DES SYSTEMES COMPLEXES DYNAMIQUES DANS UN CONTEXTE MULTI-AGENT

_____

Directeur de thèse : M. Jean-Marie FLAUS

_____

Soutenance publique le 02 mars 2005 devant le jury

Président	M^{me}	Sylviane GENTIL	Professeur INPG, LAG
Rapporteur	M	Jean-Philippe CASSAR	Professeur de l'Université de Lille 1, LAIL
Rapporteur	M^{me}	Louise TRAVE-MASSUYES	Directeur de recherche CNRS, LAAS
Examinateur	M.	Philippe DAGUE	Professeur de l'Université de Paris 13, LIPN
Examinateur	M^{me}	Sylvie PESTY	HDR INPG, Leibniz- IMAG
Directeur de thèse	M	Jean-Marie FLAUS	Professeur de l'Université Joseph Fourier, LAG
Codirecteur de thèse	M	Stéphane PLOIX	Maître de conférence INPG, LAG

LABORATOIRE D'AUTOMATIQUE DE GRENOBLE

Thèse préparée au sein du Laboratoire d'Automatique de Grenoble.

Remerciements

Je tiens à remercier tout particulièrement mes directeurs de thèse :

Monsieur Jean-Marie Flaus et Monsieur Stéphane Ploix, aussi bien pour leur aide précieuse que pour leurs qualités humaines. Malgré les nombreuses sollicitations, ils m'ont toujours consacré le temps nécessaire pour être guidé efficacement. Au-delà de leur grande rigueur scientifique et de l'aide technique qu'ils m'ont apportées, leur soutien amical et la reconnaissance dont ils ont fait preuve envers mon travail m'ont aidés à franchir bien des barrières.

Monsieur Stéphane Ploix, Maître de conférence à l'Institut Nationale Polytechnique de Grenoble. Durant ces trois années, il a su me donner ce regard scientifique qui permet de mieux comprendre les choses aussi bien dans la vie professionnelle que privée.

Je tiens à remercier les membres du jury de me faire l'honneur de juger cette thèse :

Madame Sylviane Gentil, Professeur à l'Institut Nationale Polytechnique de Grenoble, LAG, qui m'a fait l'honneur de présider le jury de cette thèse.

Madame Louise Travé-Massuyès, Directeur de recherche CNRS rattachée au laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) ainsi que Monsieur Jean-Philippe Cassar, Professeur à l'Université de Lille 1 rattaché au laboratoire d'Automatique et d'Informatique Industrielle de Lille (LAIL) d'avoir accepté de prendre ce travail en considération en tant que rapporteurs de ce jury.

Madame Sylvie Pesty, Professeur à l'Institut Nationale Polytechnique de Grenoble, rattachée au laboratoire Leibniz de ENSIMAG et Monsieur Philippe Dague, Professeur à l'université Paris 13 rattaché au Laboratoire Informatique de Paris Nord (LIPN), d'avoir accepté de prendre ce travail en considération en tant qu'examinateurs de ce jury.

Je souhaiterais exprimer mes sincères remerciements à tous les membres du Laboratoire d'Automatique de Grenoble pour leur aide ou simplement pour leur sympathie, ainsi qu'au groupe de travail du projet « MAGIC » qui tous deux m'ont permis de mener à bien cette étude.

Je voudrais également remercier tous les enseignants-chercheurs avec qui j'ai collaborés au cours de mes enseignements à ENSIEG, IUT, Polytech'Grenoble, pour la confiance qu'ils m'ont accordée et l'expérience qu'ils m'ont fait découvrir.

Enfin, je tiens à remercier ma famille pour leur soutien inconditionnel. Je ne peux oublier Karima Mimouni pour sa patience, son soutien et son dévouement qui fut particulièrement indispensable.

Table des matières

Introduction générale

Depuis les débuts de l'ère industrielle, l'homme s'est vu confronté aux problèmes des machines qu'il créait. Il a même souvent été victime de problèmes survenus à cause d'erreurs de conception, de perturbations externes diverses ou encore, à cause d'un manque d'entretien des mécanismes. Il s'est donc très rapidement soucié de l'état de ses machines et a tenté de minimiser les conséquences des problèmes techniques.

Les systèmes industriels sont également devenus de plus en plus complexes avec l'automatisation de boucles de contrôle, l'introduction des microprocesseurs à différents niveaux et, plus récemment, l'informatisation hiérarchisée et distribuée. De fait cette évolution a complexifié la tâche de diagnostic des installations industrielles. Les procédures de diagnostic de défauts dans les systèmes physiques s'avèrent devenir très complexes dès que les systèmes considérés ne sont plus élémentaires. Il est alors légitime pour les entreprises d'acquérir un système efficace de surveillance afin d'améliorer la sécurité des personnels et d'assurer une fiabilité et une disponibilité accrues de leur outil de production. En effet, vu l'effort financier consenti à l'acquisition d'une chaîne de fabrication performante, un dysfonctionnement réduisant momentanément la production est préjudiciable.

Pour que les méthodes de diagnostic puissent s'appliquer à des systèmes complexes, il est crucial de concevoir des méthodologies et des environnements adaptés. Pour implémenter ces méthodologies, des systèmes de diagnostic doivent être conçus pour supporter de nombreuses procédures se déclenchant les unes les autres et fonctionnant souvent en parallèle. De nombreux tests de détection (appeler aussi tests de cohérence ou tests de consistance), orientés bon ou mauvais fonctionnement, reposant sur différents modèles, doivent être réalisés. Certains de ces tests peuvent être effectués en permanence, d'autres déclenchés par d'autres tests. A un niveau supérieur, les résultats des tests de détection doivent être interprétés pour conduire à un diagnostic tenant compte du passé du système surveillé.

▪ OBJECTIFS DE LA THESE

L'objectif de notre travail est de concevoir un système de diagnostic fiable permettant d'appréhender les systèmes dynamiques complexes spatialement distribués. En effet, un système supervisé peut être géographiquement distribué et de grande taille. Par exemple, un réseaux de télécommunication [Poncolé, 2002]. Pour cette raison nous cherchons à concevoir une approche de diagnostic qui supporte la distribution. Nous nous appuierons sur le paradigme multi-agent et sur les techniques de diagnostic dites à base de modèle. Nous cherchons aussi par le biais de ce travail de mettre en place des algorithmes pour le diagnostic des systèmes complexes et dynamiques en étant le plus efficace possible. La complexité du

problème vient en particulier du fait que les informations à traiter sont nombreuses (taille du système, nombre d'alarmes reçues, les algorithmes sur lesquels se basent les tests de détections peuvent être hétérogènes,...) et que l'on cherche à établir une réponse exhaustive : quelles sont les explications possibles du comportement observé ? Quels dysfonctionnements le système subit-il ou a-t-il subi ?

Notre objectif consiste à proposer une méthode de diagnostic qui tire partie des deux approches DX (communauté d'Intelligence Artificielle) et FDI (Fault detection and Isolation) (communauté de l'Automatique) en distinguant la phase de génération de symptômes, qui peut se faire à base de techniques variées et parfois très sophistiquées et hétérogènes de la phase de raisonnement diagnostique, qui doit permettre de garantir ce qui peut l'être et d'analyser toutes les informations disponibles pour en déduire le diagnostic le plus juste et complet possible. La phase de génération de symptôme correspondra à la phase de la détection. En revanche, la phase de raisonnement diagnostique correspondra à la phase d'analyse diagnostique.

La génération de symptôme peut être réalisée en utilisant des algorithmes différents (détection à base de signal, détection utilisant des graphes causaux, observateurs d'état, relations de parité, détection à base des réseaux de neurones,...). De plus, ces algorithmes peuvent être physiquement distribués. L'algorithme d'analyse diagnostique doit donc être capable d'appréhender des tests de détection hétérogènes et distribués qui doivent être décrit de la même façon pour pouvoir être exploités par un algorithme d'analyse diagnostique.

Différents tests de détection peuvent vérifier des états de composants communs. Deux tests différents peuvent s'intéresser à un même sous système physique. L'analyse diagnostique doit alors être capable d'appréhender ce type de situation. De plus, les tests de détection peuvent conduirent à des fausses alarmes. L'analyse doit donc être en mesure de reconnaître les tests de détection ayant produit des symptômes erronés.

Les résidus sont affectés par les imprécisions des mesures et par les erreurs de modèle. Il en résulte des incertitudes sur la prise de décision. Le concept même de fonctionnement normal est vague et il y aura toujours une région d'ambiguïté où les incertitudes doivent être maîtrisées pour rendre la décision finale fiable. Les tests de détection peuvent fournire des décisions comprises entre 0 et 1 où 1 représente une inconsistance sûre et 0 une consistance sûre. L'analyse diagnostique doit pouvoir exploiter ces résultats.

L'un des enjeux scientifique de notre travail est de proposer une formalisation pour la modélisation des systèmes dynamiques complexes, qui permette de mettre en place des analyses diagnostiques logique telle que celles utilisées par la communauté DX. Ce formalisme propose une description complète d'un système physique complexe et dynamique, impliquant différents composants, qui peut intégrer différents modèles avec ou sans équations différentielles. Orienter la description composant pour le diagnostic rend la compréhension rapide du résultat diagnostic par l'opérateur. Par conséquent, elle lui permet aussi une maintenabilité plus aisée du système à diagnostiquer. Le formalisme proposé permet de tenir compte de la validité du modèle utilisé dans l'analyse diagnostic. La prise en compte de cette

validité lors de la détection a été mise en application dans le cas du projet MAGIC [Garcia-Beltran, 2004]. Le formalisme doit aussi permettre d'appréhender en même temps le bon et le mauvais fonctionnement des différents composants du système afin de pouvoir l'exploiter par l'analyse diagnostique.

Divers réseaux internationaux de recherche dans ce domaine ont été créés pour faire face aux problèmes associés à ces niveaux ; en particulier, il est possible de trouver des exemples de ces groupes internationaux dédiés aux problèmes du diagnostic et de la supervision comme MONET-BRIDGE, A1, COSY, OSA2. De nombreux outils informatiques ont été créés comme résultats de ces secteurs de la recherche. On peut mentionner quelques exemples orientés vers les tâches de diagnostic et de supervision en général : MIMIC [Dvorak et Kuipers, 1991], MIDAS [Oyeleye, Finch et al, 1990], DIAPASON et SALOMON [Pénalva, Coudouneau et al, 1993][Montmain, 1997], TIGER [Travé-Massuyes et Milne, 1996], DIAMON [Lackinger et Nejdl, 1993], et plus récemment les boîtes à outils de CHEM [CHEM-Consortium, 2000].

De la même manière, le travail présenté dans ce mémoire se trouve dans le cadre du projet MAGIC (*Multi-Agents-Based Diagnostic Data Acquisition and Management in Complex systems*) de recherche et développement dans le domaine de la surveillance et la supervision. Dans ce projet européen interviennent cinq participants : l'université de Duisburg, l'université de Karlsruhe, le Laboratoire d'Automatique de Grenoble ; et deux entreprises privées : SATE (*System Advanced Technologies Engineering*) et SMS-DEMAG, la première orientée vers le développement de systèmes informatiques pour la modélisation et le diagnostic de systèmes et la deuxième orientée vers le développement d'équipement et de système de contrôle commande pour l'industrie métallurgique.

Dans le cadre des systèmes industriels tels que ceux que nous avons abordés dans le projet MAGIC, pour des raisons liées à des capacités de calcul, tous les tests de détection ne peuvent pas être actifs en permanence. L'objectif consiste à déterminer quels sont les tests de détection les plus intéressants à déclencher.

▪ ORGANISATION DU DOCUMENT

PARTIE I : Conception d'algorithmes modulaires pour le diagnostic

Le **premier chapitre** rappelle dans un premier temps la terminologie adoptée des différents termes liés au diagnostic afin de se positionner par rapport à ce qu'on trouve dans la littérature scientifique. Ensuite nous avons posé et positionné le problème et défini le cadre de notre travail en illustrant les différents objectifs propres au diagnostic des systèmes, ainsi que sa place au sein d'une stratégie de surveillance et de supervision. Quelques notions préliminaires ont été étudiées car il fallait définir précisément ce qu'est un test de détection ainsi que la notion de non-exonération. Ensuite, une formulation du problème, étudié tout au

1 Abnormal Situation Management Consortium: http://www.asmconsortium.com

2 OSA-CBM : Open System Architecture for Condition Based Maintenance www.osacbm.org

long de ce mémoire, permet de mieux comprendre la procédure de diagnostic et dans quoi elle s'intègre. Les principales approches d'une procédure de détection seront explicitées dans la section 6. Celle qui s'applique le mieux à notre problématique repose sur l'approche à base de modèle et sera plus largement détaillée. Enfin, la spécificité d'un contexte distribué pour le diagnostic sera alors mise en évidence dans la section 7 et 8.

Le **deuxième chapitre** illustre les limites des approches de localisation structurelle à base de tables de signatures utilisées largement par la communauté FDI (*Fault Detection and Isolation*) issues de la communauté automatique, décrite dans [Frank, 1990][Patton et al., 1989] par rapport à l'approche dite logique ou à base de consistance de la communauté DX (Diagnoses) issue de l'intelligence artificielle, décrite dans [De Kleer and Williams, 1987] [Reiter, 1987] [Dague, 2001]. La première approche traite principalement des symptômes générés à partir des modèles de composants représentant le mauvais fonctionnement. La seconde approche, originairement dédiée aux systèmes binaires statiques, propose un raisonnement formel basé sur des modèles de bon fonctionnement des composants d'un système physique. Cette approche ne distingue pas la génération de symptômes du raisonnement diagnostique ou en d'autres termes la détection de la localisation de défaut. Nous présentons les bases théoriques qui seront utilisées dans les chapitres suivants concernant le diagnostic à base de modèles avant de présenter une étude comparative des deux approches.

Le **troisième chapitre** présente notre contribution à la modélisation pour le diagnostic des systèmes complexes et dynamiques. En effet, ce chapitre montre que les approches DX et FDI sont complémentaires. Nous proposons alors une nouvelle approche bénéficiant des atouts de chacune d'elle. Nous proposons un cadre théorique permettant de formuler rigoureusement la description du système dynamique à diagnostiquer. Ce formalisme permet la construction des supports de test de détection qui intègrent la validité des modèles ainsi que les hypothèses sur les composants. Les tests de détection doivent être décrits de la même façon pour pouvoir être exploités par un algorithme d'analyse diagnostique. C'est le rôle du sous-système testable. L'analyse diagnostique doit s'appuyer non pas sur les algorithmes de détection eux-mêmes, mais sur les sous-testables qui constituent une méta-description d'un test de détection. Enfin, la dernière section s'intéresse aux résultats des tests de détection et propose une méthode qui permet d'appréhender les incertitudes dans la décision en transposant la définition d'un test de détection avec la logique des prédicats en logique floue. Ainsi, la méthode permet une analyse diagnostique plus circonstanciée.

Le **quatrième chapitre** propose une méthode d'analyse diagnostique inspirée de l'algorithme de Reiter initialement dédiée aux systèmes statiques qui soit capable d'analyser formellement les symptômes issus de tests de détection dynamiques et hétérogènes. Il s'agit ici de développer les détails de notre approche qui tire profit des approches développées par la communauté Automatique et la communauté de l'Intelligence Artificielle. En effet, la première communauté propose des méthodes de détection très sophistiquées dédiées aux systèmes dynamiques. Par ailleurs, la deuxième communauté fournit un algorithme formel d'analyse diagnostique. L'analyse diagnostique doit pouvoir exploiter le fait que les tests de

détection mise en oeuvre dans le cadre du projet MAGIC fournissent des décisions comprises entre 0 et 1 où 1 représente une alarme sûre et 0 pas d'alarme sûre. Notre approche permet d'appréhender l'aspect distribué des systèmes à diagnostiquer. De plus, c'est une approche qui permet de traiter simultanément le bon et mauvais fonctionnement des composants du système. Enfin, pour des raisons liées à des capacités de calcul, tous les tests de détection ne peuvent pas être actifs en permanence. L'objectif consiste à déterminer quels sont les tests de détection les plus intéressants à déclencher qui est le rôle de la stratégie de diagnostic. L'approche intègre des indicateurs de performance qui permettent de choisir la plus pertinente des stratégies de diagnostic en terme de temps de calcul et de précision des résultats.

PARTIE II : Solution technologique pour le diagnostic distribué

Le **cinquième chapitre** présente une solution technologique pour concevoir des systèmes de diagnostic distribués. Cette solution se base sur le paradigme multi-agent. Les systèmes multi-agents permettent d'appréhender la complexité de la résolution d'un problème donné. Notre objectif consiste à exploiter les systèmes multi-agents dans le cadre du projet MAGIC afin de concevoir une procédure de diagnostic distribué. Pour ce faire, une bonne connaissance du paradigme multi-agent est nécessaire. Ce chapitre définit les agents logiciels, les systèmes multi-agents, et montre l'importance du sujet. Il va aussi rapidement situer les concepts introduits vis-à-vis d'autres disciplines et présenter les domaines importants de recherche liés aux agents et systèmes multi-agents.

Le **sixième chapitre** propose et décrit une architecture à base de système multi-agent produit d'un projet de collaboration européenne intitulé MAGIC. Le système MAGIC intègre diverses méthodes de diagnostic, adaptées à différentes situations, et tous les outils nécessaires pour les adapter aux besoins du client, les configurer et les employer pour la surveillance en temps réel. Cette architecture est fondée sur le concept de système multi-agent pour le diagnostic distribué. Cette architecture est multicouches (Multi-Agents Multi-Layer (MAML)). MAGIC est un système d'aide à la décision pour la supervision de procédés complexes. Les fonctionnalités requises pour l'aide à la décision ont été distribuées tout à la fois à différents niveaux et dans différents agents. La construction modulaire fondée sur les agents a permis le développement d'un système logiciel robuste. Cette architecture a permis aussi la collaboration des diverses techniques de détection de défauts et d'aide à l'opérateur. Ensuite, l'agent de décision diagnostique (nommé *Diagnostic Decision Agent* (DDA)) qui intègre notre méthode d'analyse diagnostique est présenté. Cet agent est le cœur de notre travail. Il analyse les symptômes fournis par les tests encapsulés dans les agents de détection. Il intègre dans son analyse diagnostique la stratégie de déclenchement des tests de bon ou mauvais fonctionnement. L'agent de décision diagnostique propose à l'opérateur un ensemble exhaustif de diagnostics classés par ordre de plausibilité et permet de retrouver les tests qui ont conduit à des fausses alarmes. Enfin, on illustre le fonctionnement du système MAGIC sur une application réelle qui consiste en un « enrouleur hydraulique ».

Ce mémoire s'achève par une conclusion générale où quelques perspectives des futures recherches sont évoquées.

PARTIE I

Conception d’algorithmes modulaires pour le diagnostic

Chapitre 1

Formulation du problème de diagnostic

I.1. INTRODUCTION

La surveillance d'un système physique a besoin des informations délivrées par des capteurs. La défaillance de l'un d'entre eux peut engendrer le dysfonctionnement de l'installation en conduisant à une altération de ses performances. Une procédure de surveillance doit donc être apte à traiter un défaut affectant aussi bien le système physique proprement dit, que l'un de ses organes de conduite (actionneur, capteur d'instrumentation). Cependant, la complexité et la taille de l'installation, ainsi que la gravité des risques encourus et des conséquences potentielles augmentent la quantité d'information à analyser, rendant souvent la surveillance complexe pour un opérateur humain. Il s'avère par conséquent très utile d'adjoindre à l'opérateur une aide à la décision, voire de rendre la surveillance automatique. L'intérêt est aussi de permettre l'amélioration de la disponibilité des installations en remplaçant les politiques de maintenance programmée par des stratégies de maintenance conditionnelle prenant en compte l'état effectif du système physique.

Le diagnostic des systèmes suscite, depuis une trentaine d'années, un intérêt croissant tant au niveau du monde industriel que de la recherche scientifique. A l'origine, le diagnostic se limitait aux applications industrielles à haut niveau de risque pour la communauté comme le nucléaire ou l'aéronautique [Potter et al, 1977] [Daly et al, 1979] [Desai et al, 1979], ainsi qu'aux secteurs d'activité de pointe tels que l'industrie de l'armement ou l'aérospatial [Desai et al, 1976], [Deckert et al, 1977]. Les premiers travaux concernant le thème diagnostic datent du début des années 1970, résumés notamment dans l'article de synthèse [Willsky, 1976]. En raison de l'intérêt croissant suscité dans le monde industriel, le diagnostic est devenu peu à peu un thème de recherche à part entière.

Ce chapitre introductif, vise à rappeler dans un premier temps la terminologie utilisée dans la littérature scientifique et celle que nous avons adoptée dans ce mémoire. Les principales approches de détection seront énoncées. Les notions de test de détection et le principe d'exonération seront développés, conduisant au principe fondamental sur lequel repose toute procédure de diagnostic utilisant des modèles. La problématique inhérente à ce type d'approche sera alors mise en évidence. Par ailleurs, l'intérêt d'avoir une architecture distribuée par opposition à une architecture centralisée est mis en évidence.

I.2. TERMINOLOGIE PROPRE AU DIAGNOSTIC

Il semble intéressant, dans un premier temps, de rappeler les principaux termes utilisés en diagnostic des systèmes. Reposant principalement sur le travail effectué par [Milne, 1987], [Isermann et Balle 1997, 2000] et [Ploix, 1998], ainsi que sur l'étude d'ouvrages synthétiques tels que [Brunet et al, 1990] [Zwingelstein, 1995] des définitions des concepts coïncidant avec notre acception sont proposées afin de clarifier la suite de ce document. Ce travail nous paraît nécessaire car il est courant de trouver dans la littérature internationale des définitions différentes d'une même notion, notamment lorsque ces notions sont employées par deux communautés. Par exemple la terminologie suggérée par [Isermann et Balle, 1997] n'est pas toujours compatible avec les définitions mathématiques de Reiter. La terminologie suivante sera adoptée :

a) Système physique (*Physical System*)

Un système physique est un ensemble d'éléments (composants, constituants) interconnectés ou en interaction organisés pour réaliser une fonction.

b) Composant (*Component*)

Un composant est une partie du système choisie selon des critères liés à la modélisation. En tout premier lieu, le comportement de référence de ce composant est bien adapté dans le sens où il peut être défaillant ou servir de support à la propagation de pannes dans le système. Un composant doit être simple à modéliser dans le sens où cela doit être naturel : il peut s'agir d'un composant (physique ou logique) complet du système ou d'une partie parfaitement délimitée de ce composant, d'un groupe de composants. Le comportement du composant élémentaire n'est pas décomposable ou alors cette décomposition n'est pas souhaitée, il constitue une « brique » du comportement du système.

c) Modèle (*Model*)

Un modèle d'un système physique est une description de sa structure et une représentation comportementale ou fonctionnelle de chacun de ses composants [Milne, 1987]. Une représentation comportementale est constituée de relations entre diverses variables du système, appelées classiquement relations de causes à effets. Une représentation fonctionnelle est plus abstraite puisqu'elle ne s'adresse qu'aux objectifs présumés que le système physique doit remplir. Le niveau structurel, quant à lui, s'appuie sur la structure réelle du système physique et décrit les interconnections entre ses différents éléments ou constituants. Les niveaux comportemental et fonctionnel comprennent des relations entre des grandeurs physiques (variables) et permettent de mettre en évidence la présence d'un événement anormal ou anomalie. Le niveau structurel, quant à lui, permet de déterminer l'élément affecté par le défaut. L'intérêt de cette décomposition est de rappeler que, puisqu'un modèle contient toute l'information relative à un système physique, il est utilisable ensuite par la procédure de diagnostic.

d) Défaut (*Fault*)

- Tout écart entre la caractéristique observée sur le dispositif et la caractéristique de référence, lorsque celui-ci est en dehors des spécifications [AFNOR, 1994].
- N'importe quel état indésirable d'un composant ou d'un système. Un défaut n'implique pas nécessairement une défaillance [IEEE, 1988].
- Déviation non permise d'au moins une propriété ou un paramètre caractéristique du système des conditions acceptables ou (et) standards [Isermann et Ballé, 1997].
- Un défaut est une anomalie de comportement au sein d'un système physique localisée au niveau d'un composant [Ploix, 1998].

La définition de l'AFNOR rattache la notion de défaut à celle de déviance d'une caractéristique d'un phénomène, subordonnant ainsi cette notion à l'existence d'une référence absolue. Cependant, dans le cas général, il y a plusieurs modèles de référence, s'exprimant en terme de relations entre des phénomènes, pouvant représenter un même phénomène. La notion de défaut

ne serait donc plus à rattacher à celle de phénomène, mais à celle de modèle ce qui n'est pas conforme aux autres définitions. Celles données par le dictionnaire IEEE et par Isermann et al rapprochent bien défaut de comportement ; cependant, les qualificatifs indésirables ou non permis appartiennent au jugement ; en conséquence, nous avons préféré utiliser anomalie qui procède d'une détermination arbitraire. La notion de défaut est donc voisine de celle de défaillance mais comme le souligne le dictionnaire IEEE, un défaut n'implique pas nécessairement une défaillance. Défaut, lié au comportement, est plus général que défaillance, liée aux fonctions puisqu'elles sont une abstraction du comportement tel qu'il est conçu téléologiquement. La description comportementale est plus détaillée que la description fonctionnelle et l'inclut donc. De la même manière, la notion de défaut inclut celle de défaillance ; un défaut n'altère pas nécessairement le fonctionnement d'un système physique mais peut présager d'une défaillance à venir.

e) Défaillance (*Failure*)

Une défaillance définit une anomalie fonctionnelle au sein d'un système physique [Ploix, 1998], c'est-à-dire caractérise son incapacité à accomplir certaines fonctions qui lui sont assignées.

Les défauts incluent les défaillances mais la réciproque n'est pas vraie. Un système peut remplir sa fonction tout en présentant une anomalie de comportement. Par exemple, une machine électrotechnique peut produire un bruit anormal tout en entraînant correctement une charge, en supposant que telle soit sa fonction. Le bruit anormal est un défaut qui peut permettre de présager d'une défaillance à venir. La recherche de défauts est donc fondamentale en diagnostic.

f) Panne (*Break-down*)

La panne est l'inaptitude d'un dispositif à accomplir la fonction vitale. Il est clair que dès l'apparition d'une défaillance, caractérisée par la cessation du dispositif à accomplir sa fonction, on déclarera le dispositif en panne. Par conséquent, une panne résulte toujours d'une défaillance. [Zwingelstein, 1995].

g) Symptôme (*Symptom*)

Caractère distinctif d'un état fonctionnel anormal [Ploix, 1998].

h) Résidu (*Residual*)

Souvent, lorsque le modèle comportemental de référence est analytique, les signaux porteurs de signes ou de symptômes sont appelés résidus parce qu'ils résultent d'une comparaison entre un comportement réel et un comportement de référence.

i) Diagnostic (*Diagnosis*)

Un diagnostic est un état expliqué d'un système physique compatible avec les informations disponibles sur le comportement réel du système et avec le modèle de comportement de référence disponible. Habituellement, le diagnostic est exprimé par les états des composants [Reiter, 1987] ou les états des relations de description du comportement [Cassar et al, 1994].

j) Perturbation :

Entrée du système physique qui n'est pas une commande. Autrement dit, c'est une entrée non contrôlée.

La figure 1.1 représente les anomalies suivant leur criticité. Il existe également une criticité croissante entre défaillance et panne. De la non conformité (ou anomalie) dans le cas d'une défaillance, on passe à une inaptitude à accomplir une fonction dans le cas d'une panne.

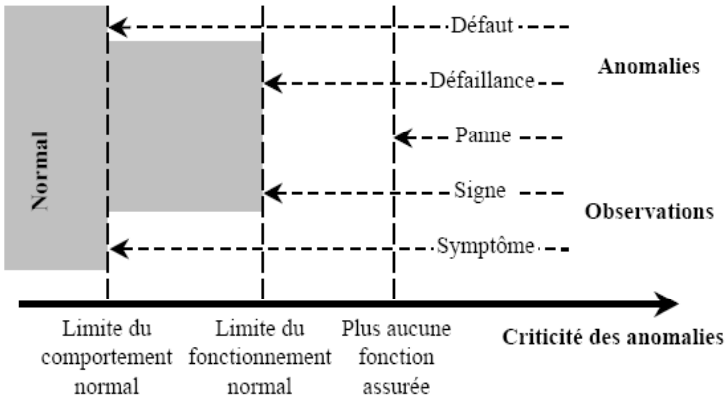


Figure 1.1-Anomalies et Observations classées par criticité croissante d'après [Adrot, 2001]

Ces notions sont illustrées à partir de l'exemple d'un moteur devant assurer une fonction de ventilation (tableau 1.1).

Définition illustrée	**Evénement (exemple)**	**Ecart au comportement nominal (courant, vitesse)**	**Aptitude à remplir la fonction de ventilation**
Perturbation	Variation de température extérieure (normal)	Petit	Totale
Défaut	Fort échauffement (anormal)	Moyen	Totale
Défaillance	Déclenchement intermittent d'un relais thermique stoppant le ventilateur jusqu'à ce que la température du moteur redescende à un niveau acceptable.	Grand	Partielle
Panne	Suite aux forts échauffements répétitifs, les isolants sont progressivement endommagés: un court-circuit apparaît; le moteur ne peut plus tourner jusqu'à ce qu'une réparation soit effectuée.	Grand	Nulle

Tableau 1.1-Illustration des définitions à l'aide d'un moteur de ventilateur

Effet des perturbations :

Contrairement à ce que pourrait laisser penser le tableau 1.1, les écarts de comportement relatifs à des perturbations ne sont pas nécessairement plus faibles que ceux associés aux défauts. Par exemple, une perturbation telle qu'une variation du couple de charge sur un moteur peut entraîner des écarts très importants par rapport à un comportement de référence, sans que cette situation soit critique. Des écarts de comportement beaucoup plus faibles, résultant par exemple de courts-circuits entre quelques spires d'une même phase, correspondent par contre à une situation plus critique pour laquelle un diagnostic pourra être envisagé.

I.3. POSITION DU PROBLEME

Le diagnostic s'intègre dans le cadre plus général de la surveillance et de la supervision. Le diagnostic à base de modèles trouve sa place dans les démarches "qualité" des entreprises et s'inscrit dans la sûreté de fonctionnement. La fonction d'une opération de diagnostic est de déterminer les composants ou organes défaillants d'un système physique. Elle peut intervenir à plusieurs stades :

- Les contrôles "qualité". Il s'agit de tester des produits afin de garantir que leur caractéristiques sont conformes à des spécifications.
- La supervision. Il s'agit de doter les systèmes physiques d'une intelligence en les équipant de dispositifs étudiant en temps réel leur comportement pour produire automatiquement un diagnostic qui sera fourni et exploité par l'opérateur.
- La maintenance prédictive. Il s'agit de déceler des dérives de comportements d'un système physique avant qu'une fonction ne soit altérée afin de remplacer les organes dégradés avant qu'ils ne tombent en panne.
- L'aide au diagnostic. Il s'agit d'aider un opérateur à remonter aux organes défectueux.

Nous nous intéressons dans notre étude à une connaissance sous forme de modèles structurels, connaissance en partie issue des lois de la physique. Les modèles représentent soit le bon ou mauvais fonctionnement du système à diagnostiquer.

Nous nous sommes restreints au diagnostic tel qu'il a été défini précédemment en excluant ce qui avait trait au pronostic, à la reconfiguration, à la sûreté de fonctionnement, à la validation de données ou même à l'étiologie.

La *surveillance* de la majeure partie des procédés industriels se limite à des systèmes de traitement d'alarmes. Des valeurs limites sont définies sur des variables clés par des experts du procédé selon des critères de sécurité concernant les hommes, l'installation et son environnement. Les mesures au-delà de ces valeurs limites déclenchent des alarmes. Un système de traitement d'alarmes est donc l'outil de base pour aider l'opérateur dans sa tâche de surveillance. Il reste cependant aux opérateurs à analyser la situation et à prendre une décision adaptée : actions correctives, conduite en mode dégradé, activation d'une procédure d'arrêt d'urgence. L'efficacité de l'opérateur est primordiale que ce soit du point de vue économie ou sécurité. De nombreux facteurs influencent sa performance à répondre à une alarme : le nombre et la fréquence des alarmes, la présentation et la complexité de l'information, son expérience et son entraînement, sa vigilance et sa réaction au stress.

La *supervision* a pour objectif de surveiller et de contrôler le fonctionnement d'une installation pour qu'elle reste dans la plage de fonctionnement normale quelles que soient les perturbations extérieures. Elle est essentiellement effectuée par les opérateurs dans la salle de contrôle [Bullemer et Nimmo 1996]. La supervision comprend la surveillance, la prise de décision au niveau de la génération des symptômes ainsi que la prise de décision au niveau de l'analyse diagnostique pour générer des diagnostics et la mise en œuvre des actions appropriées pour maintenir les fonctions d'un système (en mode normal ou en mode dégradé) et éviter des dégâts

matériels ou humains. Un autre aspect de la supervision [Travé-Massuyès et al., 1997] est que son objectif, le plus souvent, n'est pas de remédier à un véritable défaut, mais d'assurer un fonctionnement optimal d'une installation, en terme par exemple de respect de normes de qualité du produit obtenu, de rentabilité de la production, etc.

La conception d'une procédure de diagnostic se décompose en trois tâches principales.

Il faut tout d'abord extraire des symptômes du système physique à diagnostiquer : on parle de génération de symptômes ou de détection. Il s'agit de vérifier, grâce à des tests, la consistance entre des informations sur le comportement réel d'un système physique tel qu'il peut être observé par l'intermédiaire de capteurs par exemple et son comportement attendu tel qu'il peut être prédit grâce aux modèles de bon ou mauvais comportement. Toute contradiction entre les observations et les prédictions déduites des modèles est nécessairement la manifestation d'un dysfonctionnement, c'est-à-dire de la présence d'un ou plusieurs défauts. Par exemple, si une prédiction a été faite en utilisant les modèles de composants C_1, …,C_n et qu'elle entre en contradiction avec une observation, c'est donc que les composants C_1, …,C_n ne peuvent être tous corrects et que l'un d'eux est nécessairement défectueux, on parle alors de la génération de symptômes. Reiter et de Kleer parlent dans leurs travaux de génération de conflits dans [Reiter, 87][De Kleer et Williams, 1989]. La génération de symptômes grâce à des observations constitue donc la première phase du diagnostic.

La tâche de raisonnement diagnostique consiste à analyser les symptômes disponibles fournis par les tests de détection pour déterminer les états plausibles d'un système physique. Autrement dit, en recoupant ces conflits, on va progressivement affiner la localisation du (ou des) défaut(s), ce qui constitue la deuxième phase du diagnostic.

La stratégie de diagnostic organise les tâches de génération de symptômes et de raisonnement diagnostique pour être conforme à des exigences de précision des résultats et de rapidité de la procédure de diagnostic, ce qui constitue la troisième phase d'une procédure de diagnostique.

I.4. QUELQUES NOTIONS PRELIMINAIRES

La section précédente explique les concepts et principes du diagnostic à base de modèles analytiques des systèmes physiques qui ressortent de la littérature scientifique. Cependant, pour que la synthèse soit complète, il manque d'expliquer quelques principes généraux.

I.4.1. Les tests de détections

Un test de détection (dit aussi test de cohérence ou test de consistance) a pour finalité de vérifier si un ensemble d'informations représentatives de l'état d'un système physique est cohérent avec la connaissance d'un comportement donné qui peut être normal ou anormal (voir figure 1.2). Le résultat de la comparaison produit un écart, appelé résidu. Cet écart sera comparé à des seuils fixés à priori. Si le seuil de la détection est trop petit, il peut y avoir des fausses alarmes. Si le seuil est trop grand, on aboutit à des manques à la détection. Les informations sont associées à des

variables ; elles peuvent être des observations qualitatives ou des mesures[3]. Les connaissances utilisées peuvent être formalisées de nombreuses manières différentes : la nature d'un test de détection dépend fondamentalement du choix de formalisation. Le tableau suivant présente quelques formalisations et les techniques associées.

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 algébro-déterministes	Observateurs d'état, relations de parité,…
modèles algébro-stochastiques	Filtre de Kalman
modèles algébro-ensemblistes	relations de parité ensemblistes,…

Table 1.1- Quelques formalisations et les techniques associées aux tests

Avant de réaliser l'analyse diagnostique, de nombreux tests de détection qui permettent de générer les symptômes doivent être réalisés. Ces tests sont orientés soit bon ou mauvais fonctionnement, reposant sur différents modèles. Un test de détection repose sur un modèle de comportement ne faisant intervenir nécessairement que des variables et paramètres connus dits mesurés : ce modèle est un agrégat de relations comportementales élémentaires (qui peuvent faire intervenir des phénomènes non connus ou non mesurés). Ce dernier est nommé relation de redondance analytique (RRA).

Le modèle de comportement peut être une référence de comportement normal (nommé Bon Fonctionnement) ou un modèle de comportement du système étudié affecté d'un ou de plusieurs défauts donnés (Mauvais Fonctionnement). Toutes les natures de tests de détection ne conviennent pas à des tests orientés mauvais fonctionnement car, par essence, ces tests comportent généralement des indéterminismes représentatifs de l'ampleur d'un défaut. Lorsque c'est possible, ces méthodes sont souvent complétées par des analyses temporelles des signaux, des méthodes de détection de saut de moyenne par exemple. D'une manière générale, les procédures de diagnostic orientées bon fonctionnement conduisent à des diagnostics d'autant plus précis que le nombre de phénomènes connus (mesurés) est important ; en revanche, les procédures orientées mauvais fonctionnement sont d'autant plus précises que le nombre de modèles de mauvais fonctionnement est important.

[3] Qui peuvent être à valeur dans l'ensemble des réels R ou dans un ensemble fini de valeurs.

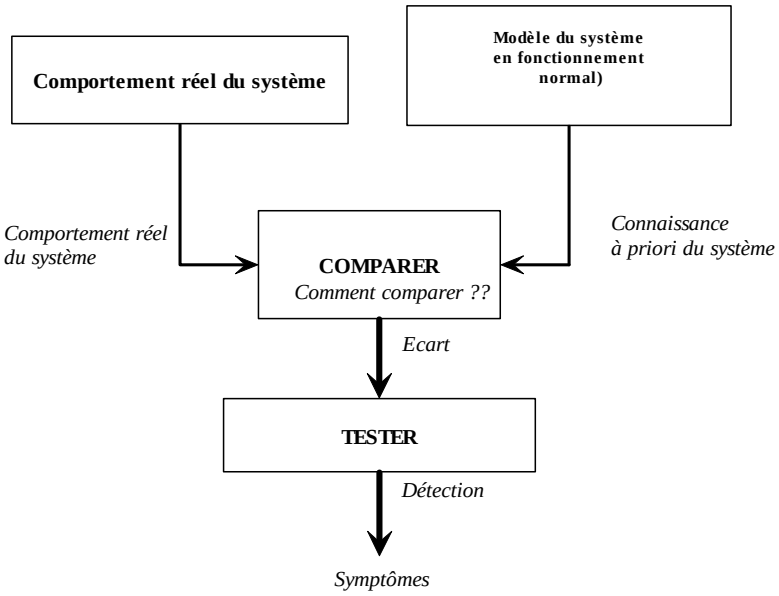


Figure 1-2- Test de cohérence (test de détection, test de consistance)

I.4.2. Principe d'exonération et de non-exonération

Un modèle analytique de bon fonctionnement n'est rien moins qu'une ou plusieurs relations analytiques caractérisant le comportement normal. Or, il est possible qu'en certains points de fonctionnement, un comportement défaillant s'apparente à un comportement normal. Ce n'est donc pas parce qu'un comportement s'apparente au comportement normal qu'il est possible de conclure à l'absence de défauts. En conséquence, seule l'incohérence entre un modèle bon fonctionnement et des observations permet de conclure avec certitude à la présence de défauts. L'hypothèse d'exonération suppose que si les algorithmes de détection orientés bon fonctionnement ne produisent pas d'alarmes alors il n'y a aucun défaut dans le sous-système testé (cette hypothèse peut mener, dans la pratique, à de faux diagnostics car un défaut n'est généralement pas détecté tout le temps). En effet, la cohérence d'un test de détection veut dire que le comportement réel est le même que le comportement modélisé au point de fonctionnement considéré. Par conséquent, la cohérence d'un test de détection ne traduit pas nécessairement une absence de défaut. Pour obtenir l'équivalence, il faudrait que la cohérence soit constatée en tout point de fonctionnement. Une incohérence a un sens exclusif que la cohérence n'a pas.

- Lorsque le test de détection est incohérent ⇨ Au moins un composant étudié par le test est dans un état anormal.
- Lorsque le test de détection est cohérent ⇨ Tous les composants étudiés par le test sont dans un état normal au point de fonctionnement.

I.5. FORMULATION DU PROBLEME

Avec le développement de l'automatisation, les progrès techniques ont permis le développement de systèmes physiques de plus en plus complexes. Le rôle de l'homme face à une installation technique a donc évolué, pour passer d'une manipulation physique directe à une tâche de contrôle, voire de supervision. Lorsqu'une défaillance survient, l'opérateur du système est

généralement face à une cascade d'informations qui met à rude épreuve ses capacités intellectuelles, sa réactivité et sa gestion du stress. Dans ces conditions, le diagnostic devient une tâche qui est plus ou moins facile à gérer, allant parfois jusqu'à augmenter les risques d'erreurs humaines. De plus, ces tâches ne sont pas aisées à réaliser car la prise de décision doit être accomplie en temps réel.

Le problème qui se pose est d'essayer de concevoir des systèmes de diagnostic qui doivent être capables d'analyser formellement des systèmes dynamiques complexes modélisés avec des équations différentielles. Comment distinguer la phase de génération de symptômes, qui peut se faire à base de techniques variées et parfois très sophistiquées (observateur d'état, relation de parité…) : il s'agit d'une opération de décision, de la phase de raisonnement diagnostique, qui doit permettre de garantir ce qui peut l'être et d'analyser toutes les informations disponibles pour en déduire le diagnostic le plus complet possible. La difficulté réside dans le fait d'essayer de mettre en place des analyses diagnostics logique, telle que celle utilisés par la communauté d'Intelligence Artificielle, appliquées à des systèmes dynamiques complexes modélisés avec des équations différentielles.

Dans une procédure de diagnostic, comme celle que nous avons abordées dans le projet MAGIC, les tests de détection peuvent être réalisés en utilisant des algorithmes différents (détection à base de signal, détection utilisant des graphes causaux, observateurs d'état, relations de parité, détection à base des réseaux de neurones,...). La phase d'analyse diagnostique doit donc être en mesure de prendre en compte des tests hétérogènes pour pouvoir les exploiter par un algorithme d'analyse diagnostique. De plus, différents tests de détection peuvent vérifier des états de composants communs. Deux tests différents peuvent s'intéresser à un même composant ou un ensemble de composants. L'analyse diagnostique doit alors être capable d'appréhender ce type de situation.

Un autre problème qui s'offre à nous est qu'un système de diagnostic peut, dans un contexte donné, conduire à un diagnostic erroné mais il faut que si ce contexte se représente, le système de diagnostic ne reproduise pas la même erreur. Cela induit donc des propriétés de garantie de la part de l'analyse diagnostic afin d'être en mesure de reconnaître les tests de détection ayant produit des symptômes erronés. Si les sources d'erreur peuvent provenir à la fois des tests de détection et de l'analyse diagnostique, il n'est plus possible de remonter à la cause d'une erreur.

Les diagnostics doivent être garantis si les tests de détection sont justes. Par ailleurs, un test de détection vrai ne prouve pas l'absence de défaut (notion d'exonération). Il est possible qu'en certains points de fonctionnement, un comportement défaillant s'apparente à un comportement normal. Ce n'est donc pas parce qu'un comportement s'apparente au comportement normal qu'il est possible de conclure à l'absence de défauts. Néanmoins, les tests de détection consistants peuvent fournir une information intéressante pour l'analyse diagnostique. Alors, comment peut-on l'exploiter afin de donner à l'opérateur un diagnostic encore plus précis ?

Le diagnostic peut être influencé par le fait qu'il peut y avoir des incertitudes dans la décision prise par les tests de détection. Tous les tests ne conduisent pas toujours à des décisions sûres. En effet, l'incertitude se caractérise par l'incapacité de savoir si un test de détection est vrai ou faux [Dubois et al, 1994]. Une information est dite imprécise, si elle s'avère incomplète, insuffisante

pour l'usage que l'on voudrait en faire. L'incertitude peut avoir plusieurs sources, parmi lesquelles l'imprécision. Une mesure imprécise engendre l'incertitude sur la valeur de la grandeur mesurée. Dans le cas du diagnostic, les résidus sont affectés par les imprécisions des mesures et par les erreurs de modèle, d'où l'incertitude qui en résulte sur la prise de la décision (phase de détection). Limiter la décision d'un test de détection à deux résultats possibles (alarme ou non) ne permet pas d'exprimer une incertitude de décision du type : une anomalie est soupçonnée sans que cela soit certain. Les tests de détection mis en oeuvre dans le cadre du projet MAGIC fournissent des décisions comprises entre 0 et 1 où 1 représente une inconsistance sûre et 0 une consistance sûre. L'analyse diagnostic doit pouvoir exploiter ces résultats.

De plus, dans le cadre des systèmes industriels tels que ceux que nous avons abordés dans le projet MAGIC, pour des raisons liées à des capacités de calcul, tous les tests de détection ne peuvent pas être actifs en permanence. L'objectif consiste à déterminer quels sont les tests de détection les plus intéressants à déclencher. C'est le rôle de la stratégie de diagnostic qui a besoin de séparer clairement la détection des symptômes (éventuellement localisation locale) de la localisation globale des défauts (diagnostic).

La procédure de diagnostic est appliquée sachant que tous les défauts ne sont pas équiprobables. En effet, une information a priori sur la probabilité d'un composant de tomber en panne peut être disponible. Comment peut-on l'exploiter afin de rendre le résultat diagnostique plus précis pour l'opérateur ? Les algorithmes proposés permettent de déduire tous les défauts possibles pour un comportement observé en les classant suivant différents critères de vraisemblance.

De Kleer représente l'opération de diagnostic orienté bon fonctionnement par la figure 1.3 ; elle montre clairement que les différences structurelles se déduisent des différences de comportements.

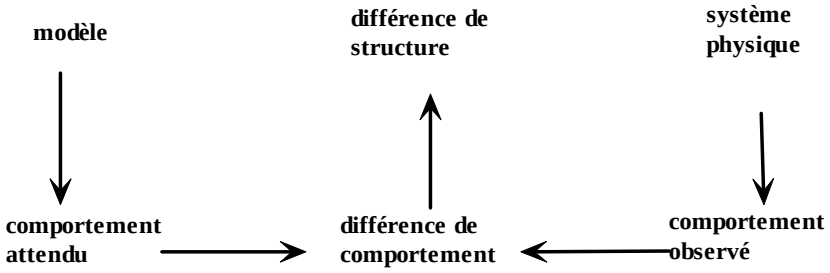


Figure 1.3 – Procédure de diagnostic orientée Bon fonctionnement d'après De Kleer (1987)

De Kleer décrit les étapes du diagnostic *bon fonctionnement* en terme de *conflit* et de *candidat*. Un conflit apparaît lorsqu'un symptôme est reconnu c'est-à-dire lorsqu'une incohérence entre le fonctionnement d'un ensemble d'éléments et leur fonctionnement attendu est constatée. Le conflit est la réunion des éléments de cet ensemble qui ne peuvent pas tous fonctionner correctement à la fois tandis qu'un candidat est un ensemble d'hypothèses sur le mauvais fonctionnement des éléments d'un système physique.

Le diagnostic permet d'identifier les éléments défaillants d'un système physique c'est-à-dire qu'il faut commencer par recenser les conflits en recherchant tous les conflits minimaux possibles.

Cela revient à remonter des symptômes aux éléments physiques en utilisant la topologie et les connaissances fonctionnelles des organes. Il faut alors générer des candidats minimaux cohérents avec l'ensemble des conflits ; autrement dit, déterminer quelles sont les hypothèses de mauvais fonctionnement d'éléments physiques minimum qui expliquent les conflits observés. Pour Reiter [Reiter, 1987], il s'agit de *déterminer les constituants du système qui, en les supposant défaillants, expliqueraient les contradictions entre les éléments observés et corrects.* Une fois la fonction défaillante déterminée, il ne reste plus qu'à exploiter l'information contenue dans la représentation structurelle pour reconnaître l'élément physique défaillant. Pour ce faire, il fallait proposer un formalisme pour la modélisation des systèmes dynamiques complexes qui permettent de mettre en place des analyses diagnostiques logique telle que celle utilisés par la communauté DX. Le formalisme proposé doit être capable de tenir compte de la validité du modèle utilisé dans l'analyse diagnostic.

I.6. LES APPROCHES USUELLES DE DETECTION

Dans ce paragraphe, on présente différentes méthodes utilisées en diagnostic de systèmes physiques. Le domaine était très vaste, des choix arbitraires ont été faits. Le but n'est donc pas de faire une synthèse exhaustive de l'existant, mais de montrer la richesse des possibilités qui s'offrent au concepteur de système de diagnostic. En effet, différents types d'algorithmes de détection dédiés aux systèmes physiques ont été conçus par les chercheurs de la communauté de l'Automatique [Frank et al., 2000][Isermann et al, 1997][Patton et al, 1997]. Néanmoins, on s'est astreint à balayer le large spectre des techniques actuellement utilisées en diagnostic, à savoir :

- Méthodes sans modèle analytique
- Méthodes basées sur les modèles analytiques

Actuellement, on s'oriente vers des systèmes de diagnostic mettant en oeuvre différentes techniques de détection. En effet, chacune d'entre elles est plus ou moins bien adaptée pour appréhender tel ou tel type de défaut. Par exemple, on s'orientera vers les méthodes à base d'estimation paramétrique lorsqu'on souhaite localiser un défaut qui se manifeste par une variation des paramètres du modèle identifié.

I.6.1. Méthodes sans modèle analytique

I.6.1.1. Analyse fréquentielle (Filtrage)

Une première approche du traitement du signal repose sur l'analyse fréquentielle (transformée de Fourier). Elle est bien évidemment très utilisée pour la détection de phénomènes périodiques comme en analyse vibratoire. Le contenu spectral des signaux est utilisé depuis de nombreuses années pour détecter des défauts dans les machines électriques [Camron et Thomson, 1986][Thomson, 1999] tels que les ruptures de barres au rotor des machines asynchrones, la dégradation des roulements, les décentrages, les courts-circuits dans les bobinages. Avec le développement des applications à vitesse variable, les recherches actuelles portent plus particulièrement sur les méthodes adaptées à la caractérisation de signaux non stationnaires : temps-

fréquence, temps-échelle (décomposition en ondelettes [Lesecq et al, 2001][Petropol, 2001] [Borras, 1999]).

L'analyse du spectre des signaux issus des capteurs permet de déterminer très efficacement l'état de l'installation sous surveillance. Les signaux sont ici tout d'abord analysés en état normal de fonctionnement. Ensuite, toute déviation des caractéristiques fréquentielles d'un signal est reliée à une situation de panne (le problème, c'est qu'un changement de consigne modifie les caractéristiques fréquentielles et cela n'a rien d'un défaut).

Cette approche possède l'avantage d'être relativement simple à mettre en pratique, mais l'inconvénient d'être assez sensible aux bruits de mesure quand ceux-ci coïncident avec la zone fréquentielle d'intérêt. De plus un échantillonnage fréquent est nécessaire pour permettre de reconstituer le signal de départ tout en minimisant la perte de fréquence [Steyer et al, 2001].

I.6.1.2. Redondance matérielle

Cette méthode consiste à multiplier physiquement les capteurs critiques d'une installation. Un traitement des signaux issus des éléments redondants effectue des comparaisons et distingue l'élément défectueux en cas d'incohérence. Cette méthode est pénalisante en termes de poids, puissance consommée, volume et coût (d'achat et de maintenance). Elle est donc essentiellement réservée aux cas où la continuité de service est obligatoire (e.g. l'aérospatiale, le nucléaire). En effet, elle apporte l'avantage, une fois la défaillance détectée et localisée, de pouvoir utiliser la partie de l'équipement encore saine mais cette technique ne s'applique généralement que sur des capteurs.

I.6.1.3. Capteurs spécifiques (capteurs-détecteurs)

Des capteurs spécifiques peuvent également être utilisés pour générer directement des signaux de détection ou connaître l'état d'un composant. Par exemple, les capteurs de fin de course, d'état de fonctionnement d'un moteur ou de dépassement de seuils sont largement employés dans les installations industrielles.

I.6.1.4. Réseaux de neurones artificiels

Quand la connaissance sur le procédé à surveiller n'est pas suffisante et que le développement d'un modèle de connaissance du procédé est impossible, l'utilisation de modèle dit « boîte noire » peut être envisagée. Pour cela des réseaux de neurones artificiels (RNA) ont été utilisés. Leur application dans les domaines de la modélisation, de la commande et du diagnostic a largement été rapportée dans la littérature (par exemple, [Bischop, 1994] et [Narendra, 1990]).

Un RNA est en fait un système informatique constitué d'un nombre de processeurs élémentaires (ou nœuds) interconnectés entre eux qui traite -de façon dynamique- l'information qui lui arrive à partir des signaux extérieurs.

De manière générale, l'utilisation des RNA se fait en deux phases. Tout d'abord, la synthèse du réseau est réalisée et comprend plusieurs étapes : le choix du type de réseau, du type de neurones, du nombre de couches, des méthodes d'apprentissage. L'apprentissage permet alors, sur

la base de l'optimisation d'un critère, de reproduire le comportement du système à modéliser. Il consiste en la recherche d'un jeu de paramètres (les poids) 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) quand les résultats d'apprentissage obtenus par le RNA sont satisfaisants, il peut être utilisé pour la généralisation. Il s'agit ici de la deuxième phase où de nouveaux exemples – qui n'ont pas été utilisés pendant l'apprentissage – sont présentés au RNA pour juger de sa capacité à prédire les comportements du système modélisé.

Comme il a été dit précédemment, les RNA peuvent être utilisés pour le diagnostic des défaillances. Leur faible sensibilité aux bruits de mesure, leur capacité à résoudre des problèmes non linéaires et multivariables, à stocker la connaissance de manière compacte, à apprendre en ligne et en temps réel, sont en effet autant de propriétés qui les rendent attrayants pour cette utilisation. Leur emploi peut alors se faire à trois niveaux :

- comme modèle du système à surveiller en état normal et générer un résidu d'erreur entre les observations et les prédictions,
- comme système d'évaluation de résidus pour le diagnostic,
- ou comme système de détection en une étape (en tant que classificateurs).

I.6.1.5. Systèmes d'inférence flous

Pendant les vingt dernières années, les systèmes d'inférence floue (SIF) – dont les bases relèvent de la théorie des ensembles flous de Zadeh [Zadeh, 1965] – sont devenus très populaires. Les applications dans le traitement du signal, la modélisation, la commande, la supervision de procédés et la prise de décision sont en effet autant d'applications qui démontrent la capacité des SIF à traiter des problèmes non linéaires grâce à l'utilisation de connaissances expertes.

La structure de base d'un SIF est constituée de :

- Un univers de discours qui contient les fonctions d'appartenance des variables d'entrée et de sortie à des classes. Ces fonctions peuvent avoir différentes formes, les plus usuelles étant les formes triangulaires, trapézoïdales, et gaussiennes,
- Une base de connaissance qui regroupe les règles liant les variables d'entrées et de sorties sous la forme « Si…Alors… »,
- Un mécanisme de raisonnement qui base son fonctionnement sur la logique du *modus ponens* généralisé.

Les SIF peuvent être qualifiés de méthode « boîte grise ». En effet, ils explicitent la connaissance experte sous la forme de règles d'inférence, tout en classant les entrées et les sorties de façon qualitative. Ils effectuent également des calculs sur la base de poids et de fonctions fixées de façon à faire correspondre des comportements observés sans qu'il y ait de signification physique explicite. Le mode de fonctionnement d'un SIF permet donc de manier indistinctement des quantités et des symboles, de les manipuler pour faire des calculs et d'expliquer le cheminement

parcouru pour obtenir un résultat. D'autre part, les tâches de diagnostic reposent sur des quantités d'heuristiques difficiles à formaliser dans un modèle mathématique :

- La corrélation entre des variables très différentes,
- Des observations qualitatives (par exemple : couleur, bruit),
- Des intuitions, liées à des statistiques (par exemple : tel appareil pose plus de problèmes que tel autre…) difficilement quantifiables mais pourtant très efficaces.

I.6.2. Méthodes basées sur les modèles analytiques

La plupart des méthodes de détection et de diagnostic en ligne s'appuient sur des les mesures. Il existe des méthodes qui utilisent plus de connaissances que celles apportées par les seuls capteurs physiques. Ces connaissances peuvent en particulier provenir de la connaissance du comportement entrée /sortie d'un procédé ou des processus qui en gouverneraient l'évolution. Cette connaissance est généralement exprimée sous forme de modèles mathématiques.

Les recherches dans ce domaine ont commencé il y a bientôt une quarantaine d'années avec l'utilisation simultanée de diverses techniques. Cette diversité a conduit de nombreux auteurs à réaliser récemment un gros effort de synthèse pour unifier les points de vues, le langage et montrer des équivalences entre les méthodes [Chen et Patton, 1999] [Gertler, 1998]. Parmi les différentes méthodes de détection utilisant des modèles mathématiques, nous trouverons principalement l'espace de parité, les Observateurs et l'estimation paramétrique.

I.6.2.1. Espace de parité

La méthode de l'espace de parité a été une des premières méthodes employées à des fins de FDI [Chow et Wilsky, 1984][Gertler et al, 1990][Gertler, 1997]. Son nom provient du domaine de l'informatique où le contrôle de parité se faisait dans les circuits logiques. Le principe de la méthode est la vérification de la consistance existante entre les entrées et les sorties du système surveillé (voir figure 1.4).

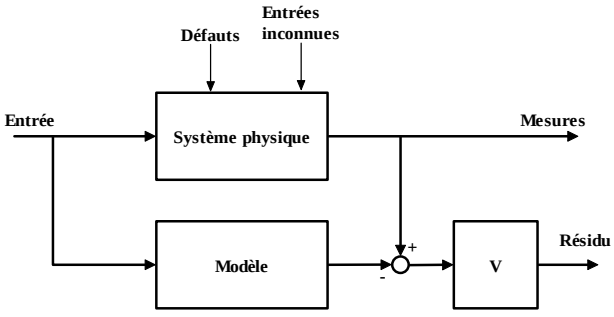


Figure 1.4 - Approche de l'espace de parité dans un format entrée-sortie

I.6.2.2. Observateurs

Par la suite, nous ferons référence à la stratégie appelée par certains auteurs « par observateurs » ou encore « en boucle fermée » par opposition aux méthodes en boucle ouverte, qui correspondent à celles de l'espace de parité décrite précédemment.

Beard et Jones ont été en fait les premiers à proposer le remplacement de la redondance matérielle par des algorithmes de détection basés sur des observateurs [Beard, 1971][Jones, 1973]. Leurs travaux concernaient des systèmes linéaires et la méthode a été appelée « filtre de détection de défauts de Beard-Jones ». D'autres travaux ont suivi cette voie en l'étendant aux systèmes non linéaires (cf. par exemple [Alcorta, 1997][Frank, 1996]).

I.6.2.3. Estimation paramétrique

L'approche d'estimation paramétrique mesure l'influence des défauts sur les paramètres et non plus, comme précédemment, sur les variables du système physique.

Le principe consiste à estimer en continu des paramètres du procédé en utilisant les mesures d'entrée/sortie et en l'évaluation de la distance qui les sépare des valeurs de référence de l'état normal du procédé (Figure 1.5).

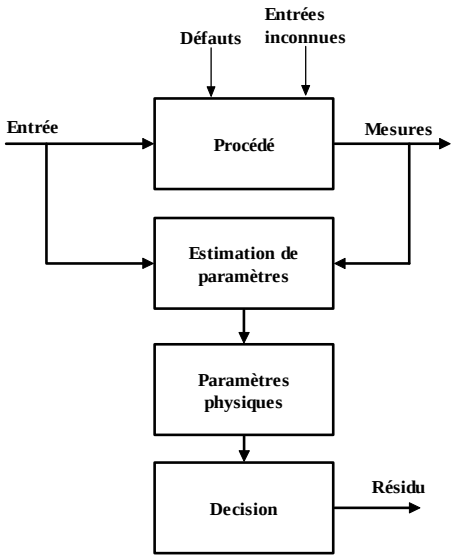


Figure 1.5 - Estimation paramétrique pour la détection et le diagnostic de défauts.

L'estimation paramétrique possède l'avantage d'apporter de l'information sur la taille des déviations. Toutefois, un des inconvénients majeurs de la méthode réside dans la nécessité d'avoir un système physique excité en permanence. Ceci pose des problèmes pratiques dans le cas de procédés dangereux ou fonctionnant en mode stationnaire. De plus, les relations entre les paramètres mathématiques et physiques ne sont pas toujours inversibles de façon unitaire, ce qui complique la tâche du diagnostic basé sur les résidus.

I.6.2.4. Graphes causaux

Un autre outil proposé par l'IA est le raisonnement causal. Le diagnostic est typiquement un processus causal [Gentil et al, 2004][Garcia-Beltran, 2004]. Il consiste à déterminer les composants défectueux qui peuvent expliquer le fonctionnement anormal observé. Un aspect significatif de la connaissance exigée au moment de l'analyse des régimes perturbés est la compréhension des mécanismes en termes de causalité. Une structure causale est une description des effets que les variables peuvent avoir les unes sur les autres. Cela peut être représenté par un graphe orienté (digraphe). Les nœuds sont les variables et les arcs symbolisent les relations entre elles. Cette

structure fournit un outil conceptuel pour le raisonnement à propos de la façon dont les changements normaux ou anormaux se propagent au sein du procédé. La méthode de modélisation causale est fondée sur une représentation causale qualitative du fonctionnement normal du procédé et sur des modèles locaux de comportement quantitatifs.

Le comportement normal de n'importe quel dispositif industriel de procédé de fabrication peut être partiellement décrit par un graphe causal, composé de relations continues entre les variables. La transcription des modèles de comportement normal en termes de graphes orientés fournit un outil de raisonnement au sujet des lois physiques régissant le dispositif et fournit ainsi une manière normale de mettre en œuvre le diagnostic. Les nœuds sont les variables mesurées concernant le niveau exigé d'abstraction et les arcs représentent les liens orientés entre elles. De cette façon, $x \rightarrow y$ signifie que l'état de y au temps t dépend de l'état de x au $t' < t$ de temps; x est la cause, y l'effet. Ce principe est valide aussi longtemps que les liens causaux ne sont pas modifiés par les défauts. Par ailleurs le graphe causal fournit un outil graphique pour la visualisation de la propagation du défaut de variable en variable sur l'interface opérateur.

Un graphe causal peut être obtenu par une analyse physique soignée et une analyse fonctionnelle descendante du processus. Des nœuds sont choisis comme variables significatives pour l'opérateur de surveillance. Ce genre de modèle est généralement concentré sur des phénomènes de transport de matière (équilibres, transferts, stockage…). Les paramètres temporels dans les relations dynamiques soutenues par les arcs peuvent être obtenus avec des procédures classiques d'identification.

I.7. SPECIFICITE D'UN CONTEXTE DISTRIBUE

Le diagnostic de défauts devient très complexe quand les systèmes physiques à diagnostiquer ne sont plus élémentaires. La plupart des systèmes de diagnostic sont centralisés. Le module de diagnostic centralisé est contraint d'avoir une vue globale du système physique et contient la description du système physique dans sa totalité. Il reçoit toutes les observations via les capteurs et doit exécuter un algorithme de détection afin de générer des symptômes. La majorité des systèmes de diagnostic reposent sur des tests de détection de même nature alors que des algorithmes de détections très variés sont généralement utiles pour appréhender les diverses informations disponibles sur un système à surveiller. Les applications industrielles complexes requièrent généralement l'utilisation de plusieurs types d'algorithmes de détection. L'approche de diagnostic centralisé rencontre quatre problèmes principaux [Debouk et al, 2000] [Su et al, 2000].

- Problèmes liés à la complexité du système : l'état du système est habituellement l'ensemble des états des composants.
- Problèmes liés à une architecture peu robuste : en effet, quel qu'il soit, le problème se produisant à l'intérieur d'un système centralisé de diagnostic peut conduire à l'échec total du système de diagnostic. En revanche, dans un système distribué, la défaillance d'un module (par exemple de détection, ou d'acquisition,…) qui participe au diagnostic ne met pas en panne toute la procédure de diagnostic.
- Problèmes liés à une architecture figée : toute modification ou évolution structurelle du système nécessite une réécriture plus au moins complète du programme de diagnostic.
- Problèmes liés à la distribution des calculs : en effet, toutes les tâches d'une procédure de diagnostic sont exécutées sur un calculateur centralisé. Distribuer les différents calculs liés au diagnostic (par exemple les tests de détection) sur des machines différentes s'avère difficile dans une approche centralisée.

Pour résoudre ces problèmes, un groupe de travail européen s'est formé dans le cadre du projet de recherche et de développement intitulé (MAGIC) « *Multi-Agents-based Diagnostic Data Acquisition and Management in Complex Systems*". Ce projet vise à concevoir une architecture logicielle distribuée et fournir un ensemble d'outils qui peuvent être utilisés pour le diagnostic des systèmes complexes et dynamiques [Köppen-Seliger et al., 2002]. Le système MAGIC est basé sur le paradigme multi-agent. Chaque agent du système MAGIC est spécialisé. Chaque algorithme de détection est encapsulé dans un agent capable de communiquer avec le reste des agents du système MAGIC. Le premier avantage à utiliser les systèmes multi-agents est de conduire à des systèmes modulaires et ouverts. En effet, le fait d'ajouter un agent ou de modifier la structure d'un système n'induit pas une re-conception d'une solution mais converge automatiquement, à la suite d'un processus d'apprentissage, vers une nouvelle solution globale. L'autre grand avantage est de permettre de résoudre des problèmes complexes sans avoir à les décomposer en une multitude de problèmes plus élémentaires, mais en construisant des entités autonomes qui pourront, en coopérant, participer à la construction d'une solution globale.

I.8. DIAGNOSTIC DISTRIBUE VS HIERARCHIQUE

Le diagnostic hiérarchique centralisé est l'approche la plus répandue dans la communauté automatique. Elle a été appliquée, à titre d'exemple, aux systèmes de fabrication comme celui étudié dans [Martin et al., 2002]. Dans les systèmes de diagnostic hiérarchisés, un contrôleur central organise la procédure de diagnostic et l'applique à un système physique. L'outil de diagnostic à base de modèle peut être vu comme un seul algorithme de diagnostic car, cet algorithme doit avoir une vue globale du système à diagnostiquer. En effet, les différentes phases d'une procédure de diagnostic (acquisition, détection et localisation, maintenance) sont programmées par un seul algorithme. Par conséquent il doit avoir accès au modèle global du système [Provan, 2002]. Cependant, il y a plusieurs raisons qui expliquent qu'une telle approche est peu appropriée. En premier lieu, si le système à diagnostiquer a une grande taille et si la période d'échantillonnage est courte alors l'approche centralisée n'aura pas assez de temps pour collecter toutes les observations

des capteurs et par la suite calculer un diagnostic. En deuxième lieu, si la structure du système est dynamique, alors le système peut changer très vite et on ne peut pas maintenir un modèle global exact du système. Par exemple la supervision de systèmes tels que les réseaux de télécommunications s'avère très difficile avec une approche centralisée. Étant donnée la taille d'un tel système, une approche diagnostiqueur de type centralisé [Rosé, 1997][Sampath et al, 1998] n'est pas implantable car elle nécessite la mise en place d'un modèle global du système. Quelquefois, l'existence d'un modèle global du système à diagnostiquer n'est pas possible ou n'est pas souhaitable.

Il n'est plus possible [Pencolé et al , 2002] de se contenter d'une approche centralisée et figée pour la conception d'un système de diagnostic pour les systèmes industriels complexes. L'avenir appartient aux systèmes ouverts distribués, c'est-à-dire capables de se construire élément par élément, apprenants et communicants, de se découvrir mutuellement les uns les autres pour déterminer l'état physique du procédé industriel à partir d'un ensemble de valeurs reçues par les capteurs présents sur le système et la description du système physique. Les années 1990 ont vu émerger la notion d'intelligence distribuée avec l'apparition des systèmes multi-agents. L'idée est de multiplier les entités intelligentes et de les doter de facultés de communication afin qu'elles aient le moyen de coopérer entre elles pour que se construise par coopération une solution globale à un problème donné. Ces technologies constituent un élément de réponse aux problèmes qui se posent au domaine de la surveillance, sûreté, supervision et diagnostic (S^3D).

Bien que le diagnostic distribué puisse permettre de mieux répartir la charge de calcul, de délester les canaux de communication et surtout d'isoler les différents rôles d'une procédure de diagnostic, de concevoir des systèmes modulaires capables de s'adapter facilement à des changements de structure ; en revanche, en pratique ceci peut s'avérer difficile à implémenter [Martin et al., 2002] [Leitao et al, 2000]. Le problème principal est que dans un système distribué, aucune entité ne dispose d'une information globale et entière sur le système à étudier. En effet, à un instant donné, dans une approche distribuée, chaque entité a une vue partielle ou locale du système à diagnostiquer. Par conséquent, pour construire une solution globale, logique et cohérente il faut pouvoir rassembler les solutions partielles. Donc, il faut que les entités coopèrent entre elles afin de partager leurs solutions et faire part de leurs problèmes et coordonner leurs activités.

Les systèmes multi-agents ont des applications dans le domaine de l'intelligence artificielle où ils permettent d'appréhender la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents [Ferber, 1995]. Nous parlons alors d'intelligence artificielle distribuée. Cette méthode s'applique, par exemple, à la surveillance d'un processus industriel comme le système DIAMON [Lackinger et Nejdl, 1993] où elle met en oeuvre la solution de bon sens qui consiste à coordonner plusieurs surveillants spécialisés, plutôt qu'à envisager un seul surveillant omniscient.

I.9. CONCLUSION

L'objectif de ce chapitre a été de présenter les différents travaux dans le domaine du diagnostic des systèmes industriels, et de justifier le contexte dans lequel nous nous sommes placés

afin d'élaborer notre système de diagnostic dédié aux systèmes complexes. Nous avons commencé par préciser et parfois par établir un certain nombre de notions fondamentales ainsi que les principes du diagnostic à base de modèles analytiques des systèmes physiques. Nous avons établi que toute procédure de diagnostic se ramène nécessairement à une série de tests de détection entre des observations et des modèles.

Le diagnostic est une procédure habituellement décomposée en deux étapes. L'étape de détection qui produit des symptômes. Ces symptômes sont généralement déduits des variables physiques connues (telles que les variables de contrôle), et de test de détection à base de (modèles analytiques, d'algorithmes de traitement de signal,…). Ces tests de détection vérifient la cohérence entre les données et les modèles. L'étape de localisation des défauts collecte les symptômes des tests de détection. Elle fournit les diagnostics possibles. Un diagnostic peut être défini littéralement comme un état plausible (vraisemblable, crédible) du système physique. Différentes approches de diagnostic ont été développées par la communauté du l'automatique [Frank, et al, 2000] [Gertler, 1998] [Isermann et Balle, 1997] [Patton, 1997]. Le point commun entre toutes ces approches est qu'elles se basent sur le modèle analytique du procédé. Ce modèle est utilisé afin de produire des symptômes, appelés aussi résidus. Une table de signature (les colonnes correspondent aux défauts et les lignes correspondent aux symptômes) (cf. chapitre 2) est probablement l'outil le plus simple à utiliser afin d'isoler un défaut avec plusieurs symptômes.

Par ailleurs, nous avons étudié la spécificité d'un contexte distribué d'une procédure de diagnostic dédiée aux systèmes industriels complexes par rapport à un contexte centralisé. En effet, l'architecture centralisée prône un noyau central qui regroupe toute la procédure de diagnostic, à savoir, l'acquisition des données, la détection, la localisation, le pronostic et la maintenance, autour duquel toutes les entités sont regroupées (ou *centralisées*). Ce noyau central prend la plupart des actions. L'avantage est une facilité d'administration. En revanche, l'architecture distribuée d'une procédure de diagnostic prône plusieurs surveillants spécialisés, voire le plus possible. Cette architecture a l'avantage de permettre une plus grande souplesse et une plus grande autonomie des entités. C'est une architecture ouverte qui permettra d'intégrer dans le futur de nouveaux algorithmes de détection. Pour réaliser une telle architecture nous avons proposé de la concevoir en utilisant le paradigme multi-agent que nous étudierons plus en détails dans le chapitre 5 de la deuxième partie de ce document qui apporte des solutions technologiques.

Chapitre 2

Comparatif entre les approches d’isolation existantes

II.1. INTRODUCTION

Le diagnostic est un thème de recherche qui peut être approché de façons très différentes suivant le type de connaissances dont on dispose. L'approche dite « à base de modèles » (*Model Based Diagnosis*) s'appuie sur l'utilisation d'un modèle analytique du système à diagnostiquer. Deux communautés scientifiques ont utilisé l'approche à base de modèles. La communauté FDI (*Fault Detection and Isolation*) utilise des techniques provenant de la théorie de la commande et de la décision statistique. Cette approche a atteint un stade de réelle maturité et a produit de nombreux résultats [Patton et al.,1991][Frank, 1996][Iserman et al., 1997]. La communauté DX est apparue plus récemment, et se rattache plutôt aux domaines de l'informatique et de l'intelligence artificielle [Reiter, 1987][de Kleer et al., 1987][Hamscher et al., 1992].

L'idée directrice de ce chapitre est de proposer une étude comparative entre l'approche FDI et l'approche DX. Pour ce faire, nous avons choisi d'appliquer le raisonnement de chaque approche sur deux systèmes physiques différents, le premier est un système dynamique et le second est un système statique. L'étude a pour objectif de révéler les limites de chacune de ces approches face à des systèmes différents. Cette analyse comparative se place dans la continuité de ce qui a été déjà fait par le groupe de travail national IMALAIA dans [Cordier et al., 2004], qui a pour objectif de déterminer une terminologie commune à FDI et à DX, d'identifier les similarités et les complémentarités des deux approches et de participer à la création d'une structure commune, profitant de la synergie des deux approches.

Le présent chapitre est organisé de la manière suivante. La section 2 présente nos exemples d'application, un système de deux bacs en cascade et un circuit logique qui représente un additionneur. Le principe de la méthode par redondance analytique de la communauté FDI est rappelé par la section 3, en mettant l'accent sur la méthode utilisée pour la prise de décision. Cette méthode est l'analyse par table de signature. Cette approche traite principalement des symptômes générés à partir des modèles de composants représentant le mauvais fonctionnement. En revanche, la section 4 présente l'approche basée sur le raisonnement logique de la communauté DX, originairement dédiée aux systèmes binaires statiques, qui propose un raisonnement logique basé sur des modèles de bon et de mauvais fonctionnement des composants d'un système physique. En décrivant brièvement les différentes techniques de Davis, De Kleer et de Reiter, le paragraphe 4 met en avant une étude comparative des deux approches afin de justifier notre choix à utiliser l'approche logique qui s'inspire des travaux sur le diagnostic à base de la consistance dans le but de proposer des diagnostics sûrs et exhaustifs.

II.2. PRESENTATION DES DEUX EXEMPLES D'APPLICATION

Les deux systèmes choisis comme support pour l'analyse comparative et l'illustration des limites des approches existant (FDI et DX) pour le diagnostic à base de modèle sont : le système des deux bacs en cascade de la figure 2.1, bien connu comme étant un système continu dynamique. Le deuxième système étudié est un système binaire statique. Ce dernier est un circuit logique présenté à

la (figure 2.2), qui est un additionneur constitué d'un ensemble de portes logiques issu de [Reiter, 1987], repris dans différents travaux, par exemple dans [Dague, 2001].

II.2.1. Système des deux bacs

Le système des deux bacs en cascade (figure 2.1) est constitué d'un ensemble de composants élémentaires, à savoir le bac du haut (BH), le bac du bas (BB), une vanne de Kammer du haut (VKH) et une vanne de Kammer du Bac (VKB), une restriction (R_1) et (R_2) au niveau du bac du haut et du bac du bas respectivement, différents capteurs qui nous permettent de mesurer la hauteur du produit dans le bac du haut (H_1) et le bac du bas (H_2) et le flux d'entrée du produit (q_{e1}) et (q_{e2}) dans le bac du haut et le bac du bas respectivement. Les flux d'entrée sont commandés directement (boucle ouverte) par un calculateur numérique. Le point intéressant pour le diagnostic des défauts est que l'information obtenue du procédé a un temps d'échantillonnage de 10s (le contrôleur utilise toute l'information) dans la simulation proposée.

II.2.2. L'additionneur

Le circuit logique, qui représente l'additionneur, est constitué de cinq composants : deux portes ET, deux portes OU-EXCLUSIF et une porte OU est décrit par la (figure 2.2). La porte ET prédit un 0 en sortie à partir du moment où une des entrées vaut 0, et prédit un 1 en sortie dès que les deux entrées valent 1. La porte OU prédit un 1 en sortie à partir du moment où une des entrées vaut 1, prédit un 0 du moment que les deux entrées valent 0. Enfin, la porte OU-EXCLUSIF prédit un 1 en sortie du moment qu'une entrée vaut 0 (resp. 1) et l'autre entrée vaut 1 (resp. 0), prédit un 0 en sortie du moment que les deux entrées sont à 1 ou à 0.

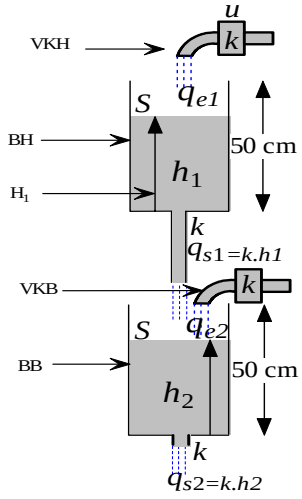


Figure 2.1 - Système des deux bacs en cascade

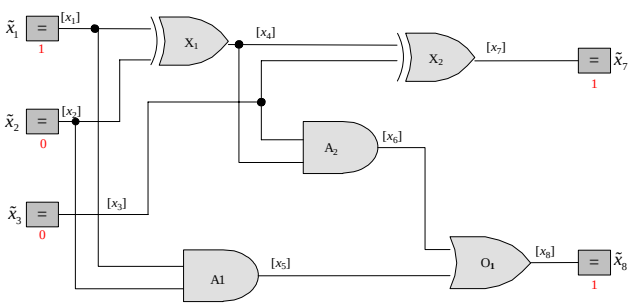


Figure 2.2 - Additionneur

La modélisation d'un système physique en général est une description de sa structure et une représentation comportementale ou fonctionnelle de chacun de ses composants [Milne, 1987]. Une représentation comportementale est constituée de relations entre divers phénomènes du processus, appelées classiquement relations de causes à effets. Une représentation fonctionnelle, constituée de relations de même nature que le niveau comportemental, est plus abstraite puisqu'elle ne s'intéresse qu'aux fonctions présumées que le système physique doit remplir. Le niveau structurel, quant à lui,

s'appuie sur la structure réelle du système physique et évoque les interconnections entre ses différents éléments ou constituants. Ces trois niveaux de représentation permettent de préciser ce qu'il est possible de faire en diagnostic avec un modèle donné. Les niveaux comportementaux et fonctionnels comprennent des relations entre des grandeurs physiques et permettent de mettre en évidence la présence d'un événement anormal ou d'une anomalie. Le niveau structurel, quant à lui, permet de déterminer l'élément affecté par le défaut. L'intérêt de cette décomposition est de rappeler que, puisqu'un modèle contient toute l'information relative à un système physique et utilisable ensuite par la procédure de diagnostic, son choix doit être en adéquation avec le cahier des charges fixé.

Pour les deux bacs en cascades, la connaissance physique du système a permis d'écrire les relations non linéaires présentées ci-dessous (cf. figure 2.3). Chaque relation décrit le comportement d'un composant. Le modèle de chaque composant exprime les contraintes qu'il impose entre les variables qui lui sont liées sur la base des équations du modèle du procédé (MP), c'est-à-dire l'ensemble des relations définissant le comportement du système, et le modèle des informations (MI), c'est-à-dire l'ensemble des variables du système qui sont renseignées (observées). Ces dernières sont aussi des contraintes analytiques ; elles décrivent de surcroît le comportement des capteurs et actionneurs. On obtient le modèle du système, décrit par les relations suivantes :

Un modèle du procédé (MP) Modèle des informations (MI)

$$\begin{cases} M(BH):\left(S_1 \dfrac{dh_1}{dt} = q_{e1} - q_{s1}\right) \\ M(RH):\left(q_{s1} = k_1 h_1\right) \\ M(BB):\left(S_2 \dfrac{dh_2}{dt} = q_{e2} + q_{s1} - q_{s2}\right) \\ M(RB):\left(q_{s2} = k_2 h_2\right) \end{cases} \qquad \begin{cases} M(VKH):\left(\tilde{q}_{e1} = [q_{e1}]\right) \\ M(VKB):\left(\tilde{q}_{e2} = [q_{e2}]\right) \\ M(H_1):\left(\tilde{h}_1 = [h_1]\right) \\ M(H_2):\left(\tilde{h}_2 = [h_2]\right) \end{cases}$$

Figure 2.3 - Le modèle du système des deux bacs

L'additionneur est décrit par des relations linéaires présentées ci-dessous (cf. figure 2.4). Le modèle du système est constitué du modèle du procédé (MP), et du modèle des informations (MI). Le modèle de chaque composant exprime les contraintes qu'il impose entre les variables qui lui sont liées.

Un modèle du procédé (MP) Modèle des informations (MI)

$$\begin{cases} M(XOR_1) \Rightarrow \left(x_4 = [x_1 \oplus x_2]\right) \\ M(XOR_2) \Rightarrow \left(x_7 = [x_4 \oplus x_3]\right) \\ M(AND_1) \Rightarrow \left(x_6 = [x_2 \wedge x_4]\right) \\ M(AND_2) \Rightarrow \left(x_5 = [x_2 \wedge x_1]\right) \\ M(OR) \Rightarrow \left(x_8 = [x_5 \vee x_6]\right) \end{cases} \qquad \begin{cases} M(x_1):\left(\tilde{x}_1 = [x_1] = 1\right) \\ M(x_2):\left(\tilde{x}_2 = [x_2] = 0\right) \\ M(x_3):\left(\tilde{x}_3 = [x_3] = 0\right) \\ M(x_7):\left(\tilde{x}_7 = [x_7] = 1\right) \\ M(x_8):\left(\tilde{x}_8 = [x_8] = 1\right) \end{cases}$$

Figure 2.4 - Le modèle du système « additionneur »

II.3. L'APPROCHE DU DIAGNOSTIC PAR REDONDANCE ANALYTIQUE

II.3.1. Principe de la méthode

Le diagnostic par redondance analytique suit une démarche qui peut se résumer comme suit : on dispose d'un modèle décrivant le fonctionnement normal du système et on surveille le fonctionnement réel en testant la cohérence entre ce modèle et les observations disponibles. Si celles-ci ne vérifient pas les équations du modèle, on en déduit que le fonctionnement réel n'est pas normal. On produit alors un symptôme (appelé aussi résidu). Cette phase est appelée la phase de génération de symptôme ou phase de détection. Dans une deuxième phase, l'analyse diagnostique se fait alors en comparant le résultat de génération des résidus aux vecteurs binaires des différents défauts dont on possède le modèle de mauvais fonctionnement. L'isolation du défaut sur le système se produit lorsque la signature des symptômes est la même que l'une des signatures de défaut. Cette phase est appelée phase de localisation des défauts.

Pour arriver à la phase de localisation, le diagnostic par redondance analytique a besoin de construire des tests de détection qui reposent sur des modèles testables, appelés relations de redondance analytique (RRA). Ces derniers sont des modèles ne faisant intervenir que des variables mesurées ou connues à l'avance. Ces modèles doivent permettre de calculer des résidus. Une RRA est constituée d'un ensemble de modèles de composants en éliminant les variables inconnues et son évaluation se fait à partir des observations (OBS). Une relation de redondance peut être utilisée pour tester les composants physiques qui constituent son support. Le but des relations de redondance analytique est de permettre de construire des tests de cohérence afin de générer des résidus qui, par filtrage et seuillage, permettront de décider si le fonctionnement s'apparente ou non à un fonctionnement normal.

En effectuant des combinaisons judicieuses des relations ou contraintes du modèle, des RRA de supports différents (sensibles à des défauts particuliers) peuvent être générées. Différentes techniques ont été mises en œuvre pour générer des RRA :

- Ould Bouamama et al [Ould Bouamama et al., 2001] génèrent les relations de redondances analytiques (RRA) via les graphs de liaison. Une application à un générateur de vapeur est proposée.
- [Krysander et al., 2002] ont développé une méthode de génération de RRA et l'ont appliquée sur un procédé complexe et non linéaire de fabrication de papier.
- Evsukoff [Evsukoff, 1998] extrait des RRA spécifiques d'un modèle causal et l'applique à un atelier de colonnes pulsées.

Les résidus sont conçus pour faciliter leur exploitation ultérieure par un outil de décision destiné à détecter et à localiser les défauts. Deux approches sont possibles, à savoir :

- Génération de résidus directionnels [Gertler J., 1991] : les résidus sont conçus de telle sorte que le vecteur des résidus reste confiné dans une direction particulière de l'espace des résidus, en réponse à un défaut particulier. Cette technique a été introduite dans les années 1970 [Frank,

1996]. L'approche dite des résidus directionnels est en fait une spécialisation des résidus structurés dans laquelle chaque défaillance est représentée par un vecteur dans l'espace des symptômes.

- Génération de résidus structurés [Gertler et al ; 1993] : Les résidus sont conçus de façon à répondre à des sous-ensembles de défauts différents. Ces sous-ensembles de défauts permettent de structurer une table de signature appelée également matrice de signatures théoriques de défauts. Ces signatures traduisent l'influence des défauts sur les résidus. Nous allons développer cette approche plus en détails dans le paragraphe suivant.

II.3.2. Analyse par table de signatures

Un autre concept défini dans l'approche FDI est celui des signatures des défauts. La signature théorique d'un défaut peut être envisagée comme la trace attendue du défaut sur les différents RRA qui modélisent le système [Paton et Chen, 1991][Gertler, 1991][Frank, 1996] [Cordier et al., 2000]. Autrement dit la signature théorique d'un défaut peut être envisagée comme les résultats de détection lorsque tous les tests sensibles au défaut réagissent.

Les tests de cohérence, en utilisant des techniques de détection pour les systèmes dynamiques (par exemple les observateurs d'état, les relations de parités, ou estimation paramétrique) effectués sur l'ensemble des relations de redondance analytique, fournissent un vecteur binaire. En effet, la plupart du temps, les valeurs des résidus sont, à chaque instant, comparées à des seuils. Les tests peuvent être réalisés en parallèle et chaque décision issue de ces tests conduit à une valeur booléenne (0 : la valeur du résidu est en dessous du seuil ; 1: la valeur du résidu a dépassé le seuil fixé). L'ensemble de ces valeurs booléennes forme un vecteur binaire appelé *signature de défaut*. Le vecteur binaire est comparé aux différentes signatures de panne ainsi qu'à la signature de fonctionnement normal (vecteur de composantes nulles). Cette comparaison conduit alors à une conclusion sur l'état du système : fonctionnement normal, défaillance identifiée ou finalement défaillance non identifiée.

La localisation qui suit la détection peut être effectuée à l'aide de la table de signature. Les colonnes de cette table (figure 2.5) sont représentatives des différents défauts et les lignes des différents résidus. Les ensembles de diagnostics dans l'approche FDI sont donnés en termes de défauts présents dans la table de signature. La génération des ensembles de diagnostic est basée sur une interprétation des colonnes de la table de signature et consiste à comparer la signature des observations avec celle des défauts. Cette comparaison est considérée à nouveau comme un problème de décision.

Pour que tous les défauts puissent être détectés, aucune colonne de la matrice des signatures théoriques de défauts ne doit être nulle, et pour que tous les défauts puissent être localisés, toutes les signatures théoriques doivent être distinctes sans l'hypothèse d'exonération. [Busson et al. 1998]. [Gertler et al., 1990] distinguent trois types de matrices d'incidence :

- Non localisante (une colonne est nulle ou au moins deux colonnes sont identiques)

- Faiblement localisante (les colonnes sont non nulles et distinctes deux à deux),

➥ Fortement localisante (en plus d'être faiblement localisante, aucune colonne ne peut être obtenue à partir d'une autre en remplaçant un '1' par un '0').

	f_1	f_2	f_3
r_1	1	1	1
r_2	1	1	1
r_3	1	0	0

Non localisante

	f_1	f_2	f_3
r_1	1	1	1
r_2	1	0	1
r_3	1	1	0

Faiblement localisante

	f_1	f_2	f_3
r_1	1	1	0
r_2	1	0	1
r_3	0	1	1

Fortement localisante

Figure 2.5 - Tables de signatures (matrice d'incidence)

Une table non localisante ne permet pas de distinguer certains défauts entre eux. Une table faiblement localisante permet de localiser les défauts uniques sous hypothèse d'exonération. Une table fortement localisante garantit que les différentes sensibilités des résidus par rapport aux défauts ne conduisent pas à un diagnostic erroné. L'utilisation des tables de signatures établies pour des défauts simples pose des problèmes pour les situations de défauts multiples. Il faudrait en effet analyser toutes les combinaisons possibles des colonnes de la table.

II.4. APPLICATION AUX DEUX EXEMPLES

II.4.1. Système des deux bacs

Le modèle dynamique du système de bacs est non linéaire. Il peut être présenté sous forme d'équations d'état non linéaires $\dot{x} = f(x,u)$ données par le système d'équations suivant :

$$\begin{cases} \dfrac{d}{dt}\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} -k_1/S_1 & 0 \\ k_2/S_2 & -k_2/S_2 \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} 1/S_1 & 0 \\ 0 & 1/S_2 \end{bmatrix}\begin{bmatrix} q_{e1} \\ q_{e2} \end{bmatrix} \\ \begin{bmatrix} \tilde{h}_1 \\ \tilde{h}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \end{cases}$$

On peut en extraire trois représentations d'état linéaires. Ces représentations permettent de prendre en compte l'ensemble des informations disponibles puisqu'elles font intervenir les mesures de débit et de niveau d'eau ; elles sont données par :

$$M_1 = \begin{cases} \dot{\hat{x}} = A_0\hat{x} + B_0\begin{bmatrix} \tilde{q}_{e1} \\ \tilde{q}_{e2} \end{bmatrix} \\ \tilde{h}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{q}_{e1} \\ \tilde{q}_{e2} \end{bmatrix} \end{cases} \quad \text{avec } A_0 = \begin{bmatrix} -k_1/S_1 & 0 \\ k_2/S_2 & -k_2/S_2 \end{bmatrix} \text{ et } B_0 = \begin{bmatrix} 1/S_1 & 0 \\ 0 & 1/S_2 \end{bmatrix} \text{ et } C_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$M_2 = \begin{cases} \dot{\hat{h}}_2 = -k_2/S_2\,\hat{h}_2 + \begin{bmatrix} 1/S_2 & k_2/S_2 \end{bmatrix}\begin{bmatrix} \tilde{q}_{e1} \\ \tilde{h}_1 \end{bmatrix} \\ \tilde{h}_2 = \hat{h}_2 \end{cases}$$

$$M_3 = \begin{cases} \dot{\hat{h}}_1 = -k_1/S_1\,\hat{h}_1 + 1/S_1\,\tilde{q}_{e1} \\ \tilde{h}_1 = \hat{h}_1 \end{cases}$$

On en déduit les relations de redondance analytique suivantes à partir des observateurs de ces représentations d'état :

$$RRA_1 = \begin{cases} \dot{\hat{x}} = (A - K_1 C_1)\hat{x} + \begin{bmatrix} B & K_1 \end{bmatrix} \begin{bmatrix} \tilde{q}_{e1} \\ \tilde{q}_{e2} \\ \tilde{h}_2 \end{bmatrix} \\ r_1 = \tilde{h}_2 - \begin{bmatrix} 0 & 1 \end{bmatrix} \hat{x} \end{cases}$$

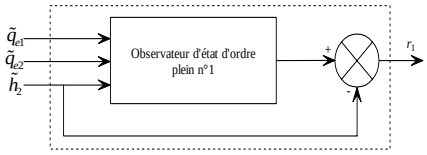
$$RRA_2 = \begin{cases} \dot{\hat{h}}_2 = \left(-k_2 / S_2 - K_2 \right) \hat{h}_2 + \begin{bmatrix} 1/S_2 & k_2/S_2 & K_2 \end{bmatrix} \begin{bmatrix} \tilde{q}_{e2} \\ \tilde{h}_1 \\ \tilde{h}_2 \end{bmatrix} \\ r_2 = \tilde{h}_2 - \hat{h}_2 \end{cases}$$

$$RRA_3 = \begin{cases} \dot{\hat{h}}_1 = \left(-k_1 / S_1 - K_3 \right) \hat{h}_1 + \begin{bmatrix} 1/S_1 & K_3 \end{bmatrix} \begin{bmatrix} \tilde{q}_{e1} \\ \tilde{h}_1 \end{bmatrix} \\ r_3 = \tilde{h}_1 - \hat{h}_1 \end{cases}$$

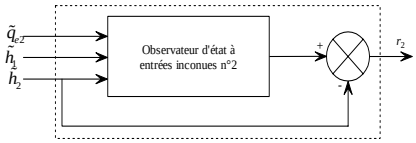
Où K_1, K_2 et K_3 sont les gains des observateurs d'états. Tous les gains sont réglés de telle sorte que les pôles des matrices d'état des observateurs soient nuls.

Des tests de détection sont effectués sur ces RRAs en utilisant des observateurs d'état [Luenberger, 1971]. Nous essayons, en général, de rendre l'observateur indépendant des perturbations non mesurées (entrées inconnues) et dépendant de certains défauts [Patton et Chen 1997]. C'est d'ailleurs le cas ici puisque certaines entrées ont été considérées comme inconnues.

- 1er observateur d'état indépendant de $\tilde{h}_1$ (RRA_1)



- 2ème observateur d'état indépendant de $\tilde{q}_{e1}$ (RRA_3)



- 3ème observateur d'état indépendant de $\tilde{h}_2$ et $\tilde{q}_{e2}$ (RRA_2)

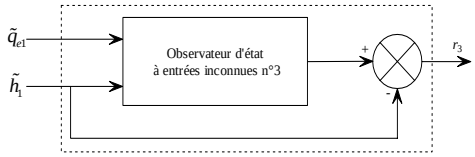


Figure 2.7 - Tests de détection avec Observateur d'état

La figure 2.7 représente les trois résidus générés par les trois observateurs d'état. Le résidu r_1 donne la liaison entre les entrées (q_{e1}) et (q_{e2}) et le capteur de hauteur de produit dans le bac du bas (h_2). Le résidu r_2 donne la liaison entre le flux d'entrée (q_{e1}), le capteur de hauteur du produit dans

le bac du haut (h_1) et le capteur de hauteur du produit dans le bac du bas (h_2). Le résidu r_3 donne la liaison entre le flux d'entrée (q_{e1}) et le capteur de hauteur dans le bac du haut (h_1).

Soit l'ensemble des signatures théoriques des défauts sur les composants du système des deux bacs à savoir les actionneurs (q_{e1}, q_{e2}) et les capteurs qui nous permettent de mesurer la hauteur du produit dans les deux bacs (h_1, h_2) respectivement ainsi que les restrictions (k_1, k_2). Dans le tableau de la (figure 2.8), la valeur 1 apparaît dans la case (i, j) lorsque le test basé sur l'observateur *i* est sensible à un défaut de capteur ou d'actionneur *j*. Cette table de signatures est de type non localisante car au moins deux colonnes sont identiques : q_{e2} , h_2 et k_2.

Tests	Actionneur(q_{e1})	Actionneur(q_{e2})	Capteur(h_1)	Capteur (h_2)	Restriction (k_1)	Restriction (k_2)
Test 1	1	1	0	1	1	1
Test 2	0	1	1	1	1	1
Test 3	1	0	1	0	1	0

Figure 2.8 - Table de signature

A partir des résidus obtenus, les aptitudes à la détection et à la localisation des défaillances vont être étudiées à travers des scénarios de pannes sur le système de bac. Les limites du raisonnement diagnostic à base de table de signatures sont montrées et expliquées.

II.4.1.1. Scénarios de pannes

La simulation du système de bacs permet d'étudier un certain nombre de scénarios de pannes. La simulation de défauts a été faite sur des bruits de mesures et de structure (comme la restriction bac haut). Avant de parler de défauts, il serait bon d'expliquer dans quelles conditions ont été réalisées les simulations : bruits de mesure et de structure. Un bruit additif de ±0.5cm a été simulé sur h_1 et h_2.

$$\begin{cases} \tilde{h}_1 = k_1 + \nu_1 \\ \tilde{h}_2 = k_2 + \nu_2 \end{cases}$$

Où ν_1 et ν_2 sont des variables aléatoires équiréparties dans [-0.5 0.5].

Une incertitude de modélisation a aussi été simulée sur les restrictions des deux bacs. Cette dernière est de type multiplicatif et correspond à une incertitude de ±50% de la valeur du paramètre de restriction :

$$\begin{cases} q_{s1} = (1+\theta_1) k_1 h_1 \\ q_{s2} = (1+\theta_2) k_2 h_2 \end{cases}$$

Où θ_1 et θ_2 sont des variables aléatoires équiréparties dans [-0.5 0.5].

La simulation du bon fonctionnement est donnée par la figure 2.9. Les courbes q_{e1} et q_{e2} représentent les débits d'entrées. Le débit q_{e1} bascule de 0 à 5 litres/h et inversement toutes les 500s tandis que le débit q_{e2} bascule de la même manière toutes les 250s.

Les seuils d'alarmes ont été fixés à ±15% des valeurs maximales des résidus obtenus lors de simulations caractérisant le bon comportement.

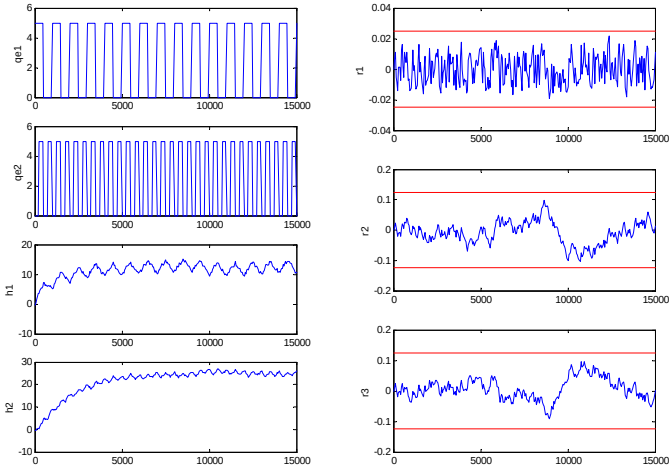


Figure 2.9- Le bon fonctionnement

Il est possible de simuler des défauts sur tous les capteurs et actionneurs du système. En fonctionnement idéal, les résidus doivent être constamment nuls. Pour notre cas d'étude les résidus ne sont pas nuls du fait des bruits. Les figures 2.10 à 2.14 présentent l'évolution des trois résidus r_1, r_2 et r_3 sur une durée de 15000s correspondant aux trois équations de redondance. Les défaillances sur les capteurs prendront la forme de biais.

Pour le premier scénario (cf. figure 2.10), un biais sur le capteur de hauteur de produit dans le bac h_2 est simulé. En effet, sa valeur a été augmentée de 5cm. La plupart du temps, les valeurs des résidus sont, à chaque instant, comparées à des seuils. Les tests peuvent être réalisés en parallèle et chaque décision issue de ces tests conduit à une valeur booléenne (0 : la valeur du résidu est en dessous du seuil ; 1: la valeur du résidu a dépassé le seuil fixé). Le résultat des tests de détection pour le premier scénario est donné par le vecteur des signatures de défaut suivant $S_D = [1,1,0]^T$, où le 1 représente le fait que les résidus r_1 et r_2 ont dépassés les seuils d'alarmes. En revanche, 0 représente le fait que r_3 reste en dessous du seuil d'alarme. Dans le premier cas le test conclu à une détection et dans le deuxième cas le test conclut à une consistance (pas d'alarme).

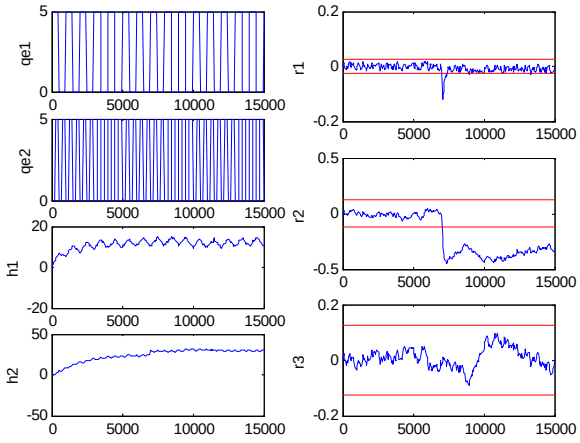


Figure 2.10- 1er scénario (biais sur le capteur h_2 de 5cm), $(S_D) = [1,1,0]^T$

La détection apparaît à l'instant t =7000s sur les résidus r_1 et r_2 et le symptôme reste présent jusqu'à la fin de la simulation t =15000s, en revanche le résidu r_3 reste insensible au défaut et la valeur du test de détection égale à 0 car la valeur du résidu est en dessous du seuil fixé. La comparaison de la signature des observations, c'est-à-dire le vecteur signatures de défaut $S_D=[1,1,0]^T$, avec celle des défauts possibles dans la table de signature de la figure 2.8 met clairement en évidence le défaut sur le capteur h_2. En revanche, plusieurs diagnostics sont possibles. En plus du défaut sur le capteur h_2, un biais sur l'actionneur q_{e2} et un défaut au niveau de la restriction (k_1) sont aussi possibles.

Le deuxième scénario (cf. figure 2.11) simule un biais d'actionneur (q_{e1}) de 0.2 litres/seconde. Le résultat des tests de détection est donné par la signature de défaut suivant $S_D =[0,0,1]^T$ (cf. figure 2.11). La détection apparaît à l'instant t= 12000s sur le résidu r_3. Par contre, r_1 et r_2 restent insensibles au défaut. La comparaison de la signature de défaut $S_D =[0,0,1]^T$ avec celle des défauts possibles dans la table de signature de la figure 2.8 conduit à la conclusion qu'aucun défaut présent dans la table de signature n'explique la situation.

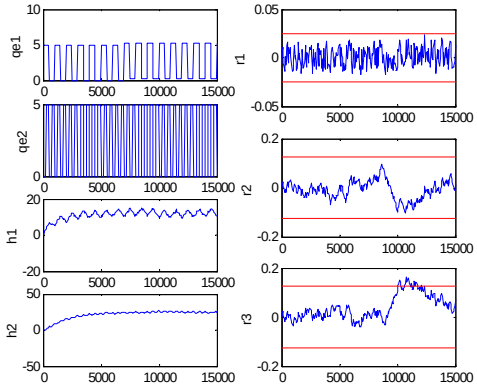


Figure 2.11 - 2ème scénario (Biais d'actionneur q_{e1} de plus 0.2 l/s), $(S_D)= [0,0,1]^T$

Cela illustre le fait qu'un défaut n'implique pas forcément la présence d'un symptôme. En effet, il existe des symptômes qui ne correspondent à aucune signature de défaut connue a priori. Dans ce cas, cette approche de localisation conduit à des résultats erronés.

Le scénario 3 (cf. figure 2.12) illustre une autre limitation du raisonnement par table de signature. Considérons un biais anormal de (20%) sur la restriction du bac du haut. Les tests de détection conduisent à la signature de défaut suivante $S_D =[0,1,1]^T$. Le test t_1 ne détecte pas le défaut. Par contre, le test t_2 et le test t_3 détectent un défaut.

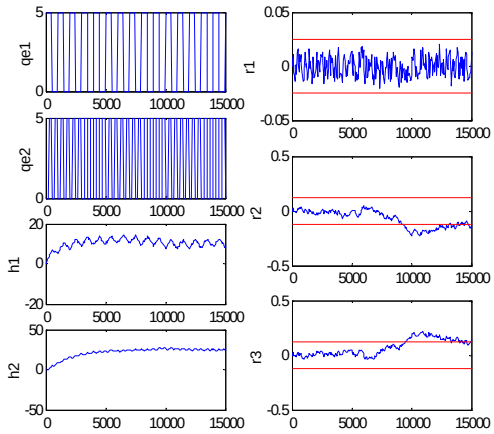


Figure 2.12: 3ème scénario (Biais (20%) sur la restriction du bac du haut), $(S_D)= [0,1,1]^T$

Après comparaison avec les signatures de défauts dans la table de signature de la figure 2.8 un défaut sur le capteur h_1 est diagnostiqué. Or, le défaut n'était pas un biais de capteur mais une modification du modèle du système. La localisation correcte du composant défaillant était impossible dans ce cas car tous les défauts possibles sur notre système ne sont pas considérés dans la table de signatures. Avec cette approche une bonne localisation présuppose que tous les défauts doivent figurer dans la table de signature afin de pouvoir les diagnostiquer. Or, dans le cas des systèmes complexes cette approche est difficile à envisager.

Le scénario 4 (cf. figure 2.13) correspond à un défaut double. Par conséquent nous essayons de montrer la limite de l'utilisation des tables de signatures établies pour des défauts simples. Un biais de capteur et un biais d'actionneur ont été simulés. Le premier est un biais de capteur de (2cm) sur la valeur réelle de h_2. Le second, est un biais d'actionneur de (0.2 litres/s) sur le débit d'entrée du produit dans le bac du haut (q_{e1}). Le vecteur de signature $S_D= [0,1,1]^T$ correspond à la signature d'un défaut sur le capteur h_2 dans la table de signatures (figure 2.8). Le fait d'avoir établi à l'avance une table des signatures des défauts simples pose des problèmes dans le cas de défauts multiples. Il faudrait en effet envisager toutes les combinaisons possibles des colonnes de la table.

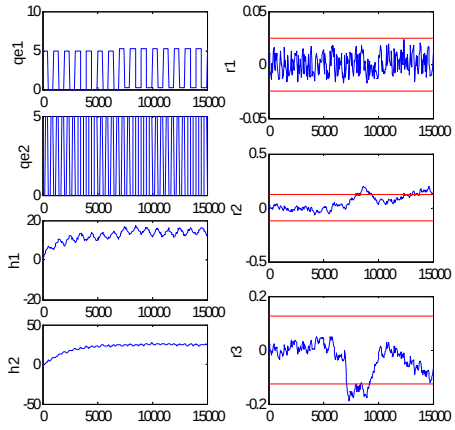


Figure 2.13 - 4ème scénario –défaut multiple (Biais sur h_2 (2 cm) et sur q_{e1} (0.2 l/s)), $(S_D)= [0,1,1]^T$

Voyons à présent, à travers le scénario 5 (figure 2.14), comment les tests de détection présentés ci–dessous ont permis la détection d'un défaut imaginaire, qui n'a jamais été simulé sur le

système de bac. En effet, le vecteur de signature généré S_D=[1,0,1]^T, correspond à la signature théorique du défaut q_{e1} (cf. figure 2.8). Que s'est-il passé ?

La validité des modèles qui interviennent dans la construction des relations de redondance n'est pas prise en compte. En pratique, les deux bacs ont une hauteur de 50cm, les capteurs de hauteur de produit dans les deux bacs doivent avoir des valeurs comprises dans l'intervalle [0, 50cm] afin que le modèle utilisé soit valide. Or, le domaine de validité d'une relation de redondance analytique est important car le test de détection en hérite : un résultat n'est valide que si l'état du système étudié appartient au domaine de validité qu'il est crucial de formaliser.

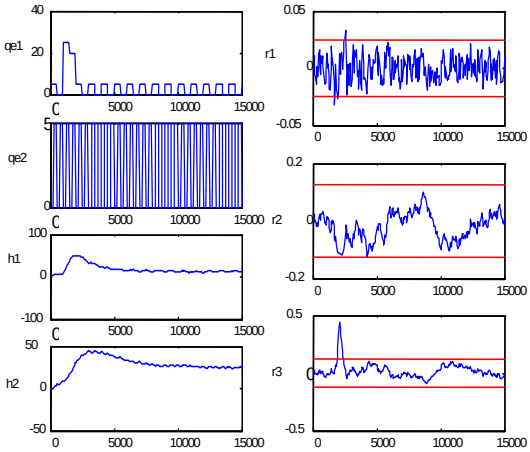


Figure 2.14 - 5ème scénario (aucun défaut), (S_D)= [1,0,1] ^{T}

La méthode de diagnostic à base de table de signatures est fondée sur une identification a priori des défauts et/ou dysfonctionnements pouvant survenir dans le système à diagnostiquer. Cette méthode présente des limitations importantes. Le recensement préalable des défauts ou des dysfonctionnements est difficilement exhaustif et il est par conséquent particulièrement mal approprié aux systèmes complexes. Ces méthodes sont en particulier mal adaptées aux systèmes techniques de conception nouvelle sur lequel aucune expérience n'existe encore.

Même si la détection ne produit jamais de fausses alarmes, l'analyse diagnostic peut néanmoins conduire à des erreurs de diagnostic car elle fait implicitement l'hypothèse qu'un défaut est détecté par tous les tests qui y sont sensibles et à tout moment. De plus, pour être complète, cette approche requiert la modélisation de tous les défauts possibles simples et multiples ; or, dans le cadre des systèmes complexes, ce recensement n'est guère envisageable.

Philippe Dague [Dague, 2001] rajoute que les connaissances sur lesquelles reposent ces méthodes sont dépendantes du système, si bien que toute modification ou évolution de ce système nécessite une réécriture plus au moins complète du système de diagnostic. Leur pouvoir explicatif est strictement limité à la donnée même des associations entre défauts et symptômes. Enfin, elles traitent mal les défauts multiples.

II.4.2. Circuit logique « Additionneur »

Le modèle du système (MP) donné par la figure 2.4, met à notre disposition dix relations reliant huit variables (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8). Certaines de ces variables sont connues à savoir (x_1,

x_2, x_3, x_7, x_8) données par le modèle d'observation. Chaque relation dans le modèle d'observation exprime le lien qui existe entre une variable inconnue et sa valeur disponible.

On a vu plus haut qu'un problème de diagnostic est défini par un modèle du système (MP), un ensemble d'observation (OBS) donnant les valeurs aux variables observées et un ensemble de défauts (DF) qui peuvent se produire sur les composants du système.

Soit OBS_1={ $\tilde{x}_1 = 1, \tilde{x}_2 = 0, \tilde{x}_3 = 0, \tilde{x}_7 = 0, \tilde{x}_8 = 1$ } l'ensemble des informations disponibles sur le système, issues des valeurs mesurées par les capteurs. Le fait qu'on a besoin d'envisager a priori les défauts susceptibles d'affecter l'additionneur pour pouvoir construire la table de signatures afin de mener à bien le diagnostic réduit considérablement le raisonnement diagnostique de l'approche par redondance analytique. En effet, pour pouvoir appliquer l'analyse par table de signatures ou l'analyse par génération de résidus directionnels sur l'additionneur, il faut que le défaut caractérisé par le vecteur de signature (S_D) soit recensé parmi une liste finie prédéterminée de signatures de défauts (DF). On ne considère ici que les pannes uniques, données par le tableau suivant :

	Xor1	*Xor2*	*And1*	*And2*	*Or*
RRA 1	1	1	0	0	0
RRA 2	1	0	1	1	1
RRA 3	0	1	1	1	1

Figure 2.15 - Table de signatures pour l'additionneur

Trois relations de redondance analytique ($RRA_{i=1,..,3}$) sont mises en évidence par élimination des variables inconnues. En effet, nous avons choisi d'effectuer ces trois tests tirés d'une combinaison linéaire des relations données par le tableau suivant :

RRA_1	RRA_2	RRA_3
$\begin{cases} M(XOR_1) \Rightarrow (x_4 = [x_1 \oplus x_2]) \\ M(XOR_2) \Rightarrow (x_7 = [x_4 \oplus x_3]) \\ M(x_1) \Rightarrow (\tilde{x}_1 = [x_1]) \\ M(x_2) \Rightarrow (\tilde{x}_2 = [x_2]) \\ M(x_3) \Rightarrow (\tilde{x}_3 = [x_3]) \\ M(x_7) : (\tilde{x}_7 = [x_7]) \end{cases}$	$\begin{cases} M(XOR_1) \Rightarrow (x_4 = [x_1 \oplus x_2]) \\ M(AND_1) \Rightarrow (x_6 = [x_2 \wedge x_4]) \\ M(AND_2) \Rightarrow (x_5 = [x_2 \wedge x_1]) \\ M(OR) \Rightarrow (x_8 = [x_5 \vee x_6]) \\ M(x_1) \Rightarrow (\tilde{x}_1 = [x_1]) \\ M(x_2) \Rightarrow (\tilde{x}_2 = [x_2]) \\ M(x_3) \Rightarrow (\tilde{x}_3 = [x_3]) \\ M(x_8) : (\tilde{x}_8 = [x_8]) \end{cases}$	$\begin{cases} M(XOR_2) \Rightarrow (x_7 = [x_4 \oplus x_3]) \\ M(AND_1) \Rightarrow (x_6 = [x_2 \wedge x_4]) \\ M(AND_2) \Rightarrow (x_5 = [x_2 \wedge x_1]) \\ M(OR) \Rightarrow (x_8 = [x_5 \vee x_6]) \\ M(x_1) \Rightarrow (\tilde{x}_1 = [x_1]) \\ M(x_2) \Rightarrow (\tilde{x}_2 = [x_2]) \\ M(x_3) \Rightarrow (\tilde{x}_3 = [x_3]) \\ M(x_8) : (\tilde{x}_8 = [x_8]) \end{cases}$

➥ Le résidu r_1 est le résultat du test de cohérence de la RRA_1, r_1 correspond à ce qui suit :

$$r_1 = \tilde{x}_7 - \left[(\tilde{x}_1 \oplus \tilde{x}_2) \oplus \tilde{x}_3 \right]$$

➥ Le résidu r_2 est le résultat du test de cohérence de la RRA_2, r_2 correspond à ce qui suit :

$$r_2 = \tilde{x}_8 - \left[(\tilde{x}_2 \wedge \tilde{x}_1) \vee (\tilde{x}_2 \wedge (\tilde{x}_1 \oplus \tilde{x}_2)) \right]$$

➥ Le résidu r_3 est le résultat du test de cohérence de la RRA_3, r_3 correspond à ce qui suit :

$$\begin{cases} (\tilde{x}_7 = [x_4 \oplus \tilde{x}_3]) \\ (r_3 = \tilde{x}_8 - [(\tilde{x}_2 \wedge \tilde{x}_1) \vee (\tilde{x}_2 \wedge x_4)]) \end{cases}$$

Les valeurs des trois résidus ci-dessus fournissent le vecteur de signature (S_D). Pour obtenir le résidu r_1 il suffit de comparer la valeur x_7 prédite par combinaison linéaire des modèles $M(XOR_1)$ et $M(XOR_2)$, connaissant les valeurs de x_1, x_2 et x_3 données par l'ensemble des observations OBS, et la valeur observée obtenue via le capteur de x_7=0. La différence entre ces deux valeurs fournit le résidu r_1=1. De même, pour le résidu r_2 = 1, sa valeur est la différence entre la valeur observée de x_8=1 et la valeur prédite par la combinaison linaire des modèles $M(XOR_1)$ et $M(AND_1)$, $M(AND_2)$ et M(OR). Avec le même raisonnement, on obtient le résidu r_3=1 qui manifeste la contradiction entre l'observation de x_8 et sa prédiction en utilisant les modèles $M(XOR_2)$, $M(AND_1)$, $M(AND_2)$ et M(OR).

On a vu précédemment qu'avec l'approche FDI, les diagnostics sont donnés en termes de défauts présents dans la matrice de signature. La génération des ensembles de diagnostic est basée sur l'interprétation des colonnes de la matrice de signature et consiste à comparer la signature du vecteur $(S_D)=[r_1, r_2, r_2]^T = [1,1,1]^T$ trouvée précédemment avec celle des défauts dans la table de signatures (cf. figure 2.15). Il en résulte l'ensemble de diagnostic vide « ϕ ». En effet, aucune signature de défaut de la table de signatures ne correspond à la signature de notre vecteur $(S_D)=[1,1,1]^T$.

Si on prend un autre jeu d'observation $OBS_2 = \{ \tilde{x}_1 = 1, \tilde{x}_2 = 0, \tilde{x}_3 = 0, \tilde{x}_7 = 1, \tilde{x}_8 = 1 \}$ et en appliquant le même raisonnement qu'auparavant alors le vecteur de signature qui en résulte est le suivant : $[0,1,1]^T$. Cette nouvelle signature correspond à trois signatures de défaut dans la table de signature à savoir AND_1, AND_2 et OR. Par conséquent, le résultat de diagnostic est donné en terme de composants défaillants {AND_1, AND_2 , OR } pour lesquels une signature de défaut est présente dans la table de signature.

II.5. L'APPROCHE DU DIAGNOSTIC LOGIQUE

La méthode, dite du diagnostic à base de modèles a vu le jour aux Etats-Unis au milieu des années soixante-dix et a été formalisée au début des années quatre-vingt. Un nombre croissant de travaux ont été menés depuis et cette problématique est devenue un domaine de recherche à part entière de l'intelligence artificielle. Les articles les plus marquants sur ce domaine et publiés avant 1991 sont regroupés dans [Hamscher et al., 1992].

Fonder le diagnostic sur la notion de cohérence n'est pas nouveau ; De Kleer et Williams présentent un parallèle entre la construction d'un modèle et le diagnostic, mais uniquement dans le cadre d'une approche orienté « bon fonctionnement » [De Kleer, 1987]. Pour Poole, un problème de diagnostic consiste à utiliser un modèle pour chercher des explications satisfaisantes aux symptômes observés [Poole, 1989].

Les systèmes de diagnostic qui font appel à cette technique utilisent un raisonnement qualifié de « profond » (*deep reasonning*). En effet, le diagnostic à base de modèle s'appuie sur une description approfondie du système à diagnostiquer et sur une observation de son comportement : si le comportement observé diffère du comportement prévu (à partir de sa description), le problème du diagnostic consiste à déterminer quels composants du système (en les supposant défaillants) peuvent expliquer la différence entre les comportements observés et corrects (*discrepancy detection problem*) [Davis et al, 1992].

a) <u>La technique de Davis</u>

On suppose qu'on dispose d'un modèle structurel du système, à un niveau d'abstraction donné, et que le comportement correct de chaque module (un module est un composant fonctionnel ou une connexion) est connu spécifiquement par un ensemble de règles. Ces règles expriment les relations entre les entrées et les sorties d'un module lorsque son fonctionnement est supposé correct : pour cette raison on les appelle contraintes de bon fonctionnement. Davis [Davis, 1984] [Davis et al, 1992] définit des règles de simulation qui donnent les valeurs des sorties en fonction des entrées et des règles d'inférence qui donnent les valeurs des entrées en fonction des sorties. L'exemple étudié par Davis est un circuit logique constitué de deux additionneurs (A_1, A_2) et de trois multiplieurs (M_1,M_2,M_3) (cf. figure 2.16).

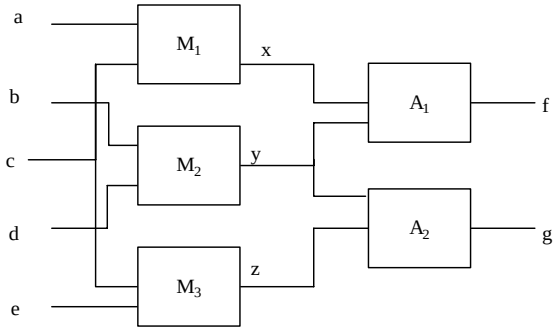


Figure 2.16 - Le système de Davis

- Règles de simulation : M_1 : a.c= x ; M_2 : b.d= y ; (M_3) : e.c=z ; (A_1) : x+y=f ; (A_2) : y+z=g.
- Règles d'inférence : x=f-y ; y=f-x règles associées à A_1 ; z=g-y ; y= g-z règles associées à A_2.

Dans le modèle abstrait du système, on propage les valeurs initiales des entrées vers les sorties à travers les modules : étape de propagation des contraintes. Chaque fois qu'on calcule la valeur d'une variable interne, on enregistre la règle appliquée. Si les valeurs mesurées (observées) et les valeurs calculées (inférées par l'application des règles) sont différentes, un symptôme est détecté. Pour ce symptôme, on met en cause les modules traversés pendant le calcul de la valeur de cette sortie (chemin de causalité). Cet ensemble de modules est appelé un ensemble de candidats potentiels. Davis propose une méthode appelée suppression de contraintes fondée sur l'hypothèse

de défaut unique. Cette méthode consiste à analyser chaque candidat potentiel en supposant que les autres modules fonctionnent correctement. Le processus d'analyse est résumé par les points suivants :

- Sélectionner un module et suspendre ses contraintes, c'est-à-dire ses règles de simulation et d'inférence.
- Relancer le processus de propagation des valeurs dans le réseau de contraintes en n'utilisant que les règles validées. Si le réseau atteint un état consistant, alors il faut retenir ce composant comme candidat potentiel, sinon il faut ignorer ce composant en validant de nouveau ses contraintes, et choisir un autre candidat.

Pour illustrer cette méthode, considérons l'exemple de la figure 2.16 où les valeurs des entrées sont a=3, b=2, c=2, d=3, e=3 et les valeurs mesurées des sorties sont f=10, g=12. Après l'application des règles de simulation, les valeurs calculées seront x=6, y=6, z=6, f=12, g=12. Or, un conflit est détecté en f : La valeur observée est 10 et celle inférée est 12. Les candidats potentiels sont alors M_1, M_2 et A_1.

L'analyse du candidat A_1 consiste à :

- Invalider les règles $x+y=f(A_1)$, x= f-y et y=f-x ; garder la valeur de f=10
- Recommencer l'inférence des autres valeurs dans le réseau de contraintes.

Dans ce cas, aucun conflit n'est détecté et A_1 est considéré comme le candidat pour la faute. Ensuite, pour analyser M_1 on invalide la règle x=a.c et on garde la valeur de sa sortie x=6. L'application des deux règles y=f-x=10-6 =4 et y=g-z=12-6=6, montre un autre conflit en y, alors le module M_1 n'est plus considéré comme un candidat potentiel. Le même raisonnement sera appliqué à M_2 pour montrer que ce module n'est pas candidat potentiel.

b) La technique de De Kleer

De Kleer utilise le même modèle que Davis dans son système de diagnostic GDE (*General Diagnostic Engine*) [De Kleer et Williams, 1987]. Contrairement à Davis qui travaille avec l'hypothèse de panne unique, De Kleer généralise la méthode pour les pannes multiples. Il considère qu'un défaut candidat est un ensemble minimal de composants qui vérifient la consistance avec tous les conflits détectés (De Kleer définit un conflit comme étant l'ensemble des composants traversés pendant le calcul de la valeur d'une sortie erronée). Pour un système de N composants, le nombre de composants à analyser dans l'espace de recherche des candidats de fautes est de 2^N. Cet espace est représenté par un treillis des combinaisons possibles. Si on considère l'exemple de la figure (2.16), on a 32 candidats (cf. figure 2.17).

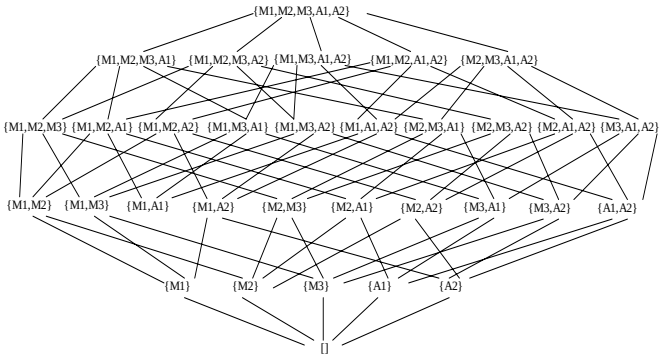


Figure 2.17 - Treillis initial des défauts candidats pour l'exemple de la figure 2.16

On définit un défaut candidat comme un sous-ensemble minimal de modules défectueux. Chaque candidat doit être consistant avec tous les conflits détectés. La génération des candidats est faite de façon itérative. A chaque pas, on considère un ensemble de conflits ; on commence par le premier niveau du treillis et on sélectionne les ensembles (candidats) qui ont une intersection non vide avec l'ensemble des conflits choisis. Si un candidat ne vérifie pas cette condition, il est remplacé (étape d'affinement) par des ensembles de niveau supérieur qui vérifient les deux conditions :

- Ils ne contiennent pas de candidats générés (condition de minimalité) ;
- Ils ont une intersection non vide avec tous les conflits considérés.

Pour illustrer cette méthode, reprenons l'exemple de la figure 2.16. Au début, tous les composants sont supposés corrects et l'ensemble des candidats minimum est vide. A partir d'un conflit minimum détecté en « f », les candidats minimums sont (A1, M1, M2).

On génère les ensembles minimaux qui contiennent l'ensemble vide : $[M_1][M_2][M_3][A_1][A_2]$ (premier niveau du treillis). Les candidats minimums n'expliquant pas ce conflit ($[M_3]$, $[A_2]$) sont remplacés par de nouveaux candidats minimums, augmentés d'un ou plusieurs composants du nouveau conflit (c'est-à-dire, par le seul ensemble minimal $[M_3, A_2]$ au deuxième niveau.

Comme les autres contiennent des candidats déjà construits, cet ensemble est éliminé. Lorsqu'on détecte l'autre conflit (M_3, A_2, M_1, A_1) en « g », les candidats construits au pas précédent sont de nouveau analysés :

- $[M_1]$, $[A_1]$ sont gardés,
- $[M_2]$ est affiné par les deux ensembles $[M_2, A_2]$ et $[M_2, M_3]$ en suivant le raisonnement précédent.

On obtient finalement 4 candidats $[M_1]$, $[A_1]$, $[M_2, A_2]$, $[M_2,M_3]$. Si on considère le cas de panne unique, on a deux candidats $[M_1]$ ou $[A_1]$.

c) La technique de Reiter

Reiter s'inspire des travaux de Davis et formalise une théorie logique de diagnostic [Reiter, 87]. Le problème du diagnostic peut s'énoncer de la façon suivante : supposons connue une description d'un système ainsi qu'un ensemble d'observations sur son comportement. Le problème

du diagnostic consiste à déterminer les composants dont le dysfonctionnement expliquerait les différences observées.

Ce qui caractérise cette approche de diagnostic est qu'il n'est nul besoin de recenser a priori les défauts ou défaillances pouvant affecter un système pour pouvoir le diagnostiquer. Modéliser le comportement correct est suffisant. L'idée fondamentale est de comparer le comportement réel du système tel qu'il peut être observé par l'intermédiaire de capteurs avec son comportement attendu tel qu'il peut être prédit grâce aux modèles de comportement. Si ces modèles sont justes, c'est-à-dire qu'ils constituent bien et caractérisent bien tous les comportements associés à un état donné, tout écart entre les observations et les prédictions déduites des modèles est nécessairement la manifestation d'un dysfonctionnement, c'est-à-dire qu'il révèle la présence d'un ou plusieurs défauts. Ce type de raisonnement par l'absurde, où un défaut est par définition n'importe quoi d'autre qu'un comportement normal attendu et, de ce fait, ne requiert pas de recenser tous les défauts spécifiques, s'avère être à la fois bien adapté aux systèmes complexes où le nombre de défauts possibles est très important, et rigoureux parce qu'il est fondé sur un raisonnement logique.

Le système physique à diagnostiquer sera décrit par un ensemble de modèles. La description du comportement du système est orientée composants. Les connaissances incluses dans ces modèles décrivent la structure du système à diagnostiquer et son comportement. Le modèle structurel décrit généralement les liens, les connexions entre les différents composants du système à diagnostiquer. Quant au modèle comportemental, il est généralement le résultat de la composition des modèles de comportement de chaque composant du système. Le cadre théorique logique permettant de formuler rigoureusement le problème du diagnostic à base de modèles a été établi dans [Reiter, 1987] [De Kleer et al., 1992].

Un système est alors défini par un couple (SD, COMPONENTS) où SD, le système de description, est un ensemble d'énoncés du premier ordre, et COMPONENTS, les composants du système, est un ensemble fini. Un système et son observation sont notés sous la forme d'un triplet : (SD, COMPONENTS, OBS). Reiter introduit alors les notations et propositions suivantes pour l'opération de diagnostic.

Supposer qu'un système (SD, $\{c_1,\ldots c_n\}$) est en panne, signifie que l'observation est en conflit avec l'état attendu du système si chacun des composants « c_i » était en état de bon fonctionnement. La notation $\{\neg AB(c_1),\ldots,\neg AB(c_n)\}$ représente l'hypothèse que tous les composants du système fonctionnent normalement (AB signifie anormal et $\neg$ est la négation). Le fait que les observations OBS soient en conflit avec ce que le système devrait logiquement faire si chaque composant fonctionnait correctement s'exprimera comme suit :

$SD \cup \{\neg AB(C_1),\ldots,\neg AB(C_n)\} \cup OBS$ est incohérent.

Un diagnostic pour (SD, COMPONENTS, OBS) est un ensemble minimal $\Delta \subseteq$ COMPONENTS tel que :

$SD \cup OBS \cup \{AB(C_i) \mid C_i \in \Delta\} \cup \{\neg AB(C_k) \mid C_k \in COMPONENTS - \Delta\}$ est cohérent (A)

Cette définition est en accord avec le principe de parcimonie, un diagnostic est une conjoncture dans laquelle un ensemble minimal de composants défectueux explique la situation observée.

Reiter propose ensuite un ensemble conflictuel pour (SD ∪ COMPONENTS ∪ OBS) est minimal si et seulement si aucun de ses sous-ensembles n'est un ensemble conflictuel pour (SD ∪ COMPONENTS ∪ OBS) ; ceci conduit à une seconde définition équivalente de (A) : Δ⊆ COMPONENTS est un diagnostic pour SD ∪ COMPONENTS ∪ OBS si est seulement si Δ est un ensemble minimal tel que COMPONENTS-Δ n'est pas un ensemble conflictuel pour (SD ∪ COMPONENTS ∪ OBS.)

A partir de ces définitions et d'une base de théorèmes et de propositions, *Reiter* a formulé le problème du diagnostic à partir des principes premiers sous la forme d'un problème de consistance de formules logiques de premier ordre. Il propose ainsi un algorithme *Diagnosis* pour déduire le diagnostic des circuits digitaux non complexes et non dynamiques. Cet algorithme est basé sur l'utilisation d'une fonction TP qui génère un ensemble de conflits. Les ensembles de conflits sont ensuite utilisés pour construire des arbres appelés H-S (hitting set). Dans ces arbres, chaque chemin allant de la racine vers une feuille marquée par « √ »est un diagnostic. Si la fonction TP produit un ensemble de conflits minimal les arbres H-S seront optimisés, c'est-à-dire qu'il y aura moins de redondances. Nous allons illustrer un peu plus loin dans nos exemples l'application de l'algorithme basé sur les « *hitting sets* ».

Des extensions de formalisme ont été proposés pour permettre d'intégrer et d'exprimer des modèles de dysfonctionnement [Struss et al, 1989] [Struss et Dressler, 2001] et Sherlok [De Kleer et Williams, 1989]. Pour un composant donné plusieurs modes de fonctionnement peuvent être définis. En effet, au lieu de n'avoir que deux modes fonctionnement par composant (normal et anormal), dont seul le premier mode est modélisé, s'ajouteront cette fois au mode correct plusieurs modes de dysfonctionnement (en général deux à deux exclusifs) [Dague, 2001] modélisés. Le mode Inconnu dépourvu de tout modèle et censé regrouper tous les comportements défectueux non répertoriés. Si l'on appelle B (pour Bon) le mode correct, F_i (pour fautif) les modes défectueux et I le mode Inconnu, un composant C aura ainsi pour modes :

{B (C), F_1(C),…,F_k(C), I(C) }

Un conflit devient dans ce cadre une attribution de modes comportementaux à certains composants qui est en contradiction avec les observations, c'est à dire que les prédictions que l'on peut tirer des modèles de ces modes sont incohérents par rapport aux observations. Un diagnostic est alors une assignation de modes comportementaux à tous les composants du système qui rétablit la cohérence avec les observations. Dans ce cadre, les diagnostics ne sont plus caractérisés comme les surensembles des conflits minimaux, bien qu'ils conservent une caractérisation logique en termes de conflits [De Kleer et al, 1992] correspondant à un problème NP-difficile [Dague, 2001].

II.6. APPLICATION SUR LES DEUX EXEMPLES

II.6.1. Système des deux bacs

On ne peut pas appliquer le raisonnement logique de Reiter sur un système dynamique décrit par des équations différentielles tel que le système des deux bacs. En effet, il est mentionné dans [Struss, 1991] que la théorie logique à base de modèle est loin d'être facile à appliquer sur un système dynamique car ce qui pose le plus problème est la construction de DS (description du système). De plus, pour pouvoir effectuer des tests de détection (qui consistent à comparer le comportement réel d'un système observé grâce aux informations disponibles au niveau des capteurs et des actionneurs avec son comportement attendu tel qu'il peut être prédit grâce aux modèles de comportements dynamiques de notre système), il nous faudrait disposer d'un solveur général produisant des formules impliquées ou, comme c'est le cas en pratique, par un simulateur de modèles couplé à un ATMS (*Assumption Based Truth Maintenance System*) de De Kleer [Friedrich et al., 1990] ou à une de ses variantes. La théorie du diagnostic elle-même ne fait que déterminer la cohérence entre le système réel et le système modélisé. Un conflit n'affirme rien d'autre qu'au moins un de ses composants ne fonctionne pas suivant son modèle.

Le raisonnement qualitatif est un recours pour palier aux problèmes de modélisation des systèmes dynamiques, en introduisant la modélisation qualitative. L'intérêt de l'approche qualitative est qu'elle demande une connaissance faible sur les relations liant les différentes variables d'un système, puisqu'il ne faut pas les quantifier. Elle permet donc de prendre en compte des relations entre variables, que l'on est incapable de décrire quantitativement, mais qui peuvent cependant s'avérer suffisantes pour confirmer ou réfuter une hypothèse dans un raisonnement de diagnostic (uniquement le signe des influences, par exemple). Sur beaucoup de systèmes physiques trop difficiles à modéliser numériquement, cette information existe, il n'y a pas de raison pour qu'elle soit dédaignée parce qu'elle n'a pas la précision du quantitatif. Dans certaines approches, on utilise le fait qu'un modèle qualitatif est plus rapidement élaboré qu'un modèle numérique pour modéliser de façon exhaustive les pannes d'un système ; le diagnostic devient alors un problème de reconnaissance du bon modèle de panne parmi plusieurs.

Par ailleurs, un ensemble de travaux [Dressler, 1994] [Dressler et al., 1994][Dressler, 1995][Malik et al., 1996][Struss et al., 1997] ont montré expérimentalement qu'il est parfois possible d'effectuer un diagnostic de systèmes dynamiques et de la supervision d'un système variable dans le temps (modélisé en général qualitativement) en tenant compte seulement des contraintes inter-états aux instants d'observation et en ignorant les contraintes inter-états qui portent sur l'évolution entre les instants d'observations, c'est à dire, en ignorant les problèmes temporels. Peter Struss a abordé cette problématique dans [Struss, 1997] dans lequel il développe des fondements théoriques pour l'application du diagnostic à base de consistance aux systèmes dynamiques. Il précise les conditions sous lesquelles on peut ainsi se limiter à un diagnostic à base d'états sans avoir besoin d'utiliser le diagnostic à base de simulation, qui revient excessivement cher. Ces conditions précisent, d'une part, que les contraintes inter-états se limitent à des contraintes mathématiques générales portant sur la continuité et la dérivabilité des lois d'évolution (pas de discontinuités par rapport à la dynamique du système). D'autre part, que les observations sur le système soient avec une fréquence suffisante pour garantir qu'aucun état dans l'évolution du

comportement ne sera manqué. En plus, il faut que ces observations soient suffisamment complètes pour détecter tout comportement incohérent avec les contraintes inter-états du mode correct. [Panati et al., 1998] montre que ces conditions sont restrictives et non satisfaites lorsqu'un défaut soudain introduit une discontinuité dans l'évolution temporelle du système, la simulation est alors nécessaire. L'utilisation des connaissances sur la causalité du système, qui restreignent les états possibles après la survenue d'un défaut soudain, peut palier le problème et rendre la méthode proposée par Struss efficace.

II.6.2. Système de l'additionneur

Nous illustrons l'algorithme de *Reiter* par l'exemple de l'additionneur. Ce dernier sera décrit par (DS, COMPOSANTS, OBS) tel que :

COMPOSANTS = { A_1, A_2, X_1, X_2, O_1}

La description du système est donnée par l'ensemble des formules logiques qui décrivent les fonctionnements de bon comportement des différentes portes logiques (AND, OR, XOR) et les connexions entre les composants.

$$\begin{cases} M(XOR_1) \Rightarrow \left(x_4 = [x_1 \oplus x_2]\right), M(XOR_2) \Rightarrow \left(x_7 = [x_4 \oplus x_3]\right), M(AND_1) \Rightarrow \left(x_6 = [x_2 \wedge x_4]\right) \\ M(OR) \Rightarrow \left(x_8 = [x_5 \vee x_6]\right), M(AND_2) \Rightarrow \left(x_5 = [x_2 \wedge x_1]\right) \end{cases}$$

Les valeurs des entrées x_1, x_2, x_3 et les sorties x_7 et x_8 sont données par notre premier ensemble d'observations :

$$OBS_1 = \left\{\left(\tilde{x}_1 = 1\right), \left(\tilde{x}_2 = 0\right), \left(\tilde{x}_3 = 0\right), \left(\tilde{x}_7 = 0\right), \left(\tilde{x}_8 = 1\right)\right\}$$

Après l'application des entrées x_1, x_2 et x_3 sur les modèles de bon comportement des différentes portes logiques, on obtient des formules impliquées par DS$\cup$ {x_1, x_2, x_3}

$$\begin{cases} \neg AB(XOR_1) \Rightarrow \left(x_4 = 1\right) \\ \neg AB(XOR_1) \wedge \neg AB(XOR_2) \Rightarrow \left(x_7 = 1\right) \\ \neg AB(XOR_1) \wedge \neg AB(AND_2) \Rightarrow \left(x_6 = 0\right) \\ \neg AB(AND_1) \Rightarrow \left(x_5 = 0\right) \\ \neg AB(AND_1) \wedge \neg AB(XOR_1) \wedge \neg AB(AND_2) \wedge \neg AB(OR) \Rightarrow \left(x_8 = 0\right) \end{cases}$$

La valeur prédite de la sortie de XOR_2 par le modèle $M(XOR_2) \Rightarrow \left(x_7 = [x_4 \oplus x_3]\right)$ est égale à 1 et la valeur observée de x_7 est égale à 0 donnée par $\left(\tilde{x}_7 = 0\right)$. La comparaison indique clairement qu'il y a une divergence entre les deux valeurs, on en déduit que ce cas de figure présente un conflit qui est la conséquence de DS$\cup$OBS. On en déduit que la prémisse $\neg AB(XOR_1) \wedge \neg AB(XOR_2)$ ne peut plus être satisfaite avec DS$\cup$OBS. Par conséquent, l'un des deux portes XOR se comporte anormalement. Alors on peut écrire cette clause $AB(XOR_1) \vee AB(XOR_2)$ qui constitue un conflit positif minimal et l'ensemble *{XOR_1, XOR_2 }* constitue le R-conflit minimal correspondant. Par ailleurs, la valeur prédite de la sortie de OR par la modèle $M(OR) \Rightarrow \left(x_8 = [x_5 \vee x_6]\right)$ étant 0, la valeur observée de la sortie x_8 est égale à 1 $\left(\tilde{x}_8 = 1\right)$. Cette différence entre la valeur prédite par le modèle et la valeur observée va donner lieu à un conflit. Par conséquent, la prémisse

$\neg AB(AND_1) \wedge \neg AB(XOR_1) \wedge \neg AB(AND_2) \wedge \neg AB(OR) \Rightarrow (x_8 = 0)$ ne peut plus être satisfaite avec DS∪OBS. Il en résulte la clause suivante $AB(AND_1) \vee AB(XOR_1) \vee AB(AND_2) \vee AB(OR)$ impliquée directement par DS∪OBS. Cette dernière clause constitue un conflit positif minimal et l'ensemble *{AND₁, XOR₁, AND₂, OR}* constitue le R-conflit minimal correspondant.

Le R-conflit peut être expliqué comme suit : les observations ne peuvent être expliquées que si un au moins des composants du R-conflit est défectueux. En utilisant les R-conflits minimaux *{ XOR₁, XOR₂ },{AND₁, XOR₁, AND₂, OR}*, il est possible de donner une caractérisation des diagnostics minimaux, qui fournit une base pour les calculer [Reiter, 87]. L'algorithme *Diagnosis* génère 4 diagnostics {XOR1}, {XOR2, AND1}, {XOR2, AND2}, {XOR2, OR} donnés par l'arbre de la figure 2.18.

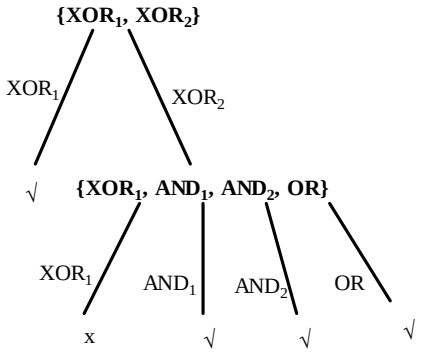


Figure 2.18 - L'arbre H-S de l'additionneur

Cette technique requiert une description détaillée du modèle comportemental. Les relations entre les entrées et les sorties de chaque composant doivent être exprimées par des équations logiques ou arithmétiques, afin de déduire les règles de propagation en amont et en aval des valeurs numériques ou logiques. De plus, les techniques utilisées pour générer les diagnostics (candidats aux défauts) sont souvent de complexité élevée (problème NP-difficile). Par conséquent, ce type de diagnostic est limité au diagnostic des circuits digitaux simples (additionneur, multiplieur,…).

II.7. ANALYSE COMPARATIVE ENTRE L'APPROCHE DX ET L'APPROCHE FDI

L'approche DX propose une théorie générale du diagnostic fondée sur les concepts de logique qui a son origine en IA. Elle se différencie des autres approches existantes, en particulier l'approche FDI, par plusieurs points (voir table 3.3). Nous proposons de discuter plus en détail cette approche par rapport à l'approche à base de modèles de l'automatique décrite par exemple dans [Dubuisson, 2001], chapitre 2 et 3. Comme leur nom identique, les principes sous-jacents sont comparables, notamment par l'utilisation d'un modèle représentant le système physique, et de nombreux concepts s'avèrent équivalents. Une étude comparative détaillée est faite dans [Cordier et al, 2004] ou dans [Nyberg, 2001] pour un cadre de diagnostic fondé sur les tests structurés d'hypothèses, développé dans une perspective de contrôle automatique mais inspiré aussi de l'IA.

La table 3.3 résume les principales différences entre l'approche FDI et l'approche DX.

	L'approche FDI	**L'approche DX**
Prise en charge de tests de détection	Sophistiqués	Relativement simples
Type de modèle	Comportement normal avec défaut additif ou multiplicatif	Comportement normal
Diagnostic formel	Non	Oui
Exonération	Implicite	Aucune hypothèse implicite
Lien composant	A posteriori	A priori
Procédure de diagnostic	Deux étapes séquentielles (Détection et localisation)	Deux étapes itératives

Table 3.3 - Bilan et comparaison avec l'approche DX

II.7.1. Prise en charge de test de détection

Les tests de détections peuvent être plus sophistiqués dans l'approche FDI que dans l'approche DX, En effet, FDI permet d'utiliser des outils mathématiques comme les observateurs d'état, les relations de parités ou l'estimation paramétrique. Les tests de cohérence qui s'appuient sur les observateurs d'état reposent sur une modélisation de type algébro-différentiel (déterministe ou stochastique pour les filtres de Kalman). Avec l'approche FDI, un sous-système peut être testé en utilisant différents méthodes mathématiques par exemple un observateur d'état [Luenberger, 1971] ou une relation de parité [Massoumnia, 1988].

Dans l'approche DX, les contradictions entre observations et prédictions sont recoupées entre elles afin de déterminer toutes les explications possibles, c'est-à-dire tous les composants qui peuvent être défaillants. Il suffit pour cela d'utiliser les prédictions préalablement enregistrées qui mènent aux contradictions en question : si une prédiction a été faite en utilisant les modèles de bon fonctionnement de composants $C_1,\ldots,C_n$, et qu'elle entre en contradiction avec une observation, c'est donc (raisonnement par l'absurde) que les composants $C_1,\ldots,C_n$ ne peuvent être tous corrects et que l'un d'eux est nécessairement défectueux (Reiter dit que ces composants forment un conflit).

II.7.2. Type de modèle

L'approche FDI utilise des modèles précis (numériques) pour la phase de détection et prend en compte explicitement les perturbations inconnues (perturbations additives), les erreurs de modélisations (perturbations multiplicatives) et les bruits dans la conception de résidus robustes généralement obtenus par des techniques d'observateurs d'état, de relations de parité ainsi que des techniques de traitement de signal.

En DX, l'existence de perturbations, d'erreur de modèle et de bruits est en général traitée de manière implicite par des modèles à des niveaux d'abstraction élevés (symboliques ou qualitatifs). Les modèles de tendances incluent des perturbations qui apparaîtraient à des niveaux de modélisation plus fins [Travé-Massuyès et al, 1997].

II.7.3. Diagnostic formel

Dans les deux approches, l'analyse diagnostique commence quand des inconsistances apparaissent entre le comportement modélisé du bon fonctionnement et les observations. Dans la structure des relations de redondance analytique, les inconsistances proviennent des relations qui ne sont pas satisfaites par les observations.

Dans l'approche DX, les divergences permettent l'identification de R-conflits, où un R-conflits est un ensemble de composants dont le bon comportement implique une inconsistance d'un test de détection. Si les modèles de composants sont justes, c'est-à-dire s'ils représentent tous les comportements normaux, toute contradiction entre les observations et les prédictions déduites des modèles est nécessairement la manifestation d'un dysfonctionnement, c'est-à-dire la manifestation de la présence d'un ou plusieurs défauts. Ce type de raisonnement qui s'appuie sur les composants du système permet de s'affranchir du recensement de tous les défauts simples et multiples nécessaires à l'approche FDI.

II.7.4. Exonération / non- exonération « diagnostic circonstanciel »

Le raisonnement diagnostique dans l'approche DX consiste à dire qu'une faute sur un composant ne peut pas affecter une relation de redondance analytique qui ne comporte pas de modèle du composant. L'idée utilisée dans l'approche FDI est qu'un défaut sur un composant se manifeste nécessairement en affectant les relations de redondance analytique dans lesquelles il est impliqué, les rendant non satisfaites par les observations données.

Il en résulte non seulement, comme dans l'approche DX, que chaque colonne de la matrice de signature impliquée dans une ligne non satisfaite de la même matrice est un défaut candidat, mais aussi que chaque colonne impliquée dans une relation de redondance analytique satisfaite est implicitement disculpée (innocentée) (les lignes satisfaites sont donc ainsi utilisées dans le raisonnement). En fait, ce résultat n'est pas logiquement cohérent : il repose sur l'hypothèse d'exonération qui est implicitement adoptée dans l'approche FDI et doit être considérée explicitement dans le but de comparer l'approche FDI avec l'approche DX.

II.7.5. Lien composant

Dans l'approche DX les composants sont modélisés a priori et explicitement dans le modèle, alors que dans l'approche FDI les composants n'apparaissent pas dans le modèle et c'est a posteriori qu'il est nécessaire de rechercher les défauts pouvant affecter le système. Le diagnostic logique n'a nul besoin de savoir quoi que ce soit a priori sur les défauts ou dysfonctionnement pouvant affecter le système pour pouvoir le diagnostiquer.

II.7.6. Procédure de diagnostic

L'approche FDI aborde le diagnostic par la tâche de détection. Cette phase est appelée génération de résidus. Ces derniers sont décris a priori dans un cahier des charges. C'est dans la phase de génération des résidus que l'on cherche des résultats pour la localisation des défauts (la signature d'un défaut). En revanche, l'approche DX aborde le problème à l'inverse, puisqu'elle pose

en premier lieu le problème algorithmique de la localisation, considérant que la détection d'anomalies relève de la démonstration de la cohérence par un solveur et ne fait pas partie intégrante de la théorie de diagnostic.

II.8. CONCLUSION

L'objectif de FDI est la conception de procédures de décision pour la détection et la localisation des défaillances. Le principal intérêt est d'offrir des techniques sophistiquées afin de combiner les observations, telles que les observateurs et les filtres. DX a pour principal but la localisation en recherchant les sous-ensembles de la description du système qui rentrent en conflit avec les observations. Moins connue par la communauté automatique, une théorie logique pour le diagnostic a été étudiée par la communauté de l'intelligence artificielle [Reiter, 1987][De Kleer, 1987][Greiner, 1989][Poole, 1989]. Néanmoins, il y a des liens entre l'approche logique et celles développées par la communauté du contrôle, étudiées récemment dans [Cordier, et al, 2000][Cordier, et al., 2004]. Cette théorie logique donne des résultats exhaustifs et sains.

Notre étude montre qu'une partie significative de ces deux théories entrent dans une structure commune qui permet une comparaison précise. Quand elles adoptent les mêmes hypothèses par rapport à la manière dont se manifestent les fautes elles-mêmes, FDI et DX s'accordent sur les diagnostics. Ceci offre la possibilité d'une coopération fructueuse entre ces deux approches, en récupérant le meilleur de chacune : compilation des connaissances par modèles sous formes de RRA prenant en compte la position des capteurs avant que les observations ne soient faites, ce qui est le principal avantage de l'approche FDI et, grâce aux hypothèses explicites sur l'état des composants, calculer, dans un même temps, les conflits potentiels (support des tests de détection) pour donner naissance, étant donné une observation, aux conflits sur lesquels la génération des diagnostics est fondée, ce qui constitue l'atout de l'approche DX.

Le raisonnement le plus courant est le raisonnement à base de consistance, formalisé par [Reiter, 1987]. Etant donné un modèle du dispositif, un diagnostic est un ensemble minimal d'hypothèses d'anormalité (défaillance de composants) tel que les observations soient consistantes avec l'hypothèse que tous les autres composants fonctionnent normalement.

L'approche DX explicite a priori les hypothèses sur les états des composants correspondant aux modèles utilisés lors de la construction de tests de détection. Les modèles ainsi construits sont suffisamment riches pour qu'une analyse diagnostique puisse être réalisée sans connaissances supplémentaires. En revanche, FDI construit des tests sans expliciter les liens entres les modèles utilisés et les états des composants correspondants. De ce fait, FDI doit a posteriori construire une table de signature pour rétablir le lien entre les défauts physiques (états anormaux des composants) et les tests utilisés.

Une autre différence majeure entre DX et FDI est qu'aucune hypothèse d'exonération n'est faite implicitement dans l'approche DX. Par contre, pour l'approche FDI, l'hypothèse d'exonération est implicite et conduit à des diagnostics erronés lorsqu'un test théoriquement sensible à un défaut n'engendre pas d'alarme bien que le défaut soit présent.

Chapitre 3

Formalisation proposée pour le diagnostic

III.1. INTRODUCTION

Concevoir des systèmes de diagnostic capables d'analyser formellement des systèmes dynamiques complexes modélisés avec des équations différentielles est un enjeu scientifique important. L'objectif de ce chapitre est de proposer une formalisation pour la modélisation des systèmes dynamiques complexes qui permettent de mettre en place des analyses diagnostiques logique telle que celle utilisés par la communauté DX. Le formalisme proposé permet de tenir compte de la validité du modèle utilisé dans l'analyse diagnostique. La prise en compte de cette validité lors de la détection a été mise en application dans le cas du projet MAGIC [Garcia-Beltran, 2004].

La notion de sous système testable est introduite pour expliciter un certain nombre d'hypothèses implicites aux tests de détection et ainsi permettre une analyse diagnostique logique (formelle). Les algorithmes de détection utilisés pourront être ceux développés par l'approche FDI pour des systèmes dynamiques tels que les observateurs d'états, les relations de parités ou l'estimation paramétrique tandis que l'analyse diagnostique s'appuiera sur une approche formelle de type DX.

Le paragraphe 5 propose une extension du formalisme à la logique floue pour prendre en compte des incertitudes de décision dans l'analyse diagnostique formelle des systèmes dynamiques complexes.

Ce formalisme sera appliqué à la modélisation de deux systèmes physiques connus: un bioprocédé et un moteur électrique à courant continu (MCC).

III.2. MODELISATION PROPOSEE POUR LE DIAGNOSTIC

La modélisation d'un composant ne peut pas toujours se réduire à une relation descriptive de son comportement. En effet, un modèle de comportement n'est pas universel ; il n'est généralement valide que dans des plages ou des modes de fonctionnement particuliers. De plus, un modèle représente un état particulier d'un composant : il peut décrire un comportement normal ou différents types de comportement défaillant.

Les phénomènes vrais, c'est-à-dire tels qu'ils pourraient être mesurés par un capteur parfait, seront appelés variables physiques. Ces variables, a fortiori mesurables, sont indépendantes de tout modèle et doivent être distinguées des paramètres. Un ~ surmontant une variable signifie qu'il s'agit d'une valeur représentative d'une variable physique. $\tilde{u}$ correspond par exemple à la valeur mesurée de [u]. Dans le cas général, les valeurs mesurées sont des observations représentant des variables de commande ou bien des mesures. Les valeurs renseignées (mesurées) sont importantes pour le diagnostic de défauts car c'est la seule information qui peut nous renseigner sur l'état actuel du système.

Définition 3.1

Un système physique Σ est caractérisé par un ensemble de variables physiques: χ et un ensemble d'informations ou de données caractérisant son état $\tilde{\gamma}$.

Une contrainte sur Σ est une fonction $\Pi(\chi_k, \tilde{\gamma}_k) : \chi_k \subseteq \chi, \tilde{\gamma}_k \subseteq \tilde{\gamma}$ admettant une fonction de vérité $\top(\Pi(\chi_k, \tilde{\gamma}))$ à valeur soit dans {*vrai*, *faux*} soit dans [0,1] s'il s'agit d'un ensemble d'appartenance à vrai.

Un modèle élémentaire autonome de Σ s'écrit :

$$M_k = \left(R_k(\chi_k, \tilde{\gamma}_k), V_k(\chi'_k, \tilde{\gamma}'_k), H_k\right) \ : \ \chi'_k \subseteq \chi_k \subseteq \chi, \tilde{\gamma}_k \subseteq \tilde{\gamma}, \tilde{\gamma}'_k \subseteq \tilde{\gamma}, \qquad (3.1)$$

où :

- $R_k(\chi_k, \tilde{\gamma}_k)$ représente une contrainte de comportement (ou relation) qui est satisfaite dans l'état H_K si la contrainte (ou condition) de validité V_k est vérifiée. Il s'agit d'une relation élémentaire décrivant les liens entre l'ensemble des phénomènes observables [x] et connus $\tilde{x}$ du système physique. Elle peut être de différentes natures : une équation différentielle, un réseau neuronal...

  Remarque : Seuls les modèles élémentaires autonomes, c'est-à-dire pour lesquels $\chi'_k \subseteq \chi_k$, sont considérés ici.

- $V_k(\chi'_k, \tilde{\gamma}'_k)$ définit la condition de validité de la relation élémentaire $R_k(\chi_k, \tilde{\gamma}_k)$. Le résultat est ici supposé booléen (mais il pourrait aussi être un degré d'appartenance à l'ensemble valide). Il s'agit d'une fonction de variables physiques (nous supposerons dans la suite que seuls les phénomènes apparaissant dans $R_k(\chi_k, \tilde{\gamma}_k)$ peuvent apparaître). Si on prend l'exemple des deux bacs (cf. chapitre 2), le modèle associé à chacun des bacs doit tenir compte du fait que le modèle de comportement n'est valide que si le niveau de produit dans le bac est inférieur à 50cm, c'est-à-dire à la hauteur du bac. Au-delà, le modèle ne s'applique plus.

- H_k est une hypothèse (ou mode suivant [Dague, 2001]) représente l'état E_i d'un composant C_j : H_k=E_i(C_j) lorsque son comportement est conforme à $R_k(\chi_k, \tilde{\gamma}_k)$ sous la condition $V_k(\chi'_k, \tilde{\gamma}'_k)$. Nous adopterons les notations de la théorie de la circonscription [Mc Carthy, 1986]. Si l'état est normal, nous noterons $H_0 = \neg AN(C_j)$ (AN signifie anormal et $\neg AN$ signifie normal) et si l'état correspond à l'état défaillant D_i, nous noterons : $H_k = D_i(C_j)$. La clause suivante est toujours satisfaite : $D_i(C_j) \Rightarrow AN(C_j)$ (par contre, la réciproque n'est pas vraie).

III.2.1. Application à la machine électrique à courant continu (MCC)

Prenons l'exemple d'un moteur électrique à courant continu alimenté par un convertisseur de puissance commandé numériquement et couplé à une charge mécanique. Les modèles élémentaires des composants disponibles, correspondant à la description du système noté SD dans [Reiter, 1987], avec leur domaine de validité et leur support en terme de composants sont donnés ci-dessous :

$$
\begin{cases}
M_1 = \{[u] = K_u[u_c], |[u]| \le U_{\max}, \neg AN(\mathrm{C}_1)\} \\
M_2 = \left\{[u] = R[i] + L\frac{d}{dt}[i] + \Phi[i_e][\Omega], -, \neg AN(\mathrm{C}_2)\right\} \\
M_3 = \left\{[u] = \rho R[i] + \rho L\frac{d}{dt}[i] + \Phi[i_e][\Omega], 0 \le \rho \le 0.95, CC(\mathrm{C}_2)\right\} \\
M_4 = \left\{J\frac{d}{dt}[\Omega] = \Phi[i_e][i] - [t_r], -, \neg AN(\mathrm{C}_3)\right\} \\
M_5 = \{[t_r] = t_0 + t_1[\Omega], [t_r] \ge t_0, \neg AN(\mathrm{C}_4)\} \\
M_6 = \{[t_r] = -t_0 + t_1[\Omega], [t_r] \le t_0, \neg AN(\mathrm{C}_4)\} \\
M_7 = \{\tilde{u}_c = [u_c], \text{-}, \neg AN(\mathrm{C}_5)\} \\
M_8 = \{\tilde{i}_e = [i_e], \text{-}, \neg AN(\mathrm{C}_6)\} \\
M_9 = \{\tilde{u} = [u], \text{-}, \neg AN(\mathrm{C}_7)\} \\
M_{10} = \{\tilde{i} = i, \text{-}, \neg AN(\mathrm{C}_8)\} \\
M_{11} = \{\tilde{\Omega} = \Omega, \text{-}, \neg AN(\mathrm{C}_9)\}
\end{cases}
$$

Figure 3.1 - La description du système de la machine à courant continu (MCC)

M_1 est le modèle élémentaire du composant C_1 (convertisseur de puissance). La relation élémentaire correspondante est R_1 : $[u] = K_u[u_c]$. Cette dernière est valide dans V_1 : $|[u]| \le U_{\max}$. Cette relation élémentaire modélise le bon comportement du composant C_1. En effet, $H_1 = \neg AN(\mathrm{C}_1)$.

La liste des composants intervenant dans la (figure 3.1) est la suivante : C_1 : convertisseur de puissance C_2 : partie électrique C_3 : partie mécanique C_4 : charge mécanique C_5 : convertisseur numérique analogique (CNA) C_6: capteur de courant magnétisant virtuel (la valeur est supposée connue) C_7 : capteur de tension, C_8 : capteur de courant, C_9 : capteur de vitesse. Les composants C_5, C_6, C_7, C_8 et C_9 sont des composants d'information dont les modèles sont généralement très simples mais qu'il est indispensable de faire apparaître.

Notons qu'il peut exister plusieurs contraintes pour un même composant. Par exemple, les modèles M_2 et M_3 sont dédiés au composant C_2, l'un modélisant le comportement normal, représenté par l'hypothèse suivante $\neg AN(C_2)$ et l'autre, le comportement lorsqu'un court-circuit affecte la partie électrique du moteur, illustré par l'hypothèse suivante $CC(\mathrm{C}_2)$. Les modèles M_5 et M_6 repose sur l'hypothèse de bon comportement du composant C_4 mais se distingue par leur domaine de validité. En effet, pour M_5 la relation élémentaire $[t_r] = t_0 + t_1[\Omega]$ n'est valide que si $[t_r] \le t_0$. En revanche, pour M_6 la relation élémentaire $[t_r] = -t_0 + t_1[\Omega]$ est valide quand $[t_r] \le t_0$. La figure 3.2 illustre ce cas de figure.

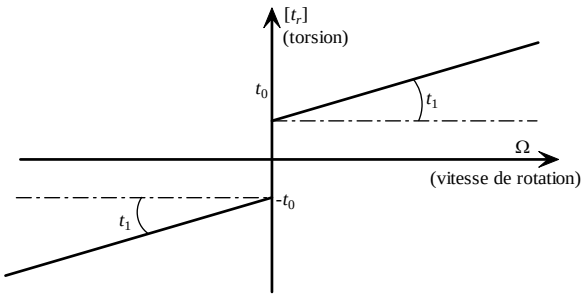


Figure 3.2 – Caractéristique de la charge mécanique

III.2.2. Application à un Bioprocédé

Le bioprocédé considéré est composé d'un agitateur de substrat, d'un bioréacteur, d'un échangeur de chaleur, d'un appareil de chauffage électrique, d'une pompe, plusieurs tuyaux et 21 capteurs. Une représentation schématique du système est donnée par la figure 3.3.

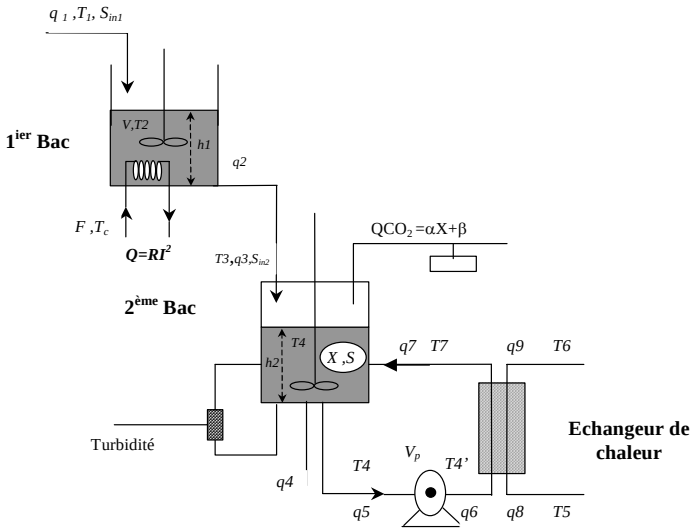


Figure 3.3 – Un système de bioprocédé

Le problème principal qui se pose pour cette application est la surveillance du système pendant que plusieurs types de défauts peuvent se produire : les défauts de capteur, défauts d'actionneur (pompes, valves…) ou même un dysfonctionnement du bioprocédé ; par exemple quand les bioréactions ne se produisent pas comme prévu (en effet, le bioprocédé est fait d'organismes vivants qui peuvent changer de propriétés par mutation, alors le processus pourrait être dans un état anormal). Les procédures diagnostiques doivent réduire le temps de maintenance du système en utilisant des informations valables sur le processus telles que le modèle de comportement du bioprocédé, les valeurs connues via les capteurs physiques et les actionneurs.

Grâce à un convertisseur numérique analogique, une vanne commande le flux d'entrée d'eau qui remplit le bac du haut. Pendant que le réservoir supérieur se remplit, le produit qu'il contient peut s'écouler à travers une restriction (q_2) pour passer dans le bac du bas (le bioréacteur). Le produit contenu dans le deuxième bac peut s'écouler aussi en dehors du bioréacteur par la restriction du bas (q_4). Les niveaux d'eau dans le bac du haut (dénoté par h_1) et dans le bac du bas (dénoté par h_2) sont mesurés grâce à des capteurs précis, sachant que ($h_{1max} = h_{2max} = 2m$). Un SCADA numérique contrôle tout le système à une période d'échantillonnage égale 10s.

Le mélangeur (1^{er} bac) est composé d'un système de brassage, un appareil de chauffage électrique et une restriction de sortie. Le bioréacteur ($2^{ème}$ bac) est composé d'un système de brassage, un capteur de la turbidité du produit et une restriction de sortie. La fonction principale du bioréacteur est d'augmenter le nombre de micro-organismes. Le bioréacteur a quatre rôles essentiels: récipient, stérilisateur, ventilateur et, dans la plupart des cas, agitateur. Il doit autoriser le contrôle des paramètres physiques et chimiques de fermentation. L'agitation est assurée par un système de brassage qui tourne dans les régimes déterminés, grâce à un moteur. De plus, l'échangeur de chaleur est fait de deux parties : le côté gauche est traversé par le produit chauffé et

le côté droit est traversé par l'eau froide. Le rôle d'un échangeur de chaleur est de transférer de la chaleur d'un fluide chaud vers un fluide circulant à contre-courant, l'échange s'effectuant à travers une paroi solide.

Les variables commandées sont : le courant de la provision de l'agitateur, la température (T_1) du produit à l'entrée, l'intensité (I) du courant traversant la résistance chauffante, la quantité du produit (q_2) s'écoulant de la première restriction, la quantité du flux de sortie(q_{4on}) au moment de vider le réservoir du bas, la vitesse de la pompe (V_p), le flux d'eau froide (q_8) dans l'échangeur de chaleur et la température de l'eau froide (T_5). De plus, plusieurs capteurs (cf. figure 3.3) appartiennent au système : ils mesurent la température de l'entrée (T_1), la température du produit (T_2) dans le mélangeur, l'intensité du courant (I), le flux d'entrée du produit dans le mélangeur (q_1), la quantité du produit (q_2) s'écoulant de la première restriction, le flux de sortie pendant la vidange du deuxième réservoir (q_{4on}), le flux d'eau froide dans l'échangeur de chaleur (q_8), la hauteur du produit (h_1) dans le mélangeur, la hauteur du produit (h_2) dans le bioréacteur, la température du produit (T_4), la température du produit dans l'échangeur de chaleur côté eau froide (T_6) et la température du produit dans l'échangeur côté produit (T_7).

La liste des composants résumés par le tableau 3.1 est la suivante : Mélangeur, Résistance de chauffage, Tuyau 1, Bioréacteur, Echangeur côté eau chaude, Echangeur côté eau froide, Pompe, capteur de Turbidité. Les capteurs de températures sont donnés par T_1, T_2, T_3, T_4, $T_{4'}$, T_5, T_6, T_7. Composants q_1, q_2, q_3, q_5, q_6, q_7, q_8 et q_9 sont des capteurs de flux. h_1, h_2 sont des capteurs de la hauteur d'eau dans les bacs. « I » le capteur qui mesure l'intensité qui traverse la résistance de chauffage et « V » le capteur de vitesse de la pompe. Tous ces capteurs sont des composants d'information dont les modèles sont généralement très simples.

L'ensemble des modèles des composants est donné par le tableau 3.1. Ce dernier regroupe les modèles de diagnostic élémentaires tels qu'ils sont tirés de la physique avec leur domaine de validité et leur support en terme de composants.

Relations Elémentaires	Conditions de validité	Etats des composants	Relations Elémentaires	Conditions de validité	Etats des composants
R_1	$0<h_1<2m$	¬*AN (mélangeur)*	R_{19}	_	¬AN(turbidité)
R_2	$(0<h_1<2cm) \wedge (10^{oc}<T_2)$		R_{20}		¬AN(capteur T_1)
R_3	$0<h_1<50cm$		R_{21}	_	¬AN(capteur T_2)
R_4	$0<i<150A$	¬*AN (résitance)*	R_{22}	_	¬AN(capteur T_3)
R_5	_	¬*AN (Tuyau 1)*	R_{23}	_	¬AN(capteur T_4)
R_6	_		R_{24}	_	¬AN(capteur T_5)
R_7	$0<h_2<2m$	¬*AN (bioréacteur)*	R_{25}	_	¬AN(capteur T_6)
R_8	$0<h_2<50cm$		R_{26}	_	¬AN(capteur T_7)
R_9	_		R_{27}	_	¬AN(actionneur q_1)
R_{10}	_		R_{28}	_	¬AN(actionneur q_2)
R_{11}	_		R_{29}	_	¬AN(actionneur q_3)
R_{12}	_	¬*AN (échangeur de chaleur coté eau froide)*	R_{30}	_	¬AN(actionneur q_5)
R_{13}	_		R_{31}	_	¬AN(actionneur q_6)
R_{14}	_	¬*AN (échangeur de chaleur coté eau chaude)*	R_{32}	_	¬AN(capteur courant)
R_{15}	_		R_{33}	_	¬AN(capteur h_1)
R_{16}	_	¬*AN(pompe)*	R_{34}	_	¬AN(capteur h_2)
R_{17}	_				
R_{18}	_				

Table 3.1 - Description du système Bioprocédé

Les relations élémentaires pour les modèles de diagnostic sont données par la figure 3.4.

$$
\begin{cases}
R_1: & \dfrac{d[h_1]}{dt} = \dfrac{[q_1]-[q_2]}{S_1} \\
R_2: & \dfrac{d[T_2]}{dt} = \dfrac{[q_1]([T_1]-[T_2])}{S_1[h_1]} + \dfrac{[Q_e]}{S_1[h_1]\rho.c_p} \\
R_3: & [q_2] = \alpha_1[h_1]\theta_2 \\
R_4: & [Q_e] = R[i]^2 \\
R_5: & [T_2] = [T_3] \\
R_6: & [q_2] = [q_3] \\
R_7: & \dfrac{d[h_2]}{dt} = \dfrac{[q_3]-[q_4]-[q_5]+[q_7]}{S_2} \\
R_8: & [q_4] = \alpha_2[h_2]p_4 \\
R_9: & \dfrac{d[b_m]}{dt} = \mu_M \dfrac{[\sigma]}{k+[\sigma]}[b_m] - \dfrac{[b_m]}{S_2[h_2]}[q_3] \\
R_{10}: & \dfrac{d[\sigma]}{dt} = -\mu_M \dfrac{[\sigma]}{k+[\sigma]}\dfrac{[b_m]}{y_x} + \dfrac{[q_3]}{S_2[h_2]}.(S_I - [\sigma]) \\
R_{11}: & S_2[h_2]\dfrac{d[T_4]}{dt} = [q_3]([T_3]-[T_4]) - [q_5][T_4] + [q_7][T_7] + \dfrac{0.216\mu_M[\sigma] + 0.024[b_m](k+[\sigma])}{(k+[\sigma]).\rho.c_p} \\
R_{12}: & \rho_w V_{e1} C_w \dfrac{d[T_6]}{dt} = \rho_w[q_8]C_w([T_5]-[T_6]) - a([T_6]-[T_7]) \\
R_{13}: & [q_8] = [q_9] \\
R_{14}: & \rho_p V_{e2} C_p \dfrac{d[T_7]}{dt} = \rho_p[q_6]C_p([T_4']-[T_7]) + a([T_6]-[T_7]) \\
R_{15}: & [q_6] = [q_7]
\end{cases}
\qquad
\begin{cases}
R_{16}: & [q_6] = \gamma.[v_p] \\
R_{17}: & [q_6] = [q_5] \\
R_{18}: & [T_4] = [T_4'] \\
R_{19}: & \tilde{T} = \beta.[b_m] \\
R_{20}: & \tilde{T}_1 = [T_1] \\
R_{21}: & \tilde{T}_2 = [T_2] \\
R_{22}: & \tilde{T}_3 = [T_3] \\
R_{23}: & \tilde{T}_4 = [T_4] \\
R_{24}: & \tilde{T}_5 = [T_5] \\
R_{25}: & \tilde{T}_6 = [T_6] \\
R_{26}: & \tilde{T}_7 = [T_7] \\
R_{27}: & \tilde{q}_1 = [q_1] \\
R_{28}: & \tilde{q}_2 = [q_2] \\
R_{29}: & \tilde{q}_3 = [q_3] \\
R_{30}: & \tilde{q}_5 = [q_5] \\
R_{31}: & \tilde{q}_6 = [q_6] \\
R_{32}: & \tilde{i} = [i] \\
R_{33}: & \tilde{h}_1 = [h_1] \\
R_{34}: & \tilde{h}_2 = [h_2]
\end{cases}
$$

Figure 3.4 - Relations élémentaires

où les valeurs de tous les paramètres qui interviennent dans la description du système sont données ci-dessous :

$\alpha_1 = \alpha_2 = 0.05, S_1 = \pi \times 0.1 \times 0.1(m^2), S_2 = \pi \times 0.1 \times 1.2(m^2), R = 0.5(\Omega), c_p = 0.1, K = 10, a = 1,$

$\rho_p = 1000$ (mélange réactionnel), $C_p = 1, V_{e1} = 10$ (volume échangeur coté eau froide),

$\rho_w = 1000, C_w = 1, V_{e2} = 1$ (volume échangeur coté produit), $y_x = 0.6, \mu_M = 0.5, \alpha = 100.$

III.3. REPRESENTATION STRUCTURELLE D'UN SYSTEME

Du point de vue structurel, le modèle est considéré comme un ensemble de contraintes [Declerck, 1991] qui s'appliquent à un ensemble de paramètres et de variables dont certaines sont mesurées. Une contrainte est une relation qui relie des variables qui appartiennent soit à l'ensemble des variables physiques du système étudié, soit éventuellement à des paramètres de modèle. L'ensemble des variables considérées est celui qui a été retenu pour décrire l'évolution temporelle du système. Le nom *contrainte* fait référence à une relation élémentaire. L'écriture habituelle de la loi d'Ohm *U= RI* qui n'est autre qu'un sens de calcul de la variable *U* à partir de la valeur connue de *I*, et la forme *I = U/R* sera utilisée si la variable *U* est connue. Par conséquent, cette relation élémentaire est sans causalité (au sens de la calculabilité). En effet, chaque variable peut être déduite des autres variables. Pour toutes les relations sous forme d'équations différentielles, nous avons préféré les intégrer et donc, imposer une causalité au sens de la calculabilité [Iwasaki et al., 1994].

Dans l'approche structurelle, le système est représenté souvent par des graphes (graphe structurel, graphe bipartie [Declerck et Staroswiecki, 1991]), cette représentation facilite l'analyse structurelle du système. Dans un graphe structurel, un nœud représente une variable physique impliquée dans la description du système. Un arc représente un modèle de composant. Comme le montrent les exemples qui suivent, un défaut de capteurs et un défaut de comportement du système physique sont modélisés avec le même formalisme. Il en va de même pour la redondance matérielle. Par exemple, le circuit physique faisant intervenir une résistance, un capteur de tension U et un capteur de courant I, se modélise avec un graphe structurel (figure 3.5). Chaque arc représente une relation élémentaire (R_k) et chaque nœud représente une variable physique impliquée dans la description du système. Les nœuds pleins sont les mesures des variables.

La description du système est donnée par les relations élémentaires (contraintes) décrivant le comportement des trois composants impliqués dans le système physique : le capteur de tension, le capteur de courant et la résistance :

$$\begin{cases} R_1 : U = RI, -, \neg AN(R) \\ R_2 : U = \tilde{U}, -, \neg AN(Capteur \text{ de tension}) \\ R_3 : I = \tilde{I}, -, \neg AN(Capteur \text{ de courant}) \end{cases}$$

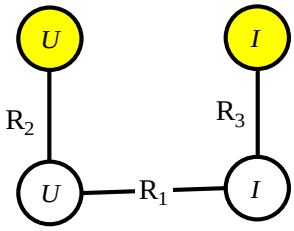


Figure 3.5 - Représentation structurelle

Le graphe structurel correspondant à l'exemple de la MCC est donné par la figure 3.6. Une orientation au sens de la calculabilité est induite par les relations élémentaires R_2 et R_3 et R_4.

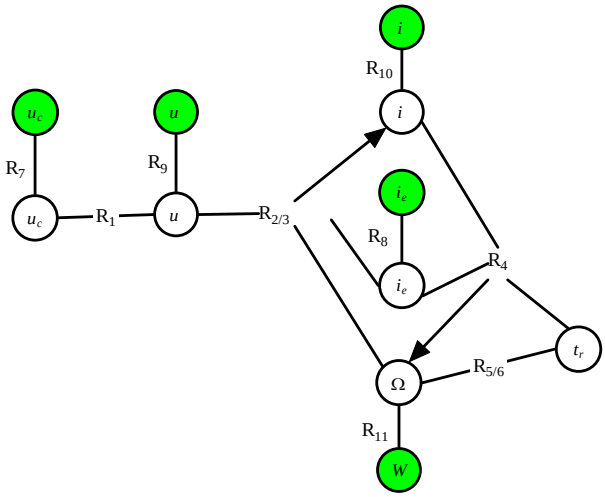


Figure 3.6 - Représentation structurelle pour la machine MCC

L'analyse diagnostique repose sur les résultats des tests de détection qui ont pour but de comparer le comportement réel du système physique avec son comportement modélisé. Les tests de détection doivent être conçus à partir des modèles élémentaires. Ce qui signifie qu'un test de détection est dédié à tester un sous système (c'est-à-dire une partie du système physique global). Or, la mise en œuvre des tests de détection nécessite de construire des modèles testables, c'est-à-dire ne faisant intervenir que des variables connues (commandées, mesurées ou connues à l'avance).

L'existence de tels modèles est conditionnée par celle d'une redondance d'information sur le système. Ces modèles doivent permettre de détecter les défaillances (détectabilité) et de les localiser, c'est-à-dire de les distinguer les unes des autres (distinguabilité). Pour cela, on va introduire la notion de sous-système testable (SST).

III.4. DEFINITION D'UN SOUS SYSTEME TESTABLE

Soit Σ un système physique constitué d'un ensemble $\mathscr{C}_\Sigma$ de composants, où chaque composant élémentaire est noté par C_i. Si l'état d'un composant modélisé est noté H_k, alors $\mathscr{H}_\Sigma$ est l'ensemble des états H_k pour tous les composants modélisés dans Σ.

Définition 3.2:

Un ensemble de modèles élémentaires $\{M_k : k \in \varphi\}$ conduit à un sous-système Σ_i :

$$\Sigma_i = \left(\left\{ R_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi \right\}, \left\{ V_k(\chi'_k, \tilde{\gamma}'_k) : k \in \varphi' \subseteq \varphi \right\}, \bigwedge_{k \in \varphi} H_k \right) \tag{3.2}$$

s'il n'y a pas d'hypothèses H_k contradictoires, c'est-à-dire d'hypothèses $H_{k1}=E_i(C_j)$ et $H_{k2}=E_l(C_j)$ portant sur un même composant C_j telles que $E_i(C_j) \wedge E_l(C_j) \vdash \perp$ et s'il existe au moins une valeur χ et de $\tilde{\gamma}$ telle que $\bigwedge_k \top(V_k(\chi'_k, \tilde{\gamma}'_k)) \vdash \top ; \chi'_k \in \chi, \tilde{\gamma}'_k \in \tilde{\gamma}$.

Définition 3.3:

Un ensemble de contraintes $\{\Pi_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}$ sera dit complet s'il est surdéterminé d'une contrainte, c'est-à-dire si:

- $\dim\left(\bigcup_k \chi_k\right) = n$
- $\mathrm{card}(\varphi) = n+1$

Définition 3.4:

Un ensemble complet de contraintes sera dit minimal s'il ne contient pas d'autres ensembles de contraintes complets que lui-même.

Comme un diagnostic s'exprime en termes d'états de composants alors, à tout test de détection, doit être associé un support qui réunit tous les états de composants physiques associés aux relations comportementales élémentaires réunies par un sous système testable (SST). Ce support de test est une conjonction des différentes hypothèses sur l'état des composants ($\wedge H_k$) avec $k \in \varphi$.

L'objectif est d'exploiter au maximum l'information structurelle sur le système afin de construire les supports de tous les tests possibles. Si un test de détection repose nécessairement sur une relation de redondance analytique alors il faut adjoindre à cette relation un domaine de validité

correspondant à l'union des domaines de validité de chacune des relations comportementales élémentaires.

Par exemple, le modèle d'une machine à courant continu où seuls la tension d'induit, la vitesse de rotation et le couple résistant sont connus, est constituée de deux relations comportementales élémentaires : l'équation électrique et l'équation mécanique. Or, comme le courant n'est pas mesuré, ces relations ne peuvent être testées séparément. La relation de redondance analytique est le modèle qui permet de déduire la vitesse à partir de la tension d'induit et du couple résistant. Un domaine de validité d'une autre nature peut être associé au hacheur sous la forme d'un intervalle de tension de non saturation. Le domaine de validité d'un sous-système testable est important car le test de détection en hérite : un résultat n'est valide que si l'état du système étudié appartient au domaine de validité qu'il est crucial de formaliser.

Exemple 1

Soient trois modèles élémentaires donnés par :

$$M_1 = \left(y - 2x = 3, x + y \le 6, H_1\right)$$
$$M_2 = \left(x = \tilde{x}, -, H_2\right)$$
$$M_3 = \left(y = \tilde{y}, -, H_3\right)$$

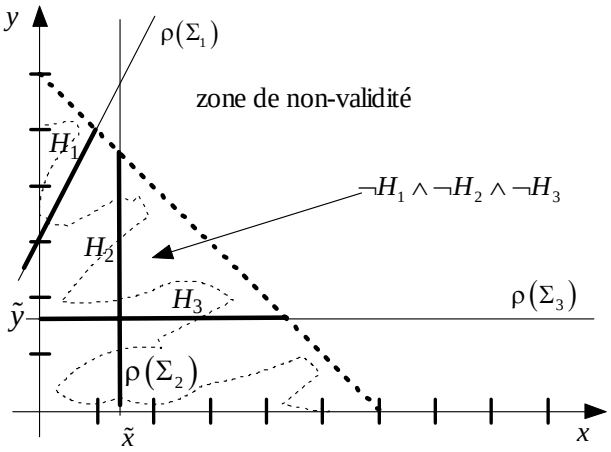


Figure 3.7 - Représentation des contraintes et leurs domaines de validités.

Trois sous systèmes testables sont ainsi déduits :

$$\Sigma_a = \left(\{y - 2x = 3\} \cup \{x = \tilde{x}\} \cup \{y = \tilde{y}\}, \{x + y \le 6\} \cup \{x = \tilde{x}\} \cup \{y = \tilde{y}\}, H_1 \wedge H_2 \wedge H_3\right)$$
$$\Sigma_b = \left(\{y - 2x = 3\} \cup \{x = \tilde{x}\} \cup \{y = \tilde{y}\}, \{x + y \le 6\} \cup \{x = \tilde{x}\} \cup \{y - 2x = 3\}, H_1 \wedge H_2 \wedge H_3\right)$$
$$\Sigma_c = \left(\{y - 2x = 3\} \cup \{x = \tilde{x}\} \cup \{y = \tilde{y}\}, \{x + y \le 6\} \cup \{y = \tilde{y}\} \cup \{y - 2x = 3\}, H_1 \wedge H_2 \wedge H_3\right)$$

Les sous-systèmes testables Σ_a, Σ_b et Σ_c testent tous les mêmes hypothèses mais ils s'appuient sur des conditions de validité différentes. Pour tester la validité,

- Σ_a utilise des relations qui reposent sur les hypothèses H_2 et H_3
- Σ_b utilise des relations qui reposent sur les hypothèses H_1 et H_2
- Σ_c utilise des relations qui reposent sur les hypothèses H_1 et H_3

Les SST conduisent aux tests suivants (toutes les fonctions de test sont identiques):

$T_a : \tilde{y} - 2\tilde{x} = 3 \text{ si } \tilde{x} + \tilde{y} \leq 6$
$T_b : \tilde{y} - 2\tilde{x} = 3 \text{ si } \tilde{x} + 1 \leq 2$
$T_c : \tilde{y} - 2\tilde{x} = 3 \text{ si } \tilde{y} - 1 \leq 4$

Hypothèse de *v*-exonération

Quelles que soient les contraintes utilisées pour construire une fonction de validité, si cette fonction $V(\Sigma_i)$ est vraie alors toutes les conditions de validité intervenant dans la fonction seront considérées comme vraies.

Remarque 1 :

Cette hypothèse est nécessaire car l'analyse diagnostique formelle ne permet de tirer une conclusion d'un test consistant (il faudrait pouvoir reproduire le test dans tous les contextes possibles).

Remarque 2 :

Cette hypothèse est plus conservative que d'exonérer en plus les hypothèses H_j implicites aux contraintes $\rho(\Sigma_j)$ utilisées lors de la construction de la fonction de validité.

Remarque 3 :

Dès qu'interviennent des conditions de validité, l'analyse diagnostic perd ses propriétés de garantie du fait de l'hypothèse de *v*-exonération.

Exemple :

Si $\tilde{x} = 1, \tilde{y} = 2$ (inconsistant),

$$\left.\begin{array}{l} \upsilon(\Sigma_a) \underset{v-exoneration}{\rightarrow} (H_2 \wedge H_3 \wedge) V(M_1) \\ \upsilon(\Sigma_b) \underset{v-exoneration}{\rightarrow} (H_1 \wedge H_2 \wedge) V(M_2) \\ \upsilon(\Sigma_c) \underset{v-exoneration}{\rightarrow} (H_1 \wedge H_3 \wedge) V(M_3) \end{array}\right\} \underset{\text{inconsistance}}{\rightarrow} \neg H_1 \vee \neg H_2 \vee \neg H_3$$

Avec une exonération complète, le résultat inconsistant du test serait contradictoire avec ceux des tests de validité (les trois conditions de validité exonéreraient H_1, H_2 et H_3 alors que le test de détection est négatif).

Si $\tilde{x} = 3, \tilde{y} = 4$ (inconsistant),

$$\neg\upsilon(\Sigma_a)\rightarrow\neg H_2\vee\neg H_3\vee\neg v(M_1)\underset{\substack{\text{aucune certitude}\\\text{sur la validité}}}{\rightarrow}\text{pas de conclusion}$$

$$\neg\upsilon(\Sigma_b)\rightarrow\neg H_1\vee\neg H_2\vee\neg v(M_2)\underset{\substack{\text{aucune certitude}\\\text{sur la validité}}}{\rightarrow}\text{pas de conclusion}$$

$$\upsilon(\Sigma_c)\underset{v-exoneration}{\rightarrow}(H_1\wedge H_3\wedge)v(M_3)\underset{\text{inconsistance}}{\rightarrow}\neg H_1\vee\neg H_2\vee\neg H_3$$

Si $\tilde{x}=3,\tilde{y}=6$ (inconsistant),

$$\left.\begin{array}{l}\neg\upsilon(\Sigma_a)\rightarrow\neg H_2\vee\neg H_3\vee\neg V(M_1)\\\neg\upsilon(\Sigma_b)\rightarrow\neg H_1\vee\neg H_2\vee\neg V(M_2)\\\neg\upsilon(\Sigma_c)\rightarrow\neg H_1\vee\neg H_3\vee\neg V(M_3)\end{array}\right\}\underset{\substack{\text{aucune certitude}\\\text{sur la validité}}}{\rightarrow}\text{pas de conclusion}$$

Si $\tilde{x}=1,\tilde{y}=5$ (consistant),

$$\upsilon(\Sigma_a)\underset{v-exoneration}{\rightarrow}(H_2\wedge H_3\wedge)V(M_1)\underset{\text{consistance}}{\rightarrow}\text{pas de conclusion}\left(H_1\wedge H_2\wedge H_3\text{ si exonération}\right)$$

$$\upsilon(\Sigma_b)\underset{v-exoneration}{\rightarrow}(H_1\wedge H_2\wedge)V(M_2)\underset{\text{consistance}}{\rightarrow}\text{pas de conclusion}\left(H_1\wedge H_2\wedge H_3\text{ si exonération}\right)$$

$$\upsilon(\Sigma_c)\underset{v-exoneration}{\rightarrow}(H_1\wedge H_3\wedge)V(M_3)\underset{\text{consistance}}{\rightarrow}\text{pas de conclusion}\left(H_1\wedge H_2\wedge H_3\text{ si exonération}\right)$$

Les trois tests précédents sont issus d'un même sous-système Σ réunissant les modèles élémentaires Σ_1, Σ_2 et Σ_3. La différence tenait en ce que les conditions de validité étaient construites différemment. Or, cet exemple montre que du fait de l'hypothèse de *v*-exonération, il suffit de trouver une fonction de validité vraie pour en déduire que parmi l'ensemble des hypothèses H_i utilisées pour construire la fonction de test à partir des $\rho(H_i)$, l'une au moins ne peut être satisfaite.

Application MCC

Soit trois modèles élémentaires M_1, M_7 et M_9 pris parmi l'ensemble de modèles élémentaires qui décrivent le comportement du système de la MCC. Dans ce cas de figure, Σ représente le sous système physique composé de trois composants, convertisseur de puissance, convertisseur numérique analogique et le capteur de tension, tel que $\mathcal{C}_\Sigma$={ C_1, C_5, C_7}. Ces trois composants sont dans un état normal. Par conséquent, l'ensemble $\mathcal{H}_\Sigma$ s'écrit $\mathcal{H}_\Sigma=\{\neg AN(C_1),\neg AN(C_5),\neg AN(C_7)\}$.

$$M_1=\left([u]=K_u[u_c],|[u]|\le U_{\max},\neg AN(\mathrm{C}_1)\right)$$
$$M_7=\left(\tilde{u}_c=[u_c],\text{-},\neg AN(\mathrm{C}_5)\right)$$
$$M_9=\left(\tilde{u}=[u],\text{-},\neg AN(\mathrm{C}_7)\right)$$

Deux sous systèmes testables sont ainsi déduits :

$$\Sigma_1=\left(\{[u]=K_u[u_c]\}\cup\{\tilde{u}_c=[u_c]\}\cup\{\tilde{u}=[u]\},\{|[u]|\le U_{\max}\}\cup\{\tilde{u}=[u]\},\neg AN(\mathrm{C}_1)\wedge\neg AN(\mathrm{C}_5)\wedge\neg AN(\mathrm{C}_7)\right)$$
$$\Sigma_2=\left(\begin{array}{l}\{[u]=K_u[u_c]\}\cup\{\tilde{u}_c=[u_c]\}\cup\{\tilde{u}=[u]\},\{|[u]|\le U_{\max}\}\cup\{[u]=K_u[u_c]\}\cup\{\tilde{u}_c=[u_c]\},\\\neg AN(\mathrm{C}_1)\wedge\neg AN(\mathrm{C}_5)\wedge\neg AN(\mathrm{C}_7)\end{array}\right)$$

Les sous-systèmes testables Σ_1 et Σ_2 testent les mêmes hypothèses $\neg AN(\mathrm{C}_1)$, $\neg AN(\mathrm{C}_5)$ et $\neg AN(\mathrm{C}_7)$ mais ils s'appuient sur des conditions de validité différentes. Pour tester la validité :

- Σ_1 utilise des relations qui reposent sur l'hypothèses $\neg AN(C_7)$.
- Σ_2 utilise des relations qui reposent sur les hypothèses $\neg AN(C_1)$ et $\neg AN(C_5)$

La différence entre Σ_1 et Σ_2 tenait en ce que les conditions de validité étaient construites différemment.

Le fait de pouvoir combiner les trois relations élémentaires correspondantes aux trois modèles élémentaires M_1, M_7, M_9 en substituant les variables u et u_c par leurs mesures, nous a permis de construire une relation testable (voir figure 3.7). Le nombre de variables connues est de deux.

Les deux tests précédents Σ_1 et Σ_2 sont issus d'un même sous-système réunissant les modèles élémentaires M_1, M_7 et M_9. La différence entre les deux tests est que les conditions de validité étaient construites différemment. Or, cet exemple montre que du fait de l'hypothèse de *v*-exonération, il suffit de trouver une fonction de validité vraie pour en déduire que parmi l'ensemble des hypothèses H_i utilisées pour construire la fonction de test à partir des $\rho(H_i)$, l'une au moins ne peut être satisfaite.

$$M_1 = \left([u] = K_u[u_c], |[u]| \le U_{\max}, \neg AN(C_1)\right)$$
$$M_7 = \left(\tilde{u}_c = [u_c], -, \neg AN(C_5)\right)$$
$$M_9 = \left(\tilde{u} = [u], -, \neg AN(C_7)\right)$$

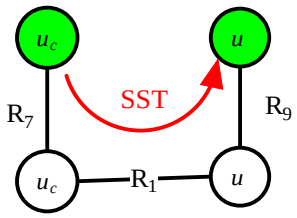


Figure 3.7 – Exemple d'un sous système testable

Définition 3.5:

Un sous-système Σ_i décrit par :

$$\Sigma_i = \left(\{R_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}, \{V_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}, \bigwedge_{k\in\varphi} H_k \right)$$

sera dit testable si

- $\{R_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}$ est un sous-ensemble de contraintes complet minimal (3.3a)

- $\exists \phi \in \varphi / \{R_k(\chi_k, \tilde{\gamma}_k) : k \in \phi\} \cup \{V_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}$ forme un sous-ensemble de contraintes complet minimal (3.3b)

Il y a plusieurs approches qui permettent de construire les différents tests de détection d'un système. Toutes ces approches ont un point commun, elles cherchent à éliminer les variables inconnues et à conserver les variables connues afin de réaliser les tests. La difficulté est de trouver une nouvelle relation satisfaisant en même temps les deux conditions (3.3a) et (3.3b).

La différence entre les approches se situe dans la qualité des résultats, certaines approches sont complètes, d'autres non. Ce que nous entendons par complètes est le fait que ces approches

permettent d'avoir tous les tests de détection possibles du système. Par conséquent, la meilleure approche serait celle qui permettrait de trouver tous les sous systèmes testables d'un système physique.

Plusieurs types de travaux nous ont inspirés pour construire nos sous systèmes testables : les travaux qui utilisent la théorie de l'élimination [Kapur et Lakshma, 1992], permettant de résoudre (3.3a) et (3.3b) en se basant sur la structure des relations élémentaires. Ils permettent de trouver toutes les relations mais peuvent conduire à des solutions non calculables qu'il faudra exclure. Les techniques d'élimination conçues sur les bases de Gröbner [Frisk, 2000] n'ont pas été exploitées car elles ne concernent que des systèmes polynomiaux. Nous avons étudié de près les travaux exploitant des modèles structurels ou l'approche par graphe bipartie qui est l'une des préoccupations de l'équipe Surveillance du Laboratoire d'Automatique et d'Informatique Industrielle de Lille [Declerck et Staroswiecki, 1991] [Guernez Jean, 1998]. L'analyse du graphe structurel d'un système permet d'identifier l'ensemble des relations de redondance qu'il est possible de construire dans le système, d'en déduire les sous systèmes surveillables, les défaillances susceptibles d'être détectées et isolées, leur degré de localisabilité, les variables calculables, les capteurs à implémenter pour répondre à des spécifications données. Les outils nécessaires pour comprendre l'approche des graphes bi-parties sont présentés [Staroswiecki, 2001] [Izadi-Zanamabadi et Staroswiecki, 2000]. L'approche par graphe bipartie comporte de nombreux intérêts. Elle s'adapte notamment à n'importe quel type de modèle car elle exprime seulement les liens entre les différentes variables sans s'intéresser aux variables internes (états) et à leur trajectoire. Cette approche est de plus particulièrement adaptée au diagnostic grâce à la décomposition canonique de système. Cependant, dans [Boutobza, 2003], il est montré que cette approche ne fournit pas tous les sous-systèmes testables possibles, et paradoxalement, plus il y a de capteurs dans le système, moins de relations de redondance analytiques sont trouvées.

Une autre méthode générale de conception automatique des sous systèmes testables nous a inspirés. Cette méthode a été étudiée au Laboratoire d'Automatique de Grenoble dans [Ploix et Follot, 2001] et améliorée dans [Ploix et al, 2005]. Cette méthode se base sur des règles d'élimination des variables inconnues. En effet, la procédure d'élimination a été conçue dans l'objectif de réduire autant que possible le nombre de calculs. Les sous systèmes testables fournis sont très utiles parce qu'ils mènent aux tables de signature ; ils permettent de prévoir les meilleures performances possibles et réalisables d'une procédure de diagnostic. Parce que l'algorithme proposé est basé sur une analyse structurelle du procédé, il s'applique aussi aux systèmes non linéaires. La surestimation de solutions peut être réduite en prenant en compte la calculabilité dans les contraintes. Ces résultats sont utiles à l'analyse diagnostique logique car elle associe à chaque RRA, un support en terme d'hypothèses.

L'application de la procédure d'élimination sur le système du bioprocédé conduit aux sous systèmes testables résumés par la table de signature 3.2. Cette table de signatures fait le lien entre un ensemble de sous systèmes testables (SST_i) et l'ensemble de modèles élémentaires (M_k).

$Hi(M_i)$ ➔	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	H_{10}	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}	H_{16}	H_{17}	H_{18}	H_{19}	H_{20}	H_{21}	H_{22}	H_{23}
SST_1	1																1					1	
SST_2			1														1	1					
SST_3							1												1	1			
SST_4			1							1	1												
SST_5	1															1	1					1	
SST_6				1				1										1					1
SST_7						1	1					1		1	1					1			
SST_8				1		1												1	1	1			1
SST_9	1	1							1							1					1	1	
SST_{10}				1		1		1			1	1			1			1	1	1			1
SST_{11}	1															1						1	
SST_{12}	1															1	1						
SST_{13}			1	1				1									1						1
SST_{14}				1		1	1											1	1				1
SST_{15}				1		1	1											1		1			1
SST_{16}	1	1							1	1						1					1		
SST_{17}	1	1	1						1		1					1					1	1	
SST_{18}				1		1	1	1										1	1				
SST_{19}				1	1	1	1				1	1			1			1	1				1
SST_{20}			1	1		1	1			1		1			1			1	1				1
SST_{21}				1		1	1				1	1			1			1	1				
SST_{22}				1		1	1				1	1		1				1	1				1
SST_{23}				1		1	1				1			1	1			1	1				1

Table 3.2 - Les sous systèmes testables retenus pour le bioprocédé

De même, l'algorithme de génération automatique des sous systèmes testables a été appliqué sur la description du système de la machine à courant continu. Les SST décrites dans le tableau 3.3 ont été obtenues.

	ME_1	ME_2	ME_3	ME_4	ME_5	ME_6	ME_7	ME_8	ME_9	ME_{10}	ME_{11}
SST_1	1						1		1		
SST_2		1						1	1	1	1
SST_3			1					1	1	1	1
SST_4				1	1			1		1	1
SST_5				1		1		1		1	1
SST_6	1	1					1	1		1	1
SST_7	1		1				1	1		1	1
SST_8		1		1	1			1	1		1
SST_9			1	1	1			1	1		1
SST_{10}		1		1		1		1	1		1
SST_{11}			1	1		1		1	1		1
SST_{12}		1		1	1			1	1	1	
SST_{13}			1	1	1			1	1	1	
SST_{14}		1		1		1		1	1	1	
SST_{15}			1	1		1		1	1	1	
SST_{16}	1	1		1	1		1	1			1
SST_{17}	1		1	1	1		1	1			1
SST_{18}	1	1		1		1	1	1			1
SST_{19}	1		1	1		1	1	1			1
SST_{20}	1	1		1	1		1	1		1	
SST_{21}	1		1	1	1		1	1		1	
SST_{22}	1	1		1		1	1	1		1	
SST_{23}	1		1	1		1	1	1		1	

Tableau 3.3 – Composition des SST de la machine MCC

III.5. TRAITEMENT DES SYMPTOMES POUR L'ANALYSE DIAGNOSTIQUE

Il existe différentes techniques de prise de décision lors de l'évaluation de résidus. Il peut s'agir d'une prise de décision basée sur un simple test logique (comparaison à un seuil), d'une prise de décision statistique (test de Bayes), de technique de reconnaissance des formes ou de technique de prises de décision basée sur la logique floue. La principale difficulté lors de l'évaluation des résidus est que les résidus contiennent souvent des phénomènes normaux non modélisés a priori. Lorsque le résultat de la comparaison à un seuil est booléen (alarme ou non), la décision peut être complètement différente suivant que le résidu est juste au-dessous ou juste au-dessus du seuil qui a été fixé. L'utilisation des seuils conduit à des difficultés lors de la prise de décision : il est pas possible de représenter un doute. En revanche, l'utilisation de la logique floue [Zadeh, 1965] permet de construire une décision plus progressive [Evsukoff, 1998] [Combastel, 2000].

La théorie des ensembles flous généralise celle des ensembles classiques. En effet, la fonction d'appartenance {0, 1} de la théorie des ensembles classiques (un élément appartient ou n'appartient pas à l'ensemble) devient un cas particulier de la fonction d'appartenance, étendue à l'intervalle [0, 1], de la théorie des ensembles flous. La fonction d'appartenance est essentielle et permet de les différencier des ensembles habituels [Zadeh, 1965].

Un état de l'art sur les méthodes d'analyse des résidus a été fait dans [Leonhardt et Ayoubi, 1997]. Les méthodes les plus utilisées ou les plus étudiées sont regroupées dans trois catégories [Leonhardt et Ayoubi 1997] :

1) Classification (e.g. reconnaissance des formes).
2) Inférence floue (i.e. basé sur des règles linguistiques)
3) Combinaison des deux méthodes précédentes (e.g. méthodes neuro-floues ou adaptatives).

Pour réduire le taux de fausses alarmes, Frank [Frank et al, 1999] propose deux façons différentes d'appliquer la logique floue pour l'évaluation des résidus. La première en faisant usage des seuils floues adaptatifs et la deuxième utilise la déduction floue avec l'assistance d'un opérateur humain pour la prise de décision. Les seuils flous adaptatifs permettent d'éviter les fausses alarmes et de détecter le défaut. Il utilise des seuils adaptatifs qui sont définis en terme de règle floue afin de détecter même le plus petit défaut sur le système. [Dalton et al, 1996] propose une application de la logique floue pour l'analyse des résidus générés en utilisant un observateur d'état. La méthode d'analyse des résidus étudiée par Dalton est basée sur le travail de Frank et Kiupel [Frank et Kiupel, 1993], qui propose l'usage de la logique floue pour une pré-évaluation des résidus avec la prise de décision finale sur l'état du système transmise à l'opérateur.

Notre approche présentée dans [Touaf et al, 2004 (d)] se différencie des approches précédentes par le fait qu'elle ne concerne pas l'évaluation des résidus mais l'analyse diagnostique. Elle considère qu'un test de détection peut produire un résultat flou (appartenance à alarme et à valide, par exemple) et se propose d'exploiter ces résultats lors de l'analyse diagnostique. L'objectif est de transposer les analyses diagnostiques basées sur la logique des prédicats vers la logique floue afin

d'exploiter les décisions douteuses en conduisant à ce que l'on pourrait appeler des pré-diagnostics, c'est-à-dire des diagnostics qui ne sont pas encore avérés.

III.5.1. Définition d'un test de détection

Un sous système testable (SST) n'est pas un test de détection en soit : il détermine simplement la cible et les hypothèses sous-jacentes à un test. Un même sous-système SS_k peut faire l'objet de plusieurs tests de détection s'appuyant sur des algorithmes différents.

La fonction de test a pour but de vérifier la vérité de la conjonction d'hypothèses (H_k avec $k \in \varphi$) d'un sous système testable SST_i en utilisant les relations élémentaires qu'il implique. La fonction de test est dite valide si est seulement si une contrainte construite à partir $\bigcup_k \{V_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi' \subseteq \varphi\}$ et de certaines relations élémentaires de $\bigcup_k \{R_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}$ est satisfaite. La fonction de test est une fonction qui ne contient plus que des valeurs connues, c'est-à-dire que toutes les variables doivent être mesurées.

Soit un test de détection t_k admettant une fonction de vérité $\top(t_k)$ à valeur soit dans {*vrai, faux*} soit dans [0,1] s'il s'agit d'un ensemble d'appartenance à faux. Le test t_k est défini par le système suivant :

$$t_k : \begin{cases} \neg\sigma_{t_k} \\ \upsilon_{t_k} \end{cases} \qquad (3.4)$$

où :

- $\neg\sigma_{t_k}$ représente le résultat du test de détection. Il s'agit d'une contrainte construite à partir des contraintes de $\bigcup_k \{R_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi\}$, qui ne contient plus que des valeurs connues $\neg\sigma_{t_k} = \neg\sigma_{t_k}(\tilde{\gamma})$. La vérité de cette contrainte n'est pas forcément booléenne. Un test de détection peut conduire à une décision incertaine dans le cas où la valeur de $\neg\sigma_{t_k}$ appartient à l'intervalle [0, 1]. $\neg\sigma_{t_k}$ =1 représente que le test détecte une alarme d'une manière sûre et $\neg\sigma_{t_k}$ =0 dans le cas contraire, c'est-à-dire, que le test est cohérent avec les observations. $\neg\sigma_{t_k}$ = 0.5 représente le cas du doute dans la prise de décision (impossibilité de prendre une décision entre une consistance ou une inconsistance). Entre les deux bornes on peut avoir un test qui soit vrai à 20% ou par complémentarité faux à 80%.
- υ_{t_k} représente la fonction de validité du test de détection. Il s'agit de l'union des ensembles de contraintes de validités : $\bigcup_k \{V_k(\chi_k, \tilde{\gamma}_k) : k \in \varphi' \subseteq \varphi\}$ avec certaines contraintes comportementales. De la même façon que le résultat du test $\neg\sigma_{t_k}$, la vérité de la fonction de validité peut aussi prendre ses valeurs dans un intervalle [0, 1].

La fonction t_k est appelée test de détection si et seulement si les conditions suivantes sont vérifiées :

$$\forall \tilde{\gamma}_k \,/\, \neg\sigma_{t_k}(\chi_k, \tilde{\gamma}_k) \wedge \upsilon_{t_k}(\chi_k, \tilde{\gamma}_k) \Rightarrow \neg\hat{\mathcal{H}}_{t_k} \text{ où } \hat{\mathcal{H}}_{t_k} = \bigwedge_{H_k \in \mathcal{H}_{t_k}} H_k(C_j) \qquad (3.5\text{ a})$$

$$\forall \tilde{\gamma}_k \,/\, \neg\upsilon_{t_k}(\chi_k, \tilde{\gamma}_k) \Rightarrow \text{ pas d'information sur } \hat{\mathcal{H}}_{t_k} \qquad (3.5\text{ b})$$

La relation (3.5a) est la définition de base d'un test de détection. Cette définition n'exonère pas du fait de l'implication. En effet, si le test est faux et valide alors un ou plusieurs des composants sur lesquels il se base, sont forcément dans un état anormal. En revanche, si le test est vrai et valide, cela n'implique pas forcément que les composants sont dans un état normal. La relation (3.5b) signifie que si le test de détection est invalide alors les hypothèses reliées au test peuvent être soit vraies, soit fausses.

III.5.2. Niveau de confiance d'un symptôme et prise de décision

Les procédures classiques de décision prennent en compte les incertitudes par l'intermédiaire des seuils (fixes) définis sur la différence entre comportement réel et de référence. La difficulté est alors de déterminer la valeur du seuil. Si, elle est trop petite, il peut y avoir des fausses alarmes. Si le seuil est trop grand, on aboutit à des manques à la détection.

Les tests de détection génèrent des symptômes qui contiennent des résultats bruts. En effet, chaque symptôme contient le résultat du test, c'est-à-dire, la décision du test noté $\neg\sigma_{t_k}$ et la validité du modèle du test noté υ_{t_k}.

La décision du test $\neg\sigma_{t_k}$ ainsi que le domaine de validité du test υ_{t_k} peuvent prendre leurs valeurs dans l'intervalle [0, 1]. Si le résultat d'un test de détection est égal à 1 ($\neg\sigma_{t_k}$ = 1) alors le test est inconsistant sans qu'aucun doute ne soit permis, sinon, si le résultat du test de détection est égal à 0 ($\neg\sigma_{t_k}$ = 0), alors le test est obligatoirement consistant. En revanche, si le résultat du test est égal à 0.5 ($\neg\sigma_{t_k}$ = 0.5) alors la décision est peu fiable, ce qui nous amène à des alarmes non détectées (impossible de décider entre une consistance et une inconsistance).

La validité possède aussi un degré de vérité. En effet, un test de détection donné repose sur des modèles élémentaires avec lesquels il est construit. Ces modèles peuvent être valides ou non valides pour un ensemble de valeurs courantes de leurs variables.

Cependant, ces résultats bruts ne sont pas directement utilisés pour l'analyse diagnostique. Tout d'abord, pour chaque test de détection, le résultat de la décision $\neg\sigma_{t_k}$ et la validité du test υ_{t_k} sont fusionnés ensemble afin de pouvoir générer la décision finale du test de détection. Cette valeur calculée correspond au degré de vérité, noté par *T(.)* du test de détection lié aux hypothèses $\hat{\mathcal{H}}_{t_k}$ définies dans (3.4) et (3.5a). La fusion des résultats $\neg\sigma_{t_k}$ et υ_{t_k} a été faite en utilisant les définitions du test de détection (3.5a) et (3.5b) tout en utilisant la déduction suivante : (A⇒B) ⇔ (¬A∨B) comme suit :

$$\begin{array}{l} \rightarrow \neg\sigma_{t_k}\left(\chi_k,\tilde{\gamma}_k\right)\wedge\upsilon_{t_k}\left(\chi_k,\tilde{\gamma}_k\right)\Rightarrow\neg\hat{\mathcal{H}}_{t_k} \\ \hline \rightarrow \neg\left(\neg\sigma_{t_k}\left(\chi_k,\tilde{\gamma}_k\right)\wedge\upsilon\left(\chi_k,\tilde{\gamma}_k\right)\right)\vee\neg\hat{\mathcal{H}}_{t_k} \\ \hline \rightarrow \neg\left(\neg\sigma_{t_k}\left(\chi_k,\tilde{\gamma}_k\right)\right)\vee\neg\upsilon_{t_k}\left(\chi_k,\tilde{\gamma}_k\right)\vee\neg\hat{\mathcal{H}}_{t_k} \end{array} \qquad (3.6)$$

La table 3.4 représente la démonstration par table de vérité de la déduction (A⇒B) ⇔ (¬A∨B) :

A	**B**	**A⇒B**	**¬A**	**¬A∨B**
0	0	1	1	1
0	1	1	1	1
1	0	0	0	0
1	1	1	0	1

Table 3.4-Table de vérité

Nous allons calculer le degré de vérité de la dernière proposition notée *P* tel que $P = \neg\left(\neg\sigma_{t_k}(\chi_k, \tilde{\gamma}_k)\right) \vee \neg\upsilon_{t_k}(\chi_k, \tilde{\gamma}_k) \vee \neg\hat{\mathcal{H}}_{t_k}$. Puisque la proposition *P* est vraie nous écrivons que *T(P)=1*. Ce calcul se fait en utilisant les principes de calcul de la logique floue.

Il existe toute une famille d'opérateurs susceptibles d'être utilisés pour décrire une intersection (resp. une union) : les T-normes (resp. les T-conormes). La fonction min est la plus grande des T-normes et la fonction max, la plus petite des T-conormes (table 3.4). Les moyennes servent à décrire une agrégation du type « compromis ».

	T-norme	T-conorme
Zadeh(z)	$\mu_A \overset{Z}{\wedge} \mu_B = \min(\mu_A, \mu_B)$	$\mu_A \overset{Z}{\vee} \mu_B = \max(\mu_A, \mu_B)$
Lukasiewicz (l)	$\mu_A \overset{l}{\wedge} \mu_B = \max(0, \mu_A + \mu_B - 1)$	$\mu_A \overset{l}{\vee} \mu_B = \min(0, \mu_A + \mu_B)$
Probabiliste (p)	$\mu_A \overset{p}{\wedge} \mu_B = \mu_A.\mu_B$	$\mu_A \overset{p}{\vee} \mu_B = \mu_A + \mu_B - \mu_A.\mu_B$

Négation floue	Moyenne
$Non(\mu_A) = 1 - \mu_A$	$Moyenne(\mu_A, \mu_B) = (\mu_A + \mu_B)/2$

Table 3.4 - Quelques opérateurs floues usuels

L'addition remplace la disjonction :

$$\begin{array}{l} \rightarrow T\left(\neg\left(\neg\sigma_{t_k}(\chi_k, \tilde{\gamma}_k)\right) \vee \neg\upsilon_{t_k}(\chi_k, \tilde{\gamma}_k) \vee \neg\hat{\mathcal{H}}_{t_k}\right) = 1 \\ \hline \rightarrow \min\left(1, (1-\mu_{\neg\sigma}) + (1-\mu_{\upsilon}) + \mu_{\neg\hat{\mathcal{H}}}\right) = 1 \\ \hline \rightarrow \min\left(1, 2 - \mu_{\neg\sigma} - \mu_{\upsilon} + \mu_{\neg\hat{\mathcal{H}}}\right) = 1 \\ \hline \rightarrow \mu_{\neg\hat{\mathcal{H}}} \geq \mu_{\neg\sigma} + \mu_{\upsilon} - 1 \end{array} \qquad (3.7)$$

sachant que : $\mu_{\neg\sigma} = T\left(\neg\sigma_{t_k}\right)$, $\mu_{\upsilon} = T\left(\upsilon_{t_k}\right)$ et $\mu_{\neg\hat{\mathcal{H}}} = T\left(\neg\hat{\mathcal{H}}_{t_k}\right)$

Par ailleurs, on déduit de la relation (3.5b) la propriété suivante :

$$\mu_{\upsilon} = 0 \Rightarrow \mu_{\neg\hat{\mathcal{H}}} = 0.5 \qquad (3.8)$$

La propriété (3.8) s'exprime littéralement ainsi : si la fonction d'appartenance de la validité du modèle est nulle alors ceci implique qu'on ne peut rien conclure (impossible de décider entre une consistance et une inconsistance) et par conséquent la décision est égale à 0.5.

Par conséquent, la fonction qui permet de fusionner entre la décision du test et sa validité doit satisfaire les propriétés (3.7) et (3.8). Néanmoins, il existe plusieurs solutions possibles et réalisables pour trouver cette fonction. Il serait pratique que la fonction choisie tienne compte de toutes les approches d'analyse diagnostique. Elle doit inclure l'approche structurelle de la communauté FDI. L'approche FDI exonère les hypothèses du test de détection qui ne manifeste aucune alarme. La fonction de fusion qui reste valide dans le cas d'une exonération serait plus intéressante. Cependant, l'existence d'une telle fonction de fusion doit être vérifiée. Si elle existe, alors la proposition suivante doit donc être considérée comme vraie.

$$\frac{\rightarrow \forall \tilde{x}_{t_k} / \sigma_{t_k}\left(\tilde{x}_{t_k}\right) \wedge \upsilon_{t_k}\left(\tilde{x}_{t_k}\right) \Rightarrow \hat{\mathcal{H}}_{t_k}}{\rightarrow \forall \tilde{x}_{t_k} / \neg\sigma_{t_k}\left(\tilde{x}_{t_k}\right) \vee \neg\upsilon_{t_k}\left(\tilde{x}_{t_k}\right) \vee \hat{\mathcal{H}}_{t_k}}$$

La vérité de la proposition Q, notée T(Q), tel que $Q = \forall \tilde{x}_{t_k} / \neg\sigma_{t_k}\left(\chi_k, \tilde{\gamma}_k\right) \vee \neg\upsilon_{t_k}\left(\chi_k, \tilde{\gamma}_k\right) \vee \hat{\mathcal{H}}_{t_k}$ peut être écrite comme suit :

$$\begin{array}{l} \rightarrow T\left(\neg\sigma_{t_k}\left(\chi_k, \tilde{\gamma}_k\right) \vee \neg\upsilon_{t_k}\left(\chi_k, \tilde{\gamma}_k\right) \vee \hat{\mathcal{H}}_{t_k}\right) = 1 \\ \hline \rightarrow \min\left(1, \mu_{\neg\sigma} + \left(1-\mu_{\upsilon}\right) + \left(1-\mu_{\neg\hat{\mathcal{H}}}\right)\right) = 1 \\ \hline \rightarrow \min\left(1, 2+\mu_{\neg\sigma} - \mu_{\upsilon} - \mu_{\neg\hat{\mathcal{H}}}\right) = 1 \\ \hline \rightarrow \mu_{\neg\hat{\mathcal{H}}} \leq 1+\mu_{\neg\sigma} - \mu_{\upsilon} \end{array} \qquad (3.9)$$

La fonction de fusion que nous avons choisie doit satisfaire les trois contraintes (3.7), (3.8) et (3.9). Par conséquent, la région possible pour cette fonction de fusion qui satisfait les contraintes correspond à la surface hachurée donnée par la figure 3.8.

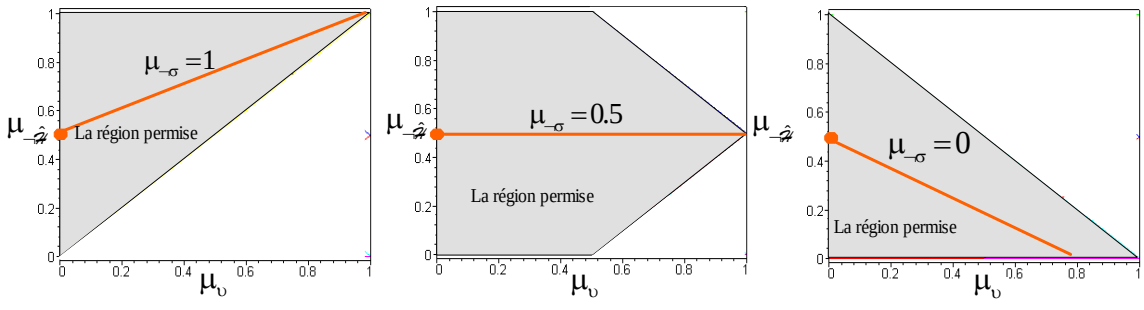


Figure 3.8 - Région possible pour la fonction de fusion

La fonction de fusion la plus simple parmi l'ensemble de fonctions possibles est une fonction linéaire représentée par une droite sur la figure 3.8. Cette fonction est donnée par la relation ci-dessous :

$$\mu_{\neg\hat{\mathcal{H}}} = \frac{1+\left(2\mu_{\neg\sigma}-1\right)\mu_{\upsilon}}{2} \qquad (3.10)$$

$\mu_{\neg\sigma}$ représente le degré d'appartenance de la décision d'un test de détection à une inconsistance. Ces décisions peuvent être obtenues grâce à 2 seuils sur les résidus (cf. figure 3.9).

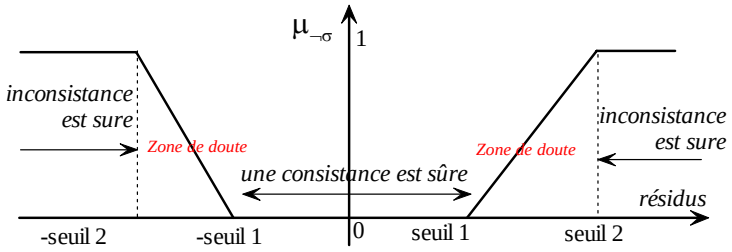


Figure 3.9 – La prise de décision incertaine

Le seuil 1 et le seuil 2 sont fixés de telle sorte que si le résidu produit par le test de détection se trouve dans l'intervalle [-seuil1 seuil1] cela représente qu'il n'y a pas d'alarme d'une manière sûre. Autrement dit, le test est consistant. Si le résidu de la détection est supérieur à (seuil2) ou inférieure à (-seuil2) alors la décision prise par le test est la détection d'alarme. Ce qui revient à dire que le test est inconsistant d'une manière sûre. En revanche, si le résidu est entre [seuil1 seuil2] ou [-seuil2 –seuil1] alors la décision est incertaine. En effet, cette zone représente la zone de doute où il n'est pas possible de décider entre une consistance ou une inconsistance.

La fonction de fusion de la relation (3.9) illustre que le niveau de confiance $\mu_{\neg\hat{\sigma}}$ sera égal à la valeur de décision $\mu_{\neg\sigma}$ si la validité est égale à 1. Quand la validité du modèle du test de détection décroît alors le niveau de confiance tend vers 0.5, c'est-à-dire vers plus d'incertitude.

III.6. CONCLUSION

A travers ce chapitre un nouveau formalisme de modélisation pour le diagnostic des systèmes complexes a été proposé. Une bonne description du système à diagnostiquer permettant d'exploiter toute la connaissance disponible sur un système et son environnement est capital. La modélisation d'un système ne peut se réduire à une relation descriptive de son comportement. En diagnostic, il est important de connaître les états des composants testés par une relation. Chaque relation doit être décrite par une hypothèse sur l'état supposé du composant associé. Par ailleurs, un modèle de comportement n'est pas universel ; il n'est généralement valide que dans certaines plages de fonctionnement, décrites par des conditions de validité. Nous avons montré qu'un test de détection repose sur un modèle bon fonctionnement ou sur un modèle de mauvais fonctionnement.

Tous les tests de détection ne conduisent pas toujours à des décisions sûres. L'incertitude se caractérise par l'incapacité de savoir si un test de détection est vrai ou faux. Les tests de détection peuvent conduire à des décisions comprises entre 0 et 1 où 1 représente une alarme sûre et 0 représente l'absence d'une alarme d'une manière sûre. Nous avons montré qu'il était possible d'appréhender des incertitudes de décision au niveau des tests de détection afin de les exploiter durant la phase d'analyse diagnostique. Le formalisme proposé transpose la logique de l'analyse diagnostique en logique floue. Ce formalisme permet de fusionner la décision du test avec celle de sa validité.

Chapitre 4

L’analyse diagnostique logique proposée

IV.1. INTRODUCTION

Les tests de détection peuvent être réalisés en utilisant des algorithmes différents (détection à base de signal, détection utilisant des graphes causaux, observateurs d'état, relations de parité, détection à base des réseaux de neurones,...). L'algorithme d'analyse diagnostique doit donc être capable d'appréhender des tests de détection hétérogènes qui doivent être décrits de la même façon pour pouvoir être exploités par un algorithme d'analyse diagnostique. C'est le rôle du sous-système testable. L'analyse diagnostique doit s'appuyer non pas sur les algorithmes de détection eux-mêmes, mais sur les sous-systèmes testables qui constituent une méta-description d'un test de détection.

Les partenaires industriels du projet MAGIC considèrent qu'un système de diagnostic peut, dans un contexte donné, conduire à un diagnostique erroné mais il faut que si ce contexte se représente, le système de diagnostic ne reproduise pas la même erreur. Cela induit donc des propriétés de garantie de la part de l'analyse diagnostic afin d'être en mesure de reconnaître les tests de détection ayant produit des symptômes erronés. Si les sources d'erreur peuvent provenir à la fois des tests de détection et à la fois de l'analyse diagnostique, il n'est plus possible de remonter à la cause d'une erreur. Ce chapitre propose une méthode d'analyse diagnostique inspirée de l'algorithme de Reiter initialement dédiée aux systèmes statiques qui soit capable d'analyser formellement les symptômes issus de tests de détection dynamiques et hétérogènes.

Différents tests peuvent vérifier des états de composants communs. Deux tests différents peuvent s'intéresser à un même sous système testable. L'analyse diagnostique doit alors être capable d'appréhender ce type de situation.

Limiter la décision d'un test de détection à deux résultats possibles (alarme ou non) ne permet pas d'exprimer une incertitude de décision du type : une anomalie est soupçonnée sans que cela soit certain. Les tests de détection mis en oeuvre dans le cadre du projet MAGIC fournissent des décisions comprises entre 0 et 1 où 1 représente une inconsistance sûre et 0 une consistance sûre. L'analyse diagnostic doit pouvoir exploiter ces résultats.

De plus, dans le cadre des systèmes industriels tels que ceux que nous avons abordés dans le projet MAGIC, pour des raisons liées à des capacités de calcul, tous les tests ne peuvent pas être actifs en permanence. L'objectif consiste à déterminer quels sont les tests de détection les plus intéressants à déclencher.

Dans ce chapitre l'analyse diagnostique proposée ne prend pas en compte les modèles de fautes. Elle considère que les modèles de bon fonctionnement.

IV.2. PROCEDURES DE CALCUL DES DIAGNOSTICS MINIMAUX

Le raisonnement diagnostique analyse les symptômes disponibles fournis par les tests de détection afin de déterminer les diagnostics. [Reiter, 1987] a proposé une théorie logique du diagnostic connue sous le nom de diagnostic à base de consistance, qui a été plus tard étendue et formalisée dans [De Kleer et Williams, 1987]. La théorie logique du diagnostic repose sur le

concept du conflit expliqué au chapitre 3. Pour les approches DX, un conflit décrit un ensemble de composants dont un au moins est nécessairement défaillant pour que les observations soient consistantes avec le modèle du système. Par ailleurs, puisque notre approche intègre des modèles de mauvais comportement alors un conflit ne peut plus être réduit à une liste de composant dans un état normal.

IV.2.1. Le principe de la méthode

La recherche des diagnostics minimaux se fait via l'algorithme des HS-Tree proposé par Reiter [Reiter, 1987] et corrigé par [Greiner al.,1989], en remplaçant les composants par des hypothèses sur les composants. Nous avons adopté l'algorithme de Reiter qui se déroule via un arbre (cf. chapitre 2). La méthode que nous proposons transpose le même raisonnement sous une forme matricielle. Les lignes de la matrice de signatures représentent les différents tests effectués sur le système physique. Les colonnes représentent les hypothèses sur les composants qui constituent le système physique. Chaque modèle élémentaire M_i contient une hypothèse sur l'état modélisé d'un composant.

Les tests de détection valides génèrent des symptômes. Le résultat de chacun des tests conduit à des résultats booléens (vrai ou faux). L'analyse diagnostique proposée est logique et elle est capable d'exploiter les symptômes disponibles afin de déduire un diagnostic en terme de composant défaillant.

Si un test T_i correspondant à un sous-système SST_i s'avère faux, toutes les hypothèses sur les composants du sous-système SST_i associés ne peuvent pas être vérifiées à la fois. Plus précisément, en appelant $H_1 \wedge H_2 \wedge \ldots \wedge H_i$, les hypothèses associées au test T_i construit à partir du sous-système SST_i, il convient logiquement de dire que si T_i conduit à un résultat faux tout en étant valide (condition de validité vérifiée) cela signifie qu'au moins une des hypothèses H_i est fausse.

Soit un système physique $\sum$ décrit par neuf modèles élémentaires $M_1,\ldots,M_9$. Pour chaque modèle élémentaire on associe une hypothèse sur l'état du composant modélisé $H_1,\ldots,H_9$. On suppose que les hypothèses du système peuvent être vérifiées via trois tests de détection SST_1, SST_2 et SST_3. La table 4.1 représente la matrice des signatures pour notre système. Une matrice des signatures sert de base au raisonnement. Chaque ligne correspond à un test et chaque colonne à une hypothèse.

	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9
SST_1	1	1	1	1					
SST_2	1	1						1	1
SST_3	1				1	1	1		

Tableau 4.1- Matrice des signatures

Un test de détection inconsistant implique que, s'il est valide, la conjonction d'hypothèses sur lesquelles il se fonde est nécessairement fausse. Si un test de détection associé au $n^{ième}$ SST, c'est à dire au $n^{ième}$ test de détection, repose sur l'hypothèse suivante $\bigwedge_{i\in\varphi} H_i$ et qu'il révèle une inconsistance alors l'hypothèse du test sera fausse et donc le conflit $\mathscr{C}_n$ s'écrira :

$$\mathcal{C}_n = \left\{ \neg H_i \,/ \bigvee_{i \in \varphi} \neg H_i \right\} \tag{4.1}$$

Tel que :

- $\neg H_i$ représente la négation de l'hypothèse sur le composant C_i .
- $\bigvee_{i \in \varphi} \neg H_i$ représente la disjonction de la négation des hypothèses sur l'état des composants vérifiés par le test.
- $\mathcal{C}_n$ représente donc l'ensemble des hypothèses dont une au moins est vérifiée par le test.

Il faut commencer par recenser les conflits en recherchant tous les conflits minimaux possibles. Cela revient à remonter des symptômes aux éléments physiques du système à diagnostiquer. La matrice de signature (table 4.1) donne les différentes hypothèses sur l'état des composants qui interviennent dans les trois tests (SST_1, SST_2 et SST_3). Nous supposons que deux d'entre eux se révèlent inconsistants (SST_1, SST_2) et le troisième (SST_3) conduit à une consistance.

Les modèles élémentaires M_1, M_2, M_3, et M_4 constituent le support du test SST_1. Par conséquent, l'hypothèse du test de détection est donnée par la conjonction des hypothèses associées aux différents modèles élémentaires impliqués :

$$H_1 \wedge H_2 \wedge H_3 \wedge H_4 \tag{4.2}$$

Comme le test est inconsistant alors l'hypothèse du test sera fausse et donc le conflit $\mathcal{C}_1$ s'écrira logiquement comme une négation de l'hypothèse du test :

$$\begin{aligned} &\neg\left(H_1 \wedge H_2 \wedge H_3 \wedge H_4\right) \\ &\neg H_1 \vee \neg H_2 \vee \neg H_3 \vee \neg H_4 \end{aligned} \tag{4.3}$$

Les conflits sont donc les suivants :

$$\mathcal{C}_1 = \left\{ \neg H_1, \neg H_2, \neg H_3, \neg H_4 \right\} \tag{4.4}$$

De la même manière pour le test de détection SST_2 les conflits sont:

$$\mathcal{C}_2 = \left\{ \neg H_1, \neg H_2, \neg H_8, \neg H_9 \right\}$$

Il est possible d'analyser un ensemble de conflits révélés par les tests de consistance pour en déduire des diagnostics. Un diagnostic *D* expliquant un ensemble de conflit $\{\mathcal{C}_n / n \in \Omega\}$ est un ensemble d'hypothèses $\mathcal{D} = \{\neg H_i\}$ qui satisfait :

$$\forall n \in \Omega, \mathcal{D} \cap \mathcal{C}_n \neq \varnothing \tag{4.5}$$

Du fait du grand nombre de diagnostics possibles, on ne s'intéresse généralement qu'aux diagnostics minimaux. Un diagnostic *D* est minimal si et seulement si :

$$\nexists \mathcal{D}' \subset \mathcal{D} \,/\, \forall n \in \Omega, \mathcal{D}' \cap \mathcal{C}_n \neq \varnothing \tag{4.6}$$

Il faut alors générer des candidats minimaux consistant avec l'ensemble des conflits ; autrement dit, déterminer quelles sont les hypothèses fausses minimum qui expliquent les conflits observés. Pour Reiter, il s'agit de déterminer les constituants du système qui, en les supposant défaillants, expliqueraient les contradictions entre les éléments observés et corrects. Le tableau suivant illustre notre raisonnement matriciel.

	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	$H_3 \wedge H_8$	$H_3 \wedge H_9$	$H_4 \wedge H_8$	$H_4 \wedge H_8$
SST_1	1	1	1	1						1	1	1	1
SST_2	1	1						1	1	1	1	1	1

Table 4.2 – Diagnostics minimaux

Les tests SST_1 et SST_2 conduisent à des résultats faux. Si l'hypothèse H_1 était fausse, alors cela expliquerait les 2 résultats faux obtenus. Il en va de même pour H_2. H_1 et H_2 sont donc des diagnostics minimaux (cf. formule 4.6). Dans cet exemple, il n'y a pas d'autre hypothèse qui, à elle seule, pourrait expliquer les symptômes observés. Néanmoins, deux hypothèses peuvent être fausses en même temps. Par exemple, si H_3 et H_8 étaient simultanément fausses alors cela pourrait expliquer les symptômes obtenus. H_3 et H_8 simultanément faux est donc un diagnostic. En utilisant les notations logiques ^ pour symboliser le 'et' logique et '¬' pour la négation, les diagnostics minimaux correspondants à cet exemple sont donnés par : $\{\neg H_1, \neg H_2, \neg H_3 \wedge \neg H_8, \neg H_3 \wedge \neg H_9, \neg H_4 \wedge \neg H_8, \neg H_4 \wedge \neg H_9\}$.

La construction de la matrice 4.2 se fait par combinaison deux à deux des colonnes de la matrice d'incidence 4.1. Les lignes sélectionnées sont les lignes qui correspondent aux tests ayant conduit à des conflits. La combinaison de deux colonnes est retenue si et seulement si elles nous rapportent de l'information. Par exemple, la combinaison de la colonne $M_3 =(1,0)^T$ avec la colonne $M_4=(1,0)^T$ ne donne pas plus d'information que la colonne M_3 ou M_4 seule. En effet, la somme des deux vecteurs précédents revient à appliquer un « ou logique » entre les colonnes. Ainsi la combinaison de M_3 et M_4 fournit le vecteur $(1,0)^T$. En revanche, la combinaison de $M_3 =(1,0)^T$ avec $M_8 =(0,1)^T$ ou bien $M_9 =(0,1)^T$ avec $M_3 =(1,0)^T$ nous permet de construire une nouvelle colonne $(1,1)^T$. Ce dernier vecteur veut dire que les deux hypothèses H_3 et H_8 simultanément fausses expliqueraient que SST_1 et SST_2 conduisent à une inconsistance.

Si un résultat logique faux a valeur de preuve, il n'en va pas de même d'un résultat vrai qui n'a qu'une valeur contextuelle. Un bon résultat ne prouve nullement l'absence de défaut. Il montre simplement que dans un contexte donné, les composants se comportent normalement. Pour cette raison, les résultats vrais ne peuvent pas être considérés de la même façon que les résultats faux. Pour exploiter l'information provenant des tests vrais, la compatibilité de chaque diagnostic avec ces tests va être étudiée. Un diagnostic est dit compatible avec un test ayant conduit à un résultat vrai si ce test ne contient pas d'hypothèses contradictoires avec le diagnostic. Supposons par exemple que le test SST_3 a conduit à un résultat vrai. Le diagnostic$\neg H_1$ est alors incompatible avec SST_3 qui contient H_1 ; il y a donc 0 test vrai compatible avec ce diagnostic. En revanche, $\neg H_2$ aura un test vrai compatible.

IV.3. LA PLAUSIBILITE CIRCONSTANCIELLE

L'analyse diagnostique logique ne prend pas en considération les tests de détection consistants. Cela s'explique par le fait que le résultat de ces tests est peu fiable car un résultat consistant ne prouve pas l'absence de défaut dans le système. Néanmoins, les tests de détection consistants peuvent fournir une information intéressante pour l'analyse diagnostique.

Même si le test correspondant au n^{ième} SST est vérifié et validé, du fait du principe de non-exonération, l'interprétation des hypothèses associées $\underset{i\in\varphi}{\wedge} H_i$ ne peut être globale. Néanmoins, un test vérifié indique que dans le contexte d'observation donné, toutes les hypothèses associées semblent satisfaites. Cette information sera exploitée à un niveau contextuel distinct du niveau formel décrit dans la partie précédente.

Lorsqu'un test associé au n^{ième} SST est vérifié alors toutes les hypothèses sur lesquelles il repose semblent contextuellement vraies : $\underset{i\in\varphi}{\wedge} H_i = \neg\left(\underset{i\in\varphi}{\vee} \neg H_i\right)$. Par conséquent, si le test associé au n^{ième} SST est vérifié, chacun des éléments de l'ensemble suivant est nécessairement faux : $\eta_n = \{\neg H_i / i \in \varphi\}$. Un diagnostic *D* sera contextuellement validé par le test associé au n^{ième} SST si : $\mathcal{D} \cap \eta_n = \varnothing$.

Les clauses (4.7) (4.8) et (4.9) définissent le principe de non-exonération

$$\exists \tilde{y} / \neg R_i(\tilde{y}) \wedge \left(\underset{i\in\varphi}{\wedge} V_i\right) \Rightarrow \neg\left(\underset{i\in\varphi}{\wedge} H_i\right) \qquad (4.7)$$

➥ Cette clause s'exprime littéralement ainsi : si le résultat du test de détection est faux tout en étant valide alors cela prouve que la conjonction d'hypothèses $\underset{i\in\varphi}{\wedge} H_i$ liées au test est fausse. La conclusion a valeur de preuve.

$$\forall \tilde{y} / R_i(\tilde{y}) \wedge \left(\underset{i\in\varphi}{\wedge} V_i\right) \Rightarrow \left(\underset{i\in\varphi}{\wedge} H_i\right) \qquad (4.8)$$

➥ Une consistance révèle seulement que les clauses $\underset{i\in\varphi}{\wedge} H_i$ sont satisfaites par rapport au contexte d'observation. Or, un test ne vérifie qu'une valeur de Y à chaque instant. Il est donc difficile dans la pratique de vérifier (4.8). Il en découlera que, pour un ensemble fini de valeur Y

$$\exists \tilde{y} \in Y / R_i(\tilde{y}) \wedge \left(\underset{i\in\varphi}{\wedge} V_i(\tilde{y})\right) \not\Rightarrow \left(\underset{i\in\varphi}{\wedge} H_i\right) \qquad (4.9)$$

Ce principe indique que les résultats cohérents et incohérents doivent être appréhendés différemment lors d'une analyse de symptômes pour le diagnostic. Il indique encore qu'un résultat inconsistant à un instant donné aura une valeur permanente tant que l'état du système ne changera pas de lui-même.

Par conséquent, une plausibilité circonstancielle (appelée aussi la mesure contextuelle) a été introduite dans [Ploix et Follot, 2001][Touaf et al, 2003 (b)]. En revanche, cette partie introduit une plausibilité circonstancielle légèrement différente. Cependant, une distance est calculée entre les

résultats des tests de détection et le symptôme attendu d'un défaut. Cela est possible du fait qu'il y a une équivalence stricte entre une signature de défaut et les résultats des tests de détection. Cette distance est basée sur un 1-norm[4]. Elle est logiquement définie comme suit :

$$\eta(d) = \frac{\sum_{t_i \in \mathcal{T}} T\left(\neg\exists H_i \in \left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i)\right) \Leftrightarrow \hat{\mathcal{H}}(t_i)\right)}{card(\mathcal{T})} \tag{4.10}$$

Ou encore, on peut l'exprimer autrement :

- Soit $\mathcal{T}$ l'espace de décision des tests de détection (t_1, t_2,…) ayant conduit à un résultat,
- Soit $\sigma(D_i)$ une signature d'un défaut exprimé dans l'espace $\mathcal{T}$,
- Soit σ le résultat des tests exprimé dans l'espace $\mathcal{T}$,

Alors $\eta(d) = \frac{\|\sigma(D_i) - \sigma\|_1}{card(\mathcal{T})}$ où $card(\mathcal{T})$ est un facteur de normalisation et qui correspond au nombre de tests ayant conduit à un résultat.

L'expression principale est reformulée par déduction logique :

$$\frac{\begin{array}{l}\rightarrow \neg\exists H_i \in \left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i)\right) \Leftrightarrow \hat{\mathcal{H}}(t_i) \\ \rightarrow \left(\exists H_i \in \left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i)\right) \vee \hat{\mathcal{H}}(t_i)\right) \wedge \ldots \quad \ldots \left(\neg\exists H_i \in \left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i)\right) \vee \neg\hat{\mathcal{H}}(t_i)\right)\end{array}}{\rightarrow \neg\left(\left(\neg\exists H_i \in \left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i)\right) \wedge \neg\hat{\mathcal{H}}(t_i)\right) \vee \ldots \left(\exists H_i \in \left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i)\right) \wedge \hat{\mathcal{H}}(t_i)\right)\right)} \tag{4.11}$$

Le degré de vérité dénotée par le prédicat *T(.)* de la dernière expression conduit à écrire :

$$\begin{array}{l} 1 - \left(T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) = \varnothing\right)\mu_{\neg\hat{\mathcal{H}}(t_i)} + T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) \neq \varnothing\right)\left(1 - \mu_{\neg\hat{\mathcal{H}}(t_i)}\right)\right) = \ldots \\ \ldots = \left(T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) = \varnothing\right)\left(1 - \mu_{\neg\hat{\mathcal{H}}(t_i)}\right) + T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) \neq \varnothing\right)\mu_{\neg\hat{\mathcal{H}}(t_i)}\right) \end{array} \tag{4.12}$$

Donc, la définition (4.10) conduit à la formule suivante qui nous permet de calculer la plausibilité circonstancielle $\eta(d)$ d'un diagnostic :

$$\boxed{\eta(d) = \frac{\sum_{t_i \in \mathcal{T}} \left(T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) = \varnothing\right)\left(1 - \mu_{\neg\hat{\mathcal{H}}(t_i)}\right) + T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) \neq \varnothing\right)\mu_{\neg\hat{\mathcal{H}}(t_i)}\right)}{card(\mathcal{T})}} \tag{4.13}$$

l'application de la formule (4.13) suivant l'algorithme suivant :

Si $T\left(\mathcal{H}(\neg d) \cap \mathcal{H}(t_i) = \varnothing\right)$ *alors* $\left(1 - \mu_{\neg\hat{\mathcal{H}}(t_i)}\right)$

Sinon $\mu_{\neg\hat{\mathcal{H}}(t_i)}$.

[4] 1-norm : $\| x_1 \| = \sum_i | x_i |$

Exemple

Soit un ensemble de tests de détection représentés par la table 4.3 ci-dessous. Chaque test t_i est en rapport avec une hypothèse $H_i(M_i)$ sur l'état du composant modélisé par M_i. $H_i(M_i)$ appartient à l'ensemble des défauts possibles. Par exemple, le test de détection t_1 vérifie si les composants C_1, C_3 et C_4 sont dans un état normal. Par conséquent, le test t_1 est sensible au défaut sur le composant C_1 mais complètement insensible au défaut sur le composant C_2.

	$H_1=\neg AN(C_1)$	$H_2=\neg AN(C_2)$	$H_3=\neg AN(C_3)$	$H_4=\neg AN(C_4)$
Test 1	1	0	1	1
Test 2	0	1	1	0
Test 3	1	0	1	0

Table 4.3- Exemple de table de signatures de défaut

Supposons maintenant que les tests de détection fournissent des décisions sûres, c'est-à-dire pas d'incertitude dans la décision. Le résultat est booléen (consistant ou inconsistant). Le test t_1 est inconsistant, t_2 est inconsistant, t_3 est consistant. Le résultat des tests s'écrit alors : $(t_1, t_2, t_3)=(1,1,0)^T$. Les résultats diagnostiques sont donnés par la table 4.4.

Diagnostics	AN(C_3)	AN(C_1)^AN(C_2)	AN(C_2)^ AN(C_4)
Plausibilité Formelle	1	1	1
Plausibilité Circonstancielle	0.66	0.66	1

Table 4.4 – Diagnostic logique

La plausibilité circonstancielle a été calculée suivant la formule (4.13). Pour illustrer le calcul, prenons l'exemple du diagnostic AN(C_3). Les trois tests induisent l'hypothèse ¬AN(C_3) (cf. table 4.3). La clause $T\left(\mathcal{H}\left(\neg d\right)\cap\mathcal{H}\left(t_i\right)\neq\varnothing\right)$ est satisfaite pour les trois tests. Dans ce cas, seul les termes contenant $\mu_{\neg\hat{\mathcal{H}}(t_i)}$ sont considérés. Cela revient à calculer la distance entre la signature du défaut AN(C_3) exprimé dans l'espace des tests $\tau_{i(i=1,2,3)}$ donnée par $(1,1,1)^T$ et le résultat des tests exprimé dans l'espace $\tau_{i(i=1,2,3)}$ donnée par $(1,1,0)^T$. Pour normaliser on divise la somme par le nombre de tests ayant conduit à un résultat.

$$\text{Plausibilité Circonstancielle (AN}(C_3))=\frac{1+1+0}{3}=\frac{2}{3}=0.66$$

IV.4. APPLICATION A LA MCC

Les 23 sous-systèmes testables conçus pour la MCC (voir chapitre 3) ne seront pas tous réalisés afin de tester l'état des composants, car cela a un coût. Plus on effectue de tests de détection simultanés et meilleure sera l'analyse diagnostique, mais le nombre de test à réaliser peut être considérable. La sélection des tests à réaliser relève de la stratégie de diagnostic que nous allons étudier plus en détails dans la section 4.8 de ce chapitre. En revanche, par un souci de clarté et pour être concis dans nos explications, nous avons considéré la situation où seuls les tests basés sur les sous-systèmes testables suivants (SST_1, SST_2, SST_3, SST_4, SST_{20}) ont fourni un résultat.

$H_i(M_i)\rightarrow$	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}
SST_1	1						1		1		
SST_2		1						1	1	1	1
SST_3			1					1	1	1	1
SST_4				1	1			1		1	1
SST_{20}	1	1		1	1		1	1		1	

Table 4.3 – Matrice des hypothèses pour des tests réalisés sur la MCC

La table 4.3 regroupe les différents tests de détection qui ont fourni un résultat. Nous avons pris l'exemple où les tests basés sur les SST_2, SST_3 et SST_{20} sont inconsistants tandis que les autres tests (SST_1, SST_4) sont consistants. La matrice des signatures (table 4.3) sert de base au raisonnement. Chaque ligne correspond à un test et chaque colonne à une hypothèse. Par conséquent, nous pouvons facilement retrouver les différentes hypothèses sur l'état des composants qui interviennent dans les tests inconsistants (i.e. SST_2, SST_3 et SST_{20}). Par exemple, les modèles élémentaires M_3, M_8, M_9, M_{10} et M_{11} (renvoi à la figure 3.1 du chapitre 3) composent le test SST_3. Par conséquent, l'hypothèse vérifiée par le test de détection est donnée par une conjonction d'hypothèses des différents modèles élémentaires. Il résulte que les conflits sont donc les suivants :

$$\mathscr{C}_2 = \{AN(C_2), AN(C_6), AN(C_7), AN(C_8), AN(C_9)\}$$
$$\mathscr{C}_3 = \{\neg CC(C_2), AN(C_6), AN(C_7), AN(C_8), AN(C_9)\}$$
$$\mathscr{C}_{20} = \{AN(C_1), AN(C_2), AN(C_3), AN(C_4), AN(C_5), AN(C_6), AN(C_8)\}$$

Le tableau suivant illustre notre raisonnement matriciel.

$H_i(M_i)\rightarrow$	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	H_{10}	H_{11}	$H_1 \wedge H_3$	$H_1 \wedge H_9$	$H_1 \wedge H_{11}$	$H_2 \wedge H_3$
SST_2		1						1	1	1	1		1	1	1
SST_3			1					1	1	1	1	1	1	1	1
SST_{20}	1	1		1	1		1	1		1		1	1	1	1

$H_2 \wedge H_9$	$H_2 \wedge H_{11}$	$H_3 \wedge H_4$	$H_3 \wedge H_5$	$H_3 \wedge H_7$	$H_4 \wedge H_9$	$H_4 \wedge H_{11}$	$H_5 \wedge H_9$	$H_5 \wedge H_{11}$	$H_7 \wedge H_9$	$H_7 \wedge H_{11}$
1	1				1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

$H_9 \wedge H_1 \wedge H_3$	$H_9 \wedge H_3 \wedge H_4$	$H_9 \wedge H_3 \wedge H_5$	$H_9 \wedge H_3 \wedge H_7$	$H_{11} \wedge H_1 \wedge H_3$	$H_{11} \wedge H_3 \wedge H_4$	$H_{11} \wedge H_3 \wedge H_5$	$H_{11} \wedge H_3 \wedge H_7$
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Table 4.4 – Matrice des hypothèses de la MCC – tests inconsistants

La table 4.4 montre clairement que si l'hypothèse $H_8=\neg AN(C_6)$ du modèle élémentaire M_8 était fausse, alors cela expliquerait les 3 résultats inconsistants obtenus. Il en va de même pour $H_{10}=\neg AN(C_8)$ du modèle élémentaire M_{10}. $AN(C_6)$, $AN(C_8)$ sont donc des diagnostics minimaux. Dans cet exemple, il n'y a pas d'autres hypothèses qui, à elles seules, pourraient expliquer les symptômes observés. Néanmoins, deux hypothèses peuvent être fausses en même temps. Par exemple, si $H_1=\neg AN(C_1)$ et $H_9=\neg AN(C_7)$ étaient simultanément fausses des modèles élémentaires M_1 *et* M_9 alors cela pourrait expliquer les symptômes obtenus. $\neg AN(C_1)$ et $\neg AN(C_7)$ ainsi que $\neg AN(C_1)$ et $\neg AN(C_9)$ simultanément faux sont donc un diagnostic. Par ailleurs, il y a d'autres combinaisons de deux hypothèses qui sont aussi des diagnostics minimaux résumés par la table 4.4. Trois hypothèses peuvent être aussi fausses en même temps. Par exemple, si les hypothèses $H_1=\neg AN(C_1)$, $H_2=\neg AN(C_2)$ et $H_3=CC(C_2)$, données par les modèles élémentaires M_1, M_2 et M_3, étaient simultanément fausses alors cela pourrait expliquer aussi les résultats obtenus. Par

conséquent, $\neg AN(C_1)$, $\neg AN(C_2)$ et $CC(C_2)$ simultanément fausses est un diagnostic minimal. Ce qui revient à dire que soit le composant C_1 est défaillant, soit le composant C_2 mais ce n'est pas un court circuit sur le composant C_2. Les diagnostics globaux et minimaux correspondant à cet exemple sont donnés par l'ensemble D_{min} suivant:

D_{min}={ $AN(C_6)$, $AN(C_8)$, $AN(C_1)\wedge AN(C_7)$, $AN(C_1)\wedge AN(C_9)$, $AN(C_2)\wedge\neg CC(C_2)$, $AN(C_2)\wedge AN(C_7)$, $AN(C_2)\wedge AN(C_9)$, $\neg CC(C_2)\wedge AN(C_3)$, $AN(C_3)\wedge AN(C_7)$, $AN(C_3)\wedge AN(C_9)$, $AN(C_4)\wedge AN(C_7)$, $AN(C_4)\wedge AN(C_9)$, $AN(C_5)\wedge AN(C_7)$, $AN(C_5)\wedge AN(C_9)$, $AN(C_1)\wedge\neg CC(C_2)\wedge AN(C_7)$, $AN(C1)\wedge\neg CC(C_2)\wedge AN(C_9)$, $\neg CC(C_2)\wedge AN(C_3)\wedge AN(C_7)$, $\neg CC(C_2)\wedge AN(C3)\wedge AN(C_9)$, $\neg CC(C_2)\wedge AN(C_4)\wedge AN(C_7)$, $\neg CC(C_2)\wedge AN(C_4)\wedge AN(C_9)$, $\neg CC(C_2)\wedge AN(C5)\wedge AN(C_7)$, $\neg CC(C_2)\wedge AN(C_5)\wedge AN(C_9)$ }.

Conformément à la procédure présentée dans la partie (4.3), les différents diagnostics trouvés dans la partie précédente ont été ré-ordonnés suivant les valeurs contextuelles décroissantes calculées via la formule (4.13). Les résultats sont présentés dans trois ensembles différents D_{min1} , D_{min2} , D_{min3} du plus vraisemblable au moins vraisemblable.

L'exploitation des résultats contextuels a permis d'ordonner les diagnostics par ordre de plausibilité en faisant ressortir un diagnostic privilégié parmi 22 : $\neg H_2\wedge\neg H_3 = AN(C_2)\wedge\neg CC(C_2)$ avec une plausibilité circonstancielle de [100%]. Le diagnostic s'exprime littéralement ainsi : au niveau de la partie électrique (C_2) il y a un défaut mais ce n'est pas un court circuit.

D_{min1}={$AN(C_2)\wedge\neg CC(C_2)$[100%]}

D_{min2}={ $AN(C_6)$[80%], $AN(C_8)$[80%], $AN(C_1)\wedge AN(C_7)$[80%], $AN(C_2)\wedge AN(C_7)$[80%], $AN(C_2)\wedge AN(C_9)$[80%], $AN(C_5)\wedge AN(C_7)$[80%], $AN(C_1)\wedge\neg CC(C_2)\wedge AN(C_7)$[80%]}

D_{min3}={ $AN(C_1)\wedge AN(C_9)$[60%], $\neg CC(C_2)\wedge AN(C_3)$[60%], $AN(C_3)\wedge AN(C_7)$[60%], $AN(C_3)\wedge AN(C_9)$[60%], $AN(C_4)\wedge AN(C_7)$[60%], $AN(C_4)\wedge AN(C_9)$[60%] , $AN(C_5)\wedge AN(C_9)$[60%], $AN(C1)\wedge\neg CC(C_2)\wedge AN(C_9)$[60%], $\neg CC(C_2)\wedge AN(C_3)\wedge AN(C_7)$[60%], $\neg CC(C_2)\wedge AN(C3)\wedge AN(C_9)$[60%], $\neg CC(C_2)\wedge AN(C_4)\wedge AN(C_7)$[60%], $\neg CC(C_2)\wedge AN(C_4)\wedge AN(C_9)$[60%], $\neg CC(C_2)\wedge AN(C5)\wedge AN(C_7)$[60%], $\neg CC(C_2)\wedge AN(C_5)\wedge AN(C_9)$[60%]}.

Le diagnostic $AN(C_2)\wedge\neg CC(C_2)$ doit être considéré avec attention car il est le seul diagnostic contextuellement compatible avec le résultat juste des tests (SST_1, SST_4).

Notons aussi que plusieurs diagnostics possibles ont été obtenus et, de ce fait, l'opérateur sur le système doit encore déterminer lequel correspond le mieux à la MCC. Pour y parvenir, il lui faudra relever d'autres informations afin d'affiner le diagnostic.

IV.5. APPLICATION AU BIOPROCEDE

Nous avons étudié quatre types de défauts qui peuvent se produire sur un bioprocédé : Un biais de température T_1, fuite dans le mélangeur, problème de croissance du produit dans le bioréacteur, et enfin le capteur de turbidité qui peut s'encrasser. Pour illustrer notre procédure de diagnostic, nous avons considéré deux défauts simultanés, à savoir d_1: fuite dans le mélangeur, d_2: biais de capteur de température T_1.

Lors de la simulation on a fixé la quantité de la fuite dans le mélangeur à $q_\sigma=7.10^{-2}$ litres/heure et la variation de température ΔT=3° (degré Celsius) entre t=29.1s et t=58.1s. Ces deux défauts sont des défauts additifs qui agissent directement et respectivement sur les relations élémentaires R_1 et R_{20} .

$$R_1 : \; S_1 \frac{d\left[h_1\right]}{dt} = [q_1] - [q_2] - q_\sigma$$
$$R_{20} : \; \tilde{T}_1 = [T_1] \pm \Delta T \qquad (4.14)$$

Il y avait au départ 276 sous-systèmes testables pour cette application. Seulement 23 sous-systèmes testables ont été retenus en appliquant le critère de rang [Ploix et al, 2005]. Par un souci de clarté et pour pouvoir être plus concis dans nos exemples et dans nos explications, nous avons considéré seulement cinq sous-systèmes testables choisis parmi les 23 retenus. La table 4.5 regroupe les SST implémentés. Le choix s'est fait du fait que les sous-systèmes testables nécessaires à la détection des deux défauts d_1 et d_2 doivent inclure les deux modèles élémentaires M_1 et M_9 correspondant aux relations élémentaires R_1 et R_{20} (voir chapitre 3).

La table 4.5 illustre que le sous-système testable SST_5 est sensible uniquement au défaut d_1. En effet, SST_5 se construit en utilisant dans son modèle de comportement le modèle élémentaire M_1 affecté par le défaut d_1. Par ailleurs, SST_9 et SST_{16} sont sensibles à la fois au défaut d_1 et au défaut d_2. En revanche, SST_7 et SST_{20} sont insensibles aux défauts d_1 et d_2 car ils n'incluent pas les relations élémentaires sur lesquelles apparaît le défaut. Ils sont choisis afin de nous servir de témoins lors de la détection.

Hi(M_i)➔	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	H_{10}	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}	H_{16}	H_{17}	H_{18}	H_{19}	H_{20}	H_{21}	H_{22}	H_{23}
SST_5	1															1	1					1	
SST_7						1	1					1		1	1					1			
SST_9	1	1							1							1					1	1	
SST_{16}	1	1							1	1						1					1		
SST_{20}			1	1		1	1			1		1			1			1	1				1

Table 4.5 – Matrice signatures pour le bioprocédé

La figure 4.1 illustre les résultats des tests de détection sensibles aux défauts d_1 et d_2 réalisés simultanément. Chacun des sous systèmes testables possède trois schémas. Le premier schéma (a) illustre le modèle de référence superposé au modèle estimé. Le deuxième schéma (b) trace la courbe du résidu avec le seuil fixé pour la détection. Le troisième schéma (c) met en évidence la détection. Les seuils sont fixés à partir de jeu de résidu en bon fonctionnement en rajoutant plus ou moins 15% de la valeur du résidu. Il en résulte donc, pour la SST_5 le seuil a été fixé à 0.006° et la détection apparaît à l'instant t=32s (voir figure 4.1). Pour le SST_9 le seuil a été fixé à 0.0003° et la détection apparaît à l'instant t=37s (voir figure 4.1 (b)). Comme le modèle ne tient pas compte des bruits de mesures, un filtre basse-bas de type moyenneur a été utilisé. Une moyenne sur 50 échantillons s'est avérée suffisante.

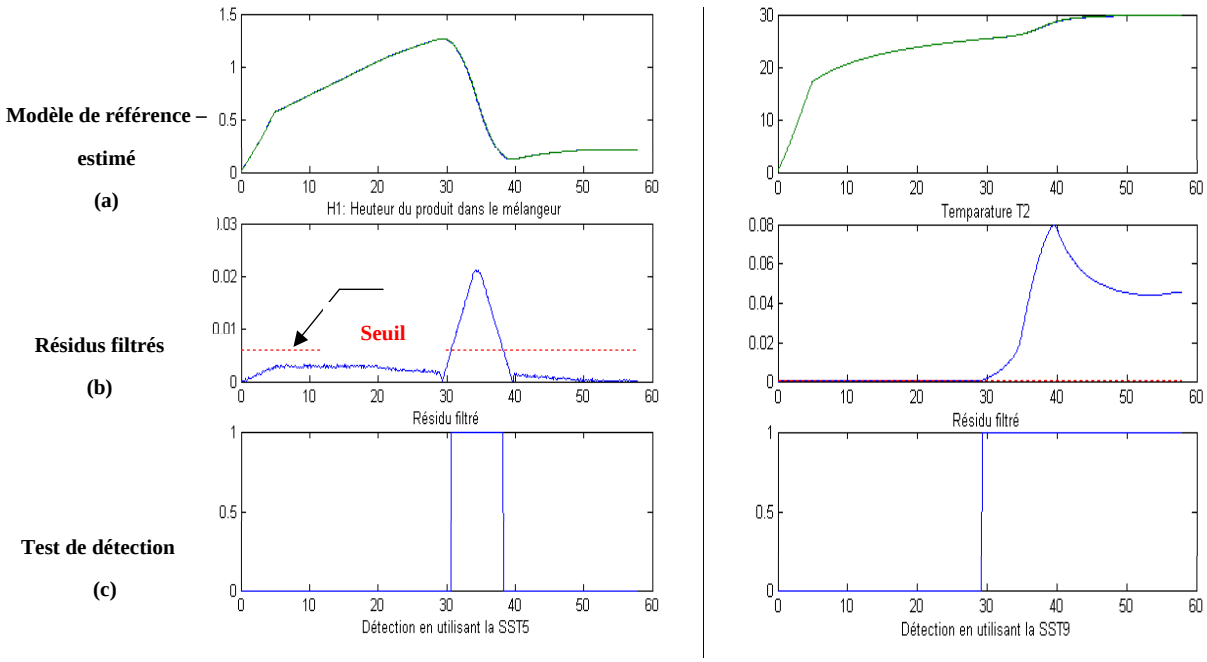


Figure 4.1 - Résultats des tests de détection (SST_5, SST_9)

La figure 4.2 illustre les résultats des tests SST_7 et SST_{16} et SST_{20}. Le SST_7 teste la cohérence entre le modèle de référence de la température (T_6) à la sortie de l'échangeur de chaleur coté eau froide et son modèle de référence. Le SST_{16} teste la cohérence entre le modèle de référence de température (T_2) du produit dans le mélangeur et son modèle attendu. La figure 4.2 met en évidence des tests de détection non sensibles aux défauts d_1 et d_2. En effet, les tests effectués sur SST_7 et SST_{16} ainsi que SST_{20} se révèlent consistants. Par conséquent, les trois tests ne détectent aucun des deux défauts. Ceci est dû au fait qu'ils ne sont pas sensibles aux défauts d_1 et d_2.

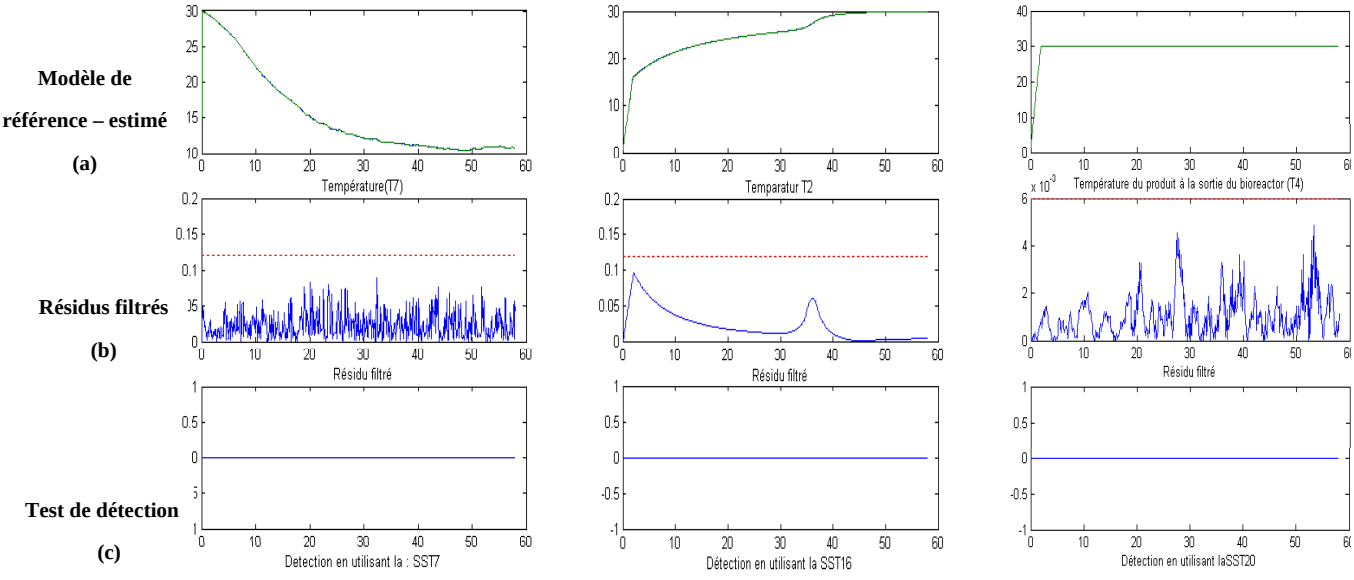


Figure 4.2 - Résultats des tests de détection (SST_7, SST_{16}, SST_{20})

Seuil sur la SST_7 s=*0.12*°, Seuil sur la SST_{16} s= *0.12*°, Seuil sur la SST_{20} s=*0.006* °.

On considère seulement les tests inconsistants regroupés par la table 4.7. Les colonnes représentent les hypothèses sur l'état des composants du bioprocédé présent dans chaque modèle élémentaire. Chaque hypothèse de composant peut être fausse et peut ainsi expliquer seule

l'inconsistance des deux tests. Néanmoins, deux hypothèses peuvent être aussi fausses en même temps et pourraient expliquer les symptômes obtenus.

$H_i(Mi)\rightarrow$	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	H_{10}	H_{11}	H_{12}	H_{13}	H_{14}	H_{15}	H_{16}	H_{17}	H_{18}	H_{19}	H_{20}	H_{21}	H_{22}	H_{23}	$H_2 \wedge H_{17}$	$H_9 \wedge H_{17}$	$H_{17} \wedge H_{21}$
SST_5	1															1	1					1		1	1	1
SST_9	1	1							1							1					1	1		1	1	1

Table 4.7 – Matrice des signatures pour les tests inconsistants

En effet, la table 4.7 montre clairement que si l'hypothèse $H_1=\neg AN(mélangeur)$du modèle élémentaire M_1 était fausse, alors cela expliquerait les 2 résultats inconsistants obtenus. Il en va de même pour $H_{16}=\neg AN(actionneur\ q_1)$ et $H_{22}=\neg AN(capteur\ h_1)$ respectivement du modèle élémentaire ME_{16} et ME_{22}. *AN(mélangeur), AN(actionneur q_1), AN(capteur h_1)* sont donc des diagnostics minimaux. Dans cet exemple, il n'y a pas d'autre hypothèse qui, à elle seule, pourrait expliquer les symptômes observés. Deux hypothèses peuvent être fausses en même temps. Par exemple, si $H_2=\neg AN(Résistance)$ et $H_{17}=\neg AN(actionneur\ q_2)$ des modèles élémentaires M_2 *et* M_{17} étaient simultanément fausses alors cela pourrait expliquer les symptômes obtenus. $\neg AN(Résistance)$ et $\neg AN(actionneur\ q_2)$, $\neg AN(Capteur\ T_1)$ et $\neg AN(\ actionneur\ q_2)$ ainsi que $\neg AN(actionneur\ q_2)$ et $\neg AN(capteur\ de\ courant)$ simultanément faux sont donc un diagnostic. Les diagnostics minimaux correspondants au bioprocédé sont donnés par :

D_{min}*={AN(mélangeur), AN(actionneur q_1), AN(capteur h_1), AN(Résistance)*$\wedge$*AN(actionneur q_2), AN(Capteur T_1)* $\wedge$ *AN(actionneur q_2), AN(actionneur q_2)* $\wedge$ *AN(capteur de courant)}* .

Les diagnostics obtenus peuvent alors être complétés par les mesures de compatibilité contextuelle et donne les ensembles suivants:

D_{min1}*={AN(capteur h_1) [100%]}*

D_{min2}*={AN(mélangeur) [80%], AN(actionneur q_1)[80%], AN(Résistance)*$\wedge$*AN(actionneur q_2) [80%], AN(Capteur T_1)*$\wedge$*AN(actionneur q_2)[80%], AN(actionneur q_2)*$\wedge$*AN(capteur de courant)[80%]}.*

L'un des défauts simulés correspondait bien à une fuite dans le mélangeur. Le diagnostic obtenu en tête *AN(capteur h1)* met en causse le mélangeur. Il révèle qu'il y a bien un problème au niveau de la hauteur du produit dans le mélangeur. Le deuxième diagnostic *AN(mélangeur)* avec une plausibilité de 80% vient confirmer le premier diagnostic.

IV.6. APPLICATION AU SYSTEME DES BACS

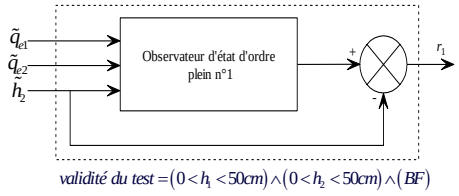
Les principes d'analyse diagnostique exposés précédemment sont appliqués sur le système des deux bacs. L'objectif est d'obtenir un diagnostic exhaustif à chaque scénario de défaut simulé. Pour l'exemple des bacs nous avons obtenu la matrice de signatures suivante qui résume les différents sous systèmes testables :

	$AN(T_1)$	$AN(R_1)$	$AN(T_2)$	$AN(R_2)$	$AN(H_1)$	$AN(H_2)$	$AN(Q_1)$	$AN(Q_2)$
SST_1	1	1	1	1		1	1	1
SST_2		1	1	1	1	1		1
SST_3	1	1			1		1	

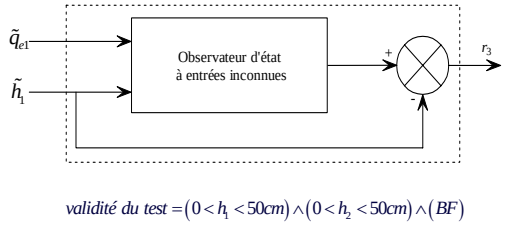
Tableau 4.19- Matrice des signatures pour le système des deux bacs

Une fois tous les tests construits et implémentés, la phase d'analyse diagnostique peut démarrer. Les tests valides génèrent des symptômes : le résultat de chacun des tests validés peut être soit vrai, soit faux. On considère ici que le résultat des tests de détection sont booléens, soit le résultat est vrai (noté 1) quand il n'y a pas d'inconsistance, soit le résultat est faux quand le test génère une alarme du fait d'une inconsistance (noté 0). Nous reprenons la réalisation des trois tests de détection T_1, T_2 et T_3 en utilisant un observateur d'état.

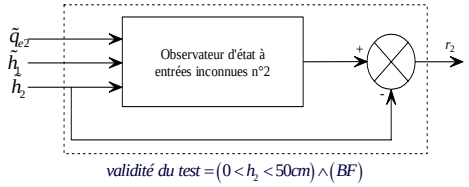
1er observateur d'état indépendant de h_1 (SST_1)



3ème observateur d'état indépendant de h_2 et q_{e2} (SST_2)



2ème observateur d'état indépendant de q_{e1} (SST_3)



Cette simulation nous a permis d'étudier 4 scénarios de défauts possibles.

Scénario 1 : Biais sur h_2 (5 cm)

Le scénario 1 simule un biais de capteur de hauteur d'eau dans le bac du bas (h_2). Les résultats des trois tests de détections sont donnés par les trois résidus (r_1, r_2, r_3) (voir figure 5.10). Les tests T_1 et T_2 conduisent à des alarmes. En effet, la courbe tracée des résidus (r_1 et r_2) dépasse les seuils imposés. Ce dépassement des seuils nous a permis de confirmer la détection. En revanche, le test T_3 conduit à dire que les modèles utilisés par le test et les observations sont consistants car le résidu correspondant au test T_3 reste à l'intérieur des seuils.

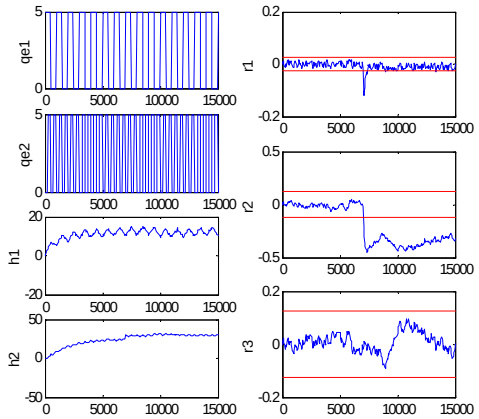


Figure 4.10 - Biais sur h_2 (5 cm)

La matrice des hypothèses donnée par la (table 4.19) montre clairement que si l'hypothèse $\neg AN(R_1)$ était fausse, alors cela expliquerait les 2 résultats faux obtenus. Il en va de même pour $\neg AN(T_2)$, $\neg AN(R_2)$, $\neg AN(H_2)$ et $\neg AN(Q_2)$. $AN(R_1)$, $AN(T_2)$, $AN(R_2)$, $AN(H_2)$ et $AN(Q_2)$ sont donc des diagnostics minimaux. Dans cet exemple, il n'y a pas d'autre hypothèse qui, à elle seule, pourrait expliquer les symptômes observés. Néanmoins, deux hypothèses peuvent être fausses en même temps. Par exemple, si $\neg AN(T_1)$ et $\neg AN(H_1)$ étaient simultanément fausses alors cela pourrait expliquer les symptômes obtenus. $\neg AN(T_1)$ et $\neg AN(H_1)$ ainsi que $\neg AN(H_1)$ et $\neg AN(Q_1)$ simultanément faux sont donc un diagnostic. Les diagnostics minimaux correspondant à cet exemple sont donnés par : {$AN(R_1)$, $AN(T_2)$, $AN(R_2)$, $AN(H_2)$ et $AN(Q_2)$, $\neg AN(T_1) \wedge \neg AN(H_1)$, $\neg AN(H_1) \wedge \neg AN(Q_1)$ }. Le test T_3 a conduit à un résultat vrai. Le diagnostic $AN(R_1)$ est alors incompatible avec T_3 qui contient l'hypothèse $\neg AN(R_1)$: il y a donc 0 test vrai compatible avec ce diagnostic. Ce diagnostic aura donc une mesure de compatibilité contextuelle de 0. En revanche, $AN(H_2)$ aura une mesure égale à 1 (1 test vrai compatible). Les diagnostics obtenus peuvent alors être complétés par les mesures de compatibilité contextuelle :

$AN(T_2)$ [1]	$AN(H_2)$ [1]	$AN(Q_2)$ [1]	$AN(R_2)$ [1]
$AN(R_1)$ [0.66]			
$AN(T_1) \wedge AN(H_1)$ [0.66]		$AN(Q_1) \wedge AN(H_1)$ [0.66]	

Dans ce cas, la mesure de compatibilité contextuelle montre que le diagnostic donnant $AN(R_1)$ est contextuellement moins plausible que les autres diagnostics $AN(T_2)$, $AN(R_2)$, $AN(H_2)$, $AN(Q_2)$. Bien que cette information soit utile, il ne faut néanmoins pas la confondre avec les autres résultats obtenus qui ont, eux, valeur de preuve. On remarque que le diagnostic $AN(H_2)$ a été trouvé avec la plus grande plausibilité.

Scénario 2 : Biais (20%) sur la restriction du bac haut (non trouvé précédemment)

Le biais de 20% sur la restriction du bac du haut a conduit aux résultats suivants : le test T_1 ne détecte rien, T_2 et T_3 détectent une anomalie. Par conséquent, les diagnostics minimaux sont donnés par { $AN(H_1)$, $AN(R_1)$, $AN(T_1) \wedge AN(T_2)$, $AN(T_1) \wedge AN(H_2)$, $AN(T_2) \wedge AN(Q_1)$, $AN(H_2) \wedge AN(Q_1)$, $AN(T_1) \wedge AN(R_2)$, $AN(T_1) \wedge AN(Q_2)$, $AN(R_2) \wedge AN(Q_1)$, $AN(Q_1) \wedge AN(Q_2)$ }.

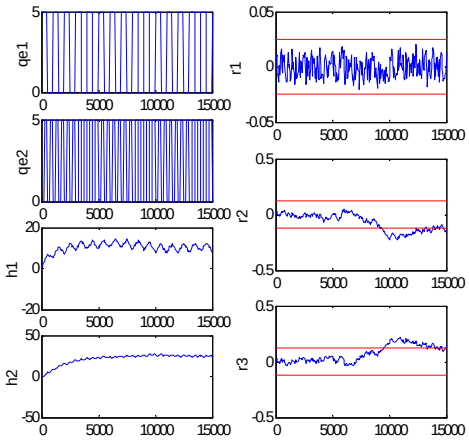


Figure 4.12 - Biais (20%) sur la restriction du bac haut

Le diagnostic $AN(H_1)$ est le diagnostic le plus contextuellement vraisemblable. Ce dernier n'est pas incompatible avec le résultat du test T_1. En effet, l'hypothèse $\neg AN(H_1)$ n'apparaît pas dans

le support du test T_1. En revanche, ce n'est pas le cas pour le diagnostic $AN(R_1)$ qui explique à lui seul le résultat des deux tests T_2 et T_3. Ce dernier diagnostic est la bonne solution, il n'avait pas été trouvé précédemment (cf. chapitre 2) en utilisant la méthode par table de signature. L'ensemble des diagnostics minimaux classés par ordre de plausibilité contextuelle est donné par le biais de ce tableau :

$AN(H_1)$ [1]	
$AN(R_1)$ [0.66]	
$AN(T_1) \wedge AN(T_2)$ [0.66]	$AN(T_1) \wedge AN(H_2)$ [0.66]
$AN(T_2) \wedge AN(Q_1)$ [0.66]	$AN(H_2) \wedge AN(Q_1)$ [0.66]
$AN(T_1) \wedge AN(R_2)$ [0.66]	$AN(T_1) \wedge AN(Q_2)$ [0.66]
$AN(R_2) \wedge AN(Q_1)$ [0.66]	$AN(Q_1) \wedge AN(Q_2)$ [0.66]

Scénario 3 : Biais sur h_2 (2 cm) et sur q_{e1} (0.2 l/s) (Les défauts multiples sont trouvés)

Ce scénario illustre que notre approche traite bien le cas des défauts multiples. En effet, deux défauts simultanés ont été simulés sur le système des bacs : un biais de capteur h_2 de (2cm) et un biais d'actionneur q_{e1} de (0.2 l/s). Les résultats des tests de détections (figure 4.13) montrent que T_1 ne détecte rien, T_2 et T_3 détectent un défaut. Ce résultat de la détection correspond au vecteur de signature $V_R=(r_1, r_2, r_3)=(0,1,1)^T$. Les ensembles des conflits minimaux sont : {$AN(R_1)$, $AN(T_2)$, $AN(R_2)$, $AN(H_1)$, $AN(H_2)$, $AN(Q_2)$}, { $AN(T_1)$, $AN(R_1)$, $AN(H_1)$, $AN(Q_1)$}.

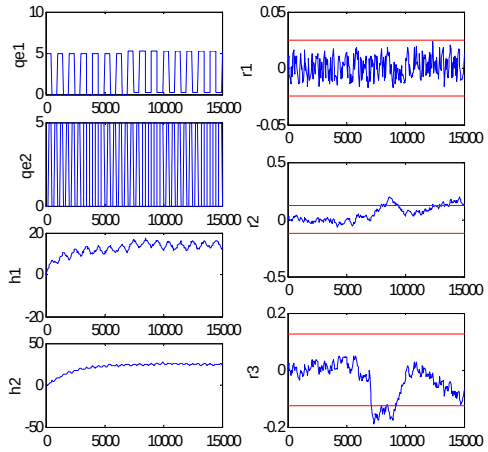


Figure 4.13 - Biais sur h_2 (2 cm) et sur q_{e1} (0.2 l/s) Les diagnostics minimaux associés correspondent aux défauts simples {$AN(R_1)$}, {$AN(H_1)$} et aux défauts doubles {$AN(T_1) \wedge AN(T_2)$, $AN(T_1) \wedge AN(H_2)$, $AN(T_2) \wedge AN(Q_1)$, $AN(H_2) \wedge AN(Q_2)$, $AN(T_1) \wedge AN(R_2)$, $AN(T_1) \wedge AN(Q_2)$, $AN(R_2) \wedge AN(Q_1)$, $AN(Q_1) \wedge AN(Q_2)$ }. Les défauts simultanés sur le h_2 (capteur de hauteur d'eau dans le bac du bas) et q_{e1} (actionneur q_{e1}) ont été bien détectés.

$AN(H_1)$ [1]	
$AN(R_1)$ [0.66]	
$AN(T_1) \wedge AN(T_2)$ [0.66]	$AN(T_1) \wedge AN(H_2)$ [0.66]
$AN(T_2) \wedge AN(Q_1)$ [0.66]	$AN(H_2) \wedge AN(Q_1)$ [0.66]
$AN(T_1) \wedge AN(R_2)$ [0.66]	$AN(T_1) \wedge AN(Q_2)$ [0.66]
$AN(R_2) \wedge AN(Q_1)$ [0.66]	$AN(Q_1) \wedge AN(Q_2)$ [0.66]

Scénario 4 : aucune alarme (prise en compte de la validité)

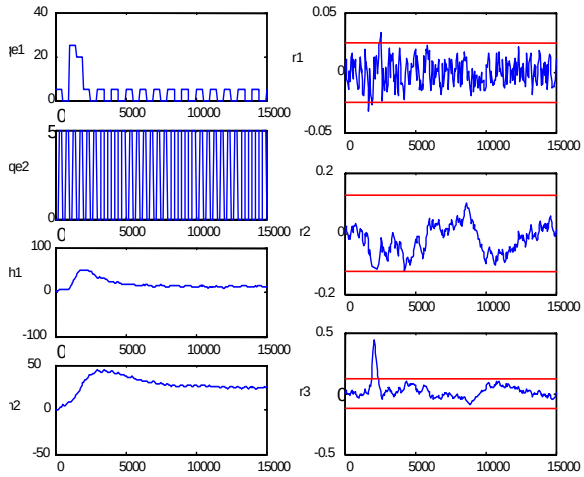


Figure 4.14 – Aucun défaut simulé (validité prise en compte)

Le domaine de validité d'un test de détection est important car un résultat n'est valide que si l'état du système étudié appartient au domaine de validité. Or la figure 4.14 illustre que la validité de la hauteur dans le bac du haut (h_1) atteint les 50cm. Par conséquent, le modèle du bac du haut n'est plus valide et le test dans lequel il est impliqué en hérite. Or, bien que les tests T_1 et T_3 détectent des défauts, notre approche n'en tient pas compte et elle les considère comme consistants car au moment de la détection les modèles sur lesquels les tests reposent n'étaient pas valides. Par conséquent, aucun diagnostic n'est proposé.

IV.7. PRISE EN COMPTE DES INCERTITUDES DE DECISION

Les tests de détection ne fournissent pas que des résultats sûrs (consistant ou inconsistant). En effet, les tests peuvent conduire à des décisions incertaines. On a vu précédemment (section 8 dans le chapitre 4) que la fonction de test notée ($\neg\sigma_{t_k}$) ainsi que la fonction de validité du test notée (υ_{t_k}) peuvent appartenir à l'intervalle [0, 1]. Quand $\neg\sigma_{t_k} = 1$, cela signifie que le test est inconsistant d'une manière sûre et quand $\neg\sigma_{t_k} = 0$ signifie que le test consistant d'une manière sûre. En revanche, $\neg\sigma_{t_k} = 0.5$ signifie qu'il y a un doute dans la décision du test (impossible de trancher entre une consistance ou une inconsistance). Par exemple, 0.2 veut dire que le test présente 20% de possibilité d'être inconsistant et par la complémentarité, le test a une possibilité de 80% d'être consistant. La validité d'un test de détection peut aussi être graduelle. Elle indique si les modèles utilisés par le test sont valides ou non en un point de fonctionnement. Considérons par exemple un modèle linéarisé autour d'un point de fonctionnement. La validité tend vers 0 (non appartenance à l'ensemble valide) si le point de fonctionnement courant s'éloigne de point autour duquel le modèle a été linéarisé.

Pour prendre en compte cette incertitude dans la décision qui va nous servir à affiner l'analyse diagnostique, certains concepts de la logique floue ont été utilisés [Zadeh, 1965]. Nous avons étudié la prise en compte des incertitudes de décision dans les deux cas : le cas structurel (avec exonération) et le cas logique (sans exonération).

IV.7.1. Prise en compte d'incertitudes de décision en analyse structurelle

Cette partie s'intéresse à la méthode d'analyse diagnostique utilisée habituellement par la communauté FDI (*Fault Detection and Isolation*). Cette approche est appelée approche structurelle. Cette dernière, consiste à comparer directement la signature d'une alarme générée par un test de détection avec les colonnes de la table de signatures des défauts.

A titre d'exemple, la table 4.8 regroupe trois tests de détections t_1, t_2 et t_3. Chaque test de détection t_i est en rapport avec une hypothèse H_j sur l'état du composant. H_j appartient à l'ensemble des hypothèses $\mathcal{H}(t_i)$, c'est-à-dire à l'ensemble des défauts possibles. Par exemple, le test de détection t_1 vérifie si les composants C_1, C_3 et C_4 sont dans un état normal. Le test t_1 est par exemple sensible au défaut sur le composant C_1 mais complètement insensible au défaut sur le composant C_2.

	$\neg AN(C_1)$	$\neg AN(C_2)$	$\neg AN(C_3)$	$\neg AN(C_4)$
t_1	1	0	1	1
t_2	0	1	1	0
t_3	1	0	1	0

Table 4.8- Exemple de table de signatures de défaut

Dans le contexte de la logique binaire, $\neg H(t_1)$, $H(t_2)$ et $\neg H(t_3)$ mèneront à la conclusion que *AN(C_1)* est un diagnostic parce que sa signature de défaut est cohérente avec les symptômes. La question maintenant qui se pose est de chercher des signatures de défaut quand les tests de détection mènent à des décisions incertaines.

Soit $[\lambda_{ij}]$ une matrice qui représente une table de signature de défaut telle que la table 4.8 représentée ci-dessus. L'indice *i* fait référence au i^{ème} test de détection et l'indice *j* fait référence à la *j*^{ème} hypothèse H_j. La table de la signature de défaut est définie par :

$$\begin{cases} H_j \in \mathcal{H}(t_i) \Leftrightarrow \lambda_{ij} = 1 \\ H_j \notin \mathcal{H}(t_i) \Leftrightarrow \lambda_{ij} = 0 \end{cases} \tag{4.15}$$

Logiquement une signature de défaut sera cohérente avec le résultat du test de détection si :

$$\forall t_i, \left(H_j \in \mathcal{H}(t_i)\right) \Leftrightarrow \neg \hat{\mathcal{H}}(t_i) \tag{4.16}$$

où $\hat{\mathcal{H}}(t_i)$ représente l'ensemble des hypothèses sur lequel est construit le test de détection.

Si la proposition (4.16) est satisfaite, alors $\neg H_j$ est un diagnostic. La proposition (4.16) peut être reformulée comme suit :

$$\frac{\begin{array}{c} \rightarrow \forall t_i, \left(H_j \in \mathcal{H}(t_i)\right) \Leftrightarrow \neg \hat{\mathcal{H}}(t_j) \\ \hline \rightarrow \wedge_{t_i} \left(\left(H_j \in \mathcal{H}(t_i)\right) \Leftrightarrow \neg \hat{\mathcal{H}}(t_i)\right) \end{array}}{\rightarrow \wedge_{t_i} \left(\left(\left(H_j \in \mathcal{H}(t_i)\right) \wedge \neg \hat{\mathcal{H}}(t_i)\right) \vee \left(\left(H_j \notin \mathcal{H}(t_i)\right) \wedge \hat{\mathcal{H}}(t_i)\right)\right)} \tag{4.17}$$

Soit *P* la dernière expression ainsi obtenue. Le degré de vérité *T(P)* est calculé grâce à l'arithmétique de la logique floue. Cette fonction de vérité correspond à la fonction de vérité du test de détection. L'opérateur *min(.)* a été choisi comme *T-norm* et la somme comme *T-conorm* pour simplifier l'écriture de *T(P)*.

$$\begin{aligned}\mu_{\neg H_j} &= T\left(\underset{t_i}{\wedge}\left(\left(\left(H_j \in \mathcal{H}(t_i)\right) \wedge \neg\hat{\mathcal{H}}(t_i)\right) \vee \ldots \left(\left(H_j \notin \mathcal{H}(t_i)\right) \wedge \hat{\mathcal{H}}(t_i)\right)\right)\right) \\ &= \min_{t_i}\left(\lambda_{ij}\mu_{\neg\hat{\mathcal{H}}(t_i)} + \left(1-\lambda_{ij}\right)\left(1-\mu_{\neg\hat{\mathcal{H}}(t_i)}\right)\right)\end{aligned} \tag{4.18}$$

La formule (4.18) représente la transcription de la recherche de signatures dans la matrice de signature en logique nette

Exemple.

Supposons que les tests de détections représentés par la table 4.1 mènent aux résultats suivants :

$$\left(\mu_{\neg\hat{\mathcal{H}}(t_1)}, \mu_{\neg\hat{\mathcal{H}}(t_2)}, \mu_{\neg\hat{\mathcal{H}}(t_3)}\right) = \left(0.8, 0.5, 0.4\right) \tag{4.19}$$

La table 4.9 résume la plausibilité formelle des différents diagnostics.

Diagnostics	**AN(C_4)**	**AN(C_1)**	**AN(C_3)**	**AN(C_2)**
Plausibilité formelle	0.50	0.40	0.40	0.20

Table 4.9 – Diagnostics structurels

Le diagnostic *AN(C_4)* est le diagnostic le plus plausible. Ce n'est pas vraiment surprenant à cause de la forte ressemblance entre sa signature et les symptômes collectés. En effet, le résultat des trois tests de détections $(0.8, 0.5, 0.4)^T$ est plus proche de la signature du défaut sur le composant C_4 $(1, 0, 0)^T$ que par rapport à d'autres signatures des autres défauts.

IV.7.2. Prise en compte d'incertitudes de décision en analyse logique

Cette section étudie l'influence de l'incertitude de la décision dans l'analyse diagnostique logique. La phase d'analyse logique doit appréhender ce type de situation.

IV.7.2.1. Le calcul des diagnostics

Une distinction est faite entre la situation où tous les résultats fournis par les tests de détection ne sont pas inconsistants d'une manière sûre et la situation où il existe au moins un test de détection qui est en contradiction certaine avec les observations. En effet, parce qu'aucune des décisions n'est sûre, les diagnostics sont plutôt des pré-diagnostics car ce n'est pas certain qu'un défaut soit véritablement présent dans le système. Dans ce contexte, chaque diagnostic possible implique seulement un composant, ce qui permet d'écrire l'ensemble D_{min} donné par 4.20. :

Soit t_i un test de détection et $\hat{\mathcal{H}}_{t_i}$ l'hypothèse sur les états des composants testés par le test t_i. $\mu_{\neg\hat{\mathcal{H}}_{t_i}}$ représente la fonction d'appartenance à faux de la décision du test de détection.

$\forall t_i, \mu_{\neg\hat{\mathcal{H}}_{t_i}} \in [0,1[$ alors les diagnostics possibles sont donnés par D_{min}:

$$D_{\min} = \left\{ \neg H_k / H_k \in \bigcup_{t_i / \mu_{\neg\hat{\mathcal{H}}_{t_i}} \in]0,1[} \mathcal{H}_{t_i} \right\} \tag{4.20}$$

Démonstration

Soit $\mathcal{T}_u$ un ensemble de tests de détection dont la décision est incertaine : $\mathcal{T}_u = \left\{ t_i / \mu_{\neg\hat{\mathcal{H}}_{t_i}} \in]0,1[\right\}$.

Par définition, chaque test de l'ensemble $\mathcal{T}_u$ peut mener soit à un résultat consistant, soit à un résultat inconsistant. Considérant la situation où seulement un seul test de détection $t_i \in \mathcal{T}_u$ est inconsistant, les diagnostics minimaux sont donnés par :

$$\left\{ \neg H_k / H_k \in \mathcal{H}(t_i) \right\} \tag{4.21}$$

Soit la situation suivante où plusieurs tests de détection $t_{i:i\in\phi}$ sont inconsistants. Chaque diagnostic peut contenir une ou plusieurs hypothèses $\neg H_k$. Cependant chaque défaut apparaît nécessairement au moins dans un diagnostic obtenu en considérant strictement un test de détection $t_{i:i\in\phi}$ inconsistant. Par ailleurs, du fait que seuls les diagnostics minimaux sont pris en compte, seuls les diagnostics qui contiennent uniquement une hypothèse fausse doivent être considérés. Tous les diagnostics minimaux sont donc donnés par :

$$\bigcup_{t_i \in \mathcal{T}_u} \left\{ \neg H_k / H_k \in \mathcal{H}(t_i) \right\} \tag{4.22}$$

Cette union donne l'ensemble défini par (4.20).

Soit maintenant la deuxième situation où il existe au moins un test de détection inconsistant d'une manière sûre. Soit $\mathcal{T}_c$ l'ensemble des tests de détection sûrement inconsistant donné par :

$$\mathcal{T}_c = \left\{ t_i / \mu_{\neg\hat{\mathcal{H}}_{t_i}} = 1 \right\} \tag{4.23}$$

Soit $\mathcal{D}_c$ l'ensemble des diagnostics logiques calculés suivant l'algorithme de Reiter. Chaque élément d_j de l'ensemble $\mathcal{D}_c$ explique les conflits obtenus à partir des différents tests de détection inconsistants impliqués dans $\mathcal{T}_c$.

Si les tests de détection de l'ensemble $\mathcal{T}_c$ ne peuvent pas être consistants alors d'autres tests de détection de l'ensemble $\mathcal{T}_u$ peuvent mener à des résultats consistants ou inconsistants. Si tous les tests de l'ensemble $\mathcal{T}_u$ sont consistants, alors aucun conflit n'apparaît et les diagnostics seront donnés par l'ensemble $\mathcal{D}_c$. On considère maintenant que quelques tests de détection de l'ensemble $\mathcal{T}_u$ mènent à des résultats inconsistants. Alors un diagnostic d_j de l'ensemble $\mathcal{D}_c$ peut ne pas pouvoir expliquer les conflits supplémentaires et quelques hypothèses $\neg H_j$ de plus devraient être ajoutées à d_j pour expliquer tous les conflits. En revanche, puisque d_j est déjà un diagnostic, alors d_j plus les

hypothèses supplémentaires ne sera plus minimal. Par conséquent, les diagnostics minimaux sont tous donnés par $\mathcal{D}_c$.

IV.7.2.2. Le degré de vérité des diagnostics

Les deux situations précédentes seront distinguées dans cette partie.

Dans la première situation tous les tests de détection sont inconsistants d'une manière sûre, autrement dit, avec aucune incertitude ($\mathcal{T}_c$=1). La vérité d'un diagnostic donné par $\neg H_k$ est égale par définition à :

$$T(\neg H_k)= 1-T(H_k) \tag{4.24}$$

Le degré de vérité du diagnostic H_k noté $T(H_k)$ peut être calculé de telle sorte que le résultat de la détection $\neg H_k$ ne soit impliqué dans aucun diagnostic.

On dénote par $\mathcal{T}_{Hk}$ l'ensemble des tests de détection qui ne font pas référence à H_k:

$$\mathcal{T}_{H_k} = \left(t_i / H_k \in \mathcal{R}(t_i)\right) \tag{4.25}$$

Tous les résultats de la détection qui n'implique pas l'hypothèse H_k sont donnés par :

$$H_k = \bigwedge_{t_i \in \mathcal{T}_{H_k}} \hat{\mathcal{R}}(t_i) \tag{4.26}$$

Par conséquent, la vérité de l'expression (4.18) peut être calculée comme suit:

$$\rightarrow T\left(\neg H_k\right) = T\left(\neg \bigwedge_{t_i \in \mathcal{T}_{H_k}} \hat{\mathcal{R}}(t_i)\right) = T\left(\bigvee_{t_i \in \mathcal{T}_{H_k}} \neg\hat{\mathcal{R}}(t_i)\right)$$

$$\boxed{\mu_{\neg H_k} = \max_{t_i \in \mathcal{T}_{H_k}}\left(\mu_{\neg\hat{\mathcal{R}}(t_i)}\right)} \tag{4.27}$$

Nous allons étudier maintenant la deuxième situation où $\mathcal{T}_c \neq 1$. La vérité de chaque diagnostic d_j de l'ensemble $\mathcal{D}_c$ doit être calculée. Les diagnostics d_i apparaîtront si :

$$d_i = \bigwedge_{t_i \in \mathcal{T}_c} \neg\hat{\mathcal{R}}(t_i) \tag{4.28}$$

A cause de la minimalité, et puisque le diagnostic est indépendant des résultats des tests de détection de l'ensemble $\mathcal{T}_u$. On écrit alors

$\forall d_j \in \mathcal{D}_c$,

$$\boxed{T\left(d_j\right) = T\left(\bigwedge_{t_i \in \mathcal{T}_c} \neg\hat{\mathcal{R}}(t_i)\right) = 1} \tag{4.29}$$

IV.7.3. Application sur le système des deux bacs

Reprenons l'exemple des deux bacs étudié précédemment (chapitre 2). Les résultats des tests de détection t_1, t_2 et t_3 conduisent respectivement aux trois résidus r_1, r_2 et r_2. La décision des tests de détection implémentée est incertaine comprise entre 0 et 1. Deux scénarios de défaut ont été étudiés et analysés.

Scénario 1 : Biais sur (H_2) de 1.2cm

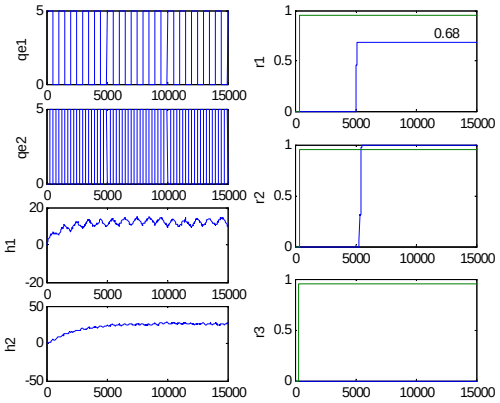


Figure 4.15 - Biais sur hauteur bac du bas (*H_2)* (1.2 cm)

La figure 4.15 illustre les symptômes de la détection des trois tests effectués sur le système des bacs. Ce résultat de la détection est égal à (r_1, r_2, r_3) =(0.68, 1, 0). Certains de ces résidus donnent un résultat sûr tels que les tests t_2 et t_3. En revanche, le test t_1 est incertain. En effet, sa valeur est de 0.68, c'est-à-dire : ce test est faux à 68% et par complémentarité ce test est vrai à 32%. Les résultats logiques du diagnostic ainsi obtenu sont résumés dans la table 4.8. La plausibilité circonstancielle a été calculée suivant la formule (4.13) et la plausibilité des diagnostics formels d'après la formule (4.29).

	AN(R_1)	AN(T_2)	AN(R_2)	AN(H_1)	AN(H_2)	AN(Q_2)
Formel	1	1	1	1	1	1
Circonstancielle	0,56	0,89	0,89	0,44	0,89	0,89

Table 4.10 – Résultat diagnostic

Les diagnostics les plus vraisemblables (89%) sont : soit le bac du bas (T_2), soit la restriction (R_2), soit l'actionneur (Q_2), ou bien le capteur de hauteur d'eau (H_2) sont dans des états anormaux. Le biais sur le capteur (H_2) apparaît parmi les diagnostics les plus vraisemblables.

Scénario 2 : Biais de 6% sur la restriction du bac du haut

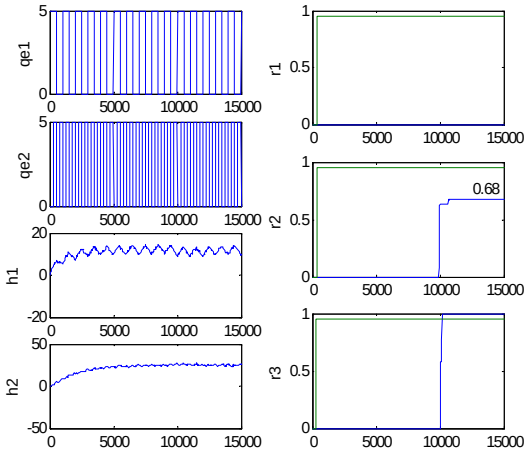


Figure 4.4 -Biais (6%) sur la restriction du bac haut (R_1)

Les symptômes des tests de détection donnent les résidus suivant : (r_1, r_2, r_3)= (0, 0.68, 1). Les pré-diagnostics sont résumés par la table 4.11 car le diagnostic n'est pas avéré (aucun résultat sûr). Le pré-diagnostic le plus vraisemblable avec 89% est *AN(H_1)*, autrement dit, il y a un problème sur le capteur de hauteur d'eau (H_1). En revanche, le deuxième pré-diagnostic AN(R_1) indique qu'il pourrait y avoir un problème sur la restriction du bac du haut.

	AN(T_1)	AN(R_1)	AN(H_1)	AN(Q_1)
Formel	1	1	1	1
Circonstancielle	0,44	0,56	0,89	0,44

Table 4.11 – Résultat diagnostique

IV.8. STRATEGIES DU DIAGNOSTIC

IV.8.1. Pourquoi mettre en place une stratégie de diagnostic ?

Dans le cas de systèmes physiques complexes, le nombre de tests de détection peut devenir très important. Il devient alors nécessaire de n'effectuer que certains tests dits initiaux qui, si un symptôme apparaît, déclencheront, selon les résultats obtenus, d'autres tests. Cette organisation relève de la stratégie de diagnostic. Les tests effectués lors de la phase initiale sont en principe des tests orientés bon fonctionnement (BF) tandis que durant la deuxième phase, des tests orientés bon ou mauvais fonctionnement (MF) peuvent être déclenchés.

Développer une stratégie pour un système de diagnostic consiste principalement à concevoir une politique de déclenchement des tests de détection. La décomposition d'une stratégie de diagnostic en deux étapes est naturelle à l'instar du diagnostic médical. La première étape correspond à l'analyse des symptômes fournis par les tests exclusivement basés sur des connaissances de comportement normal. La deuxième étape se base sur les tests de détection qui supposent des comportements anormaux spécifiques. En effet, il est plus rentable de commencer par déterminer ce qui se comporte normalement avant de chercher des défauts spécifiques, parce que le nombre de défauts envisageable peut-être très grand (MF) alors que le nombre de composants est finis (BF). Le MF n'apporte que peu d'info car on ne peut que rejeter certaines hypothèses de défaut de manière sûr (exonération) : impossible de prouver avec le MF que l'on est en présence d'un défaut spécifique.

De plus, dans le cadre des systèmes industriels tels que ceux que nous avons abordés dans le projet MAGIC, pour des raisons liées à des capacités de calcul, tous les tests ne peuvent pas être actifs en permanence. L'objectif consiste à déterminer quels sont les tests de détection les plus intéressants à déclencher [Touaf et al, 2003 (a)].

IV.8.2. Stratégie orientée bon fonctionnement

Pour l'architecture d'un système illustré par le graphe structurel de la figure 4.6 deux étapes stratégiques sont distinguées durant la procédure de diagnostic orientée bon fonctionnement (comportement normal). La première étape est basée sur des tests permanents qui représentent les tests de détection de niveau 0 et la deuxième étape est composée de tests de niveau 1 déclenchés par des inconsistances détectées par les tests permanents. Des niveaux plus hauts peuvent exister (voir la stratégie 4) de la figure 4.7.

Un exemple d'architecture de système est présenté par le graphe structurel de la figure 4.6, les variables physiques sont représentées par des cercles vides, les variables connues sont représentées par les cercles pleins, et les modèles de comportement sont représentés par les lignes solides. Pour cet exemple, il y a trois tests de détection orientés bon fonctionnement possibles (NBO) (*Normal Behavior Oriented*): Ceux-ci sont représentés par des lignes pointillées et dénotés par T_1, T_2 et T_{12}.

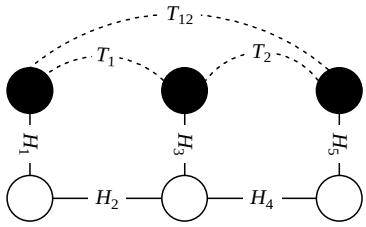


Figure 4.6- Représentation structurelle d'un système physique

Le nombre possible de tests de détection orientés bon fonctionnement (NBO) peut être très grand. Pour les architectures simples comme celle illustrée par la figure 4.6, le nombre de tests possibles de bon fonctionnement est égal à la combinaison suivante C_n^2 où « n » est le nombre de variables connues. Par conséquent, il est intéressant de ne déclencher initialement que certains tests de détection. C'est là où la stratégie de diagnostic doit jouer son rôle.

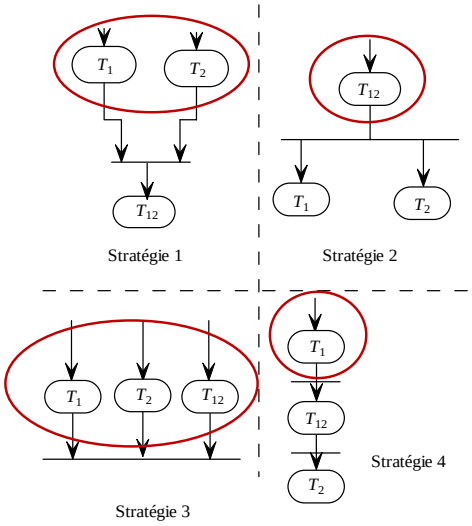


Figure 4.7- Différentes stratégies de test pour l'exemple de la figure 4.6

La stratégie de diagnostic doit être définie séparément du raisonnement diagnostique parce que différentes façons de raisonner peuvent être implémentées par le même système de diagnostic. Par conséquent, les outils utilisés, pour la conception d'une stratégie de diagnostic, doivent être indépendants de l'analyse des symptômes.

Pour l'exemple simple de la figure 4.6, il y a beaucoup de possibilités représentées sur la figure 4.7. Le problème est d'essayer de sélectionner la plus pertinente des stratégies. Comparer les stratégies entre elles, requiert des indicateurs de performance.

IV.8.3. Indicateurs de performances

La performance d'une stratégie dépend :

- ➥ du temps de calcul requis pour exécuter une stratégie
- ➥ de la sensibilité aux défauts
- ➥ de la tolérance aux incertitudes de modélisation

IV.8.3.1. Le Temps de Cycle Normal (TCN)

Le premier estimateur proposé sera appelé le *Temps de Cycle Normal* (TCN). C'est le nombre de tests de détection exécutés quand le système à diagnostiquer est sans défaut. C'est un estimateur important parce qu'il donne une idée du temps de calcul requis lorsque le comportement du système est normal, sachant qu'un test de détection sera supposé prendre une unité de temps. Par exemple, pour la stratégie 1 les tests T_1 et T_2 prennent deux unités de temps. Les résultats pour les 4 stratégies présentées par la figure 4.8 sont exposés par la table suivante.

	Stratégie $_1$	Stratégie $_2$	Stratégie $_3$	Stratégie $_4$
TCN (unités de temps)	2	1	3	1

Table 4.13 – Temps de Cycle Normal (TCN)

En l'absence de défaut, les stratégies 2 et 4 consomment moins de temps que les autres stratégies. En revanche, la 3ème stratégie est la plus lente.

IV.8.3.2. Le Taux de Détection par Niveau (TDN)

La capacité de détection doit être maintenant estimée. Un Taux de Détection par Niveau (TDN) peut être défini. Par exemple, la 1ère stratégie contient 2 niveaux : le niveau 0 pour les tests T_1 et T_2 et le niveau 1 pour le test T_{12}. Au niveau 0, le TDN_0 correspond à la division entre le nombre d'états anormaux détectables et le nombre d'états anormaux possibles. Au niveau l>0, TDN_l correspond à la proportion entre le nombre d'états anormaux détectables par les tests de niveau l avec le nombre d'états anormaux qui auraient été détectés dans le cas où ces tests étaient au niveau 0.

La table suivante résume les liens entre les tests et les états du système qui doit être diagnostiqué :

	H_1	H_2	H_3	H_4	H_5
T_1	1	1	1		
T_2			1	1	1
T_{12}	1	1		1	1

Table 4.14 – Composition des tests orientés bon fonctionnement

Examinons en détail la seconde stratégie. Le test T_{12} détecte les défauts qui satisfont $\neg H_1 \vee \neg H_2 \vee \neg H_4 \vee \neg H_5$ (l'hypothèse H_1 est fausse ou l'hypothèse H_2 est fausse ou…). Seul un état anormal d'un composant est non détecté et il correspond à H_3 : $H_1 \wedge H_2 \wedge \neg H_3 \wedge H_4 \wedge H_5$. Le TDN_0 est égal à 30/31 (il y a 2^5-1 états anormaux possibles). Le TDN_1 est inchangé parce qu'au niveau 1, les défauts détectés satisfont :

$$\begin{aligned}&(\neg H_1 \vee \neg H_2 \vee \neg H_4 \vee \neg H_5) \wedge \\ &\quad ((\neg H_1 \vee \neg H_2 \vee \neg H_3) \vee (\neg H_3 \vee \neg H_4 \vee \neg H_5)) = \\ &= (\neg H_1 \vee \neg H_2 \vee \neg H_3 \vee \neg H_4)\end{aligned} \tag{4.30}$$

Et parce que les tests T_1 et T_2 détecteraient 31 états anormaux s'ils étaient au niveau 0.

La table suivante résume les résultats pour les différentes stratégies :

Stratégies	TDN_0	TDN_1	TDN_2
1	31/31	30/30	-
2	30/31	30/31	-
3	31/31	-	-
4	28/31	27/30	24/28

Table 4.15 – Taux de Détection par Niveau (TDN)

Le nivaux 0 fournit les résultats les plus importants. Dans les stratégies 1 et 3, tous les états anormaux seront détectés. Dans la stratégie 2, un défaut sur H_3 ne sera pas détecté. Dans la stratégie 3 et 4 les états anormaux ne seront jamais détectés.

Le Taux de Détection par Niveau *(TDN)* pour les autres niveaux fournissent de l'information sur la capacité de la détection d'autres niveaux. Si les scores sont grands, cela veut dire que la stratégie choisie ne diminue pas les capacités du test de détection.

Par exemple, il y a seulement 27 états anormaux qui peuvent être détectés au niveau 1 de la stratégie 4, alors que 30 états anormaux pourraient être détectés si le test T_{12} avait été exécuté au niveau 0.

Considérons maintenant la robustesse de la stratégie de diagnostic. La stratégie 3 est plus robuste que la première. S'il y a une grande incertitude sur le comportement du composant relié à l'hypothèse H_3, chaque test contenant H_3, à savoir les tests T_1 et T_2, sera imprécis et ne sera pas capable de détecter quelque chose.

IV.8.3.3. Le Taux de Détection par Niveau avec Incertitudes (TDNI)

Pour estimer la robustesse, les TDN_l sont recalculés successivement en prenant en compte les grandes incertitudes sur les composants reliés à H_i. Autrement dit, les incertitudes sur les modèles des composants sont considérées comme suffisamment importantes pour que les tests qui contiennent ces modèles ne détectent rien.

Soit I_i une grande incertitude sur le composant relié à H_i. Recalculer les TDN_0 pour la 2ème stratégie quand une grande incertitude est présente sur H_1 mène à écrire ce qui suit :

$$TDN_0^{I_1} = \left(\neg H_1 \vee \neg H_2 \vee \neg H_4 \vee \neg H_5\right)\ldots \\ \ldots \wedge \neg I_1 \wedge \neg I_2 \wedge \neg I_4 \wedge \neg I_5 = 0 \tag{4.31}$$

De la même façon, si le composant en rapport avec H_2 est incertain ce qui veut dire que I_2 est vrai alors TDN_0=0; si I_3 est vrai alors TDN_0=30/31; si I_4 est vrai alors TDN_0=0 et si I_5 est vrai alors TDN_0=0. La valeur moyenne de tous ces TDN_0 représente le Taux de Détection du Niveau 0 avec des Incertitudes. $TDNI_0$. Pour la deuxième stratégie, le résultat suivant est obtenu :

$$TDNI_0 = \frac{0+0+30+0+0}{5} = 6 \tag{4.32}$$

De la même façon, les valeurs moyennes des différentes incertitudes sur H_i sont alors évaluées. Le Taux de Détection par Niveau avec Incertitudes ($TDNI_I$) est donc obtenu.

Stratégies	$TDNI_0$	$TDNI_1$	$TDNI_2$
1	22.4/31	0/30	-
2	6/31	0/31	-
3	28,4/31	-	-
4	11.2/31	0/30	0/28

Table 5.16 – Taux de Détection par Niveau avec Incertitudes

Dans le cas général, chaque test de détection en rapport avec les hypothèses H_i où $i \in \Omega$ détectera les états anormaux qui satisfont l'expression suivante :

$$\bigvee_{i\in\Omega}(\neg H_i) \wedge \bigwedge_{i\in\Omega}(\neg I_i) \tag{4.33}$$

TDN_l et $TDNI_l$ résultent du nombre d'états anormaux satisfaisant la disjonction et/ou la conjonction de ce type d'expression, suivant le schéma de déclenchement des tests.

La deuxième et la quatrième stratégie sont les plus rapides. Cependant, les deux stratégies ne permettent pas la détection de tous les états anormaux. La seconde stratégie a la meilleure capacité de détection. En revanche, la 4ème stratégie qui a été conçue pour être une stratégie factice a un meilleur taux de détection moyen en présence de grandes incertitudes.

La 1ère stratégie est la plus lente mais elle détecte tous états anormaux. La 2ème stratégie est la plus lente; elle détecte aussi tous états anormaux mais elle a de meilleurs résultats que la 4ème stratégie en présence des grandes incertitudes.

Grâce à ces trois estimateurs, l'utilisateur fera une demande de diagnostic qui lui permettra de tenir compte de la rapidité d'exécution, de la précision du diagnostic ainsi que la robustesse attendue. Les trois estimateurs de performance proposés sont donc des outils pour sélectionner la meilleure stratégie.

IV.8.4. Stratégie orientée mauvais fonctionnement

Dans un système de diagnostic, les tests de détection qui intègrent dans leur construction des modèles de mauvais fonctionnement sont généralement disponibles. Tous ces tests devraient être déclenchés car quelques-uns d'entre eux peuvent être incompatibles avec les résultats des tests orientés bon fonctionnement. Pour répondre à la problématique, il est important de définir des règles de déclenchement des tests.

La composition des tests orientés mauvais fonctionnement peut être décrit par la table 4.16. Les hypothèses du mauvais fonctionnement d'un composant sont représentées par des colonnes. Un composant peut avoir une hypothèse de bon fonctionnement et d'autres hypothèses de mauvais fonctionnement. Pour signaler la relation entre les deux états normaux et anormaux, des références aux composants ont été ajoutées. Par exemple, l'hypothèse H_1 normal en rapport avec le composant C_1 sera écrite par $H_1(C_1)$. La Table 4.16 regroupe des hypothèses sur des composants différents et sur les mêmes composants.

	$H_1(C_1)$	$D_1(C_1)$	$H_2(C_2)$	$H_3(C_3)$	$D_1(C_3)$	$D_2(C_3)$	$H_4(C_4)$	$H_5(C_5)$
T_4		1	1	1			1	1
T_5	1		1		1		1	1
T_6	1		1			1	1	1
T_7		1	1		1		1	1

Table 4.16 – Les hypothèses et les tests orientés mauvais fonctionnement

A titre d'exemple, si le test de détection T_{12} est faux, sachant qu'il vérifie l'hypothèse suivante $H_1(C_1) \wedge H_2(C_2) \wedge H_4(C_4) \wedge H_5(C_5)$ d'après la table 4.14, alors l'expression suivante $\neg H_1(C_1) \vee \neg H_2(C_2) \vee \neg H_4(C_4) \vee \neg H_5(C_5)$ est vraie. Par conséquent, tout test orienté mauvais fonctionnement compatible avec cette dernière expression doit être déclenché. Le test T_4 contient aussi l'hypothèse $D_1(C_1) \subset \neg H_1(C_1)$. Le test T_7 contient aussi l'hypothèse $D_1(C_1) \subset \neg H_1(C_1)$. Donc, dans ce cas de figure, seulement les tests T_4 et T_7 doivent être déclenchés parce qu'ils peuvent compléter les résultats des tests de bon fonctionnement.

Dans le cas général, un test orienté mauvais fonctionnement est compatible avec un test orienté bon fonctionnement décrit avec l'expression suivante : $\bigvee_{i \in \Omega} \neg H_i(C_i)$ s'il contient au moins une hypothèse de défaut $D_j(C_j)$ avec $j \in \Omega$.

Tout test compatible avec au moins un test faux orienté bon fonctionnement devrait être déclenché parce qu'il peut apporter plus d'information sur l'état réel du système à diagnostiquer.

IV.8.5. Application sur le système des deux bacs

Reprenons l'exemple des deux bacs (chapitre 2). La table 4.17 regroupe les différents tests orientés bons ou mauvais fonctionnements ainsi que leurs hypothèses sur l'état des composants constituant le système des bacs. Si le composant C_i est dans un état normal alors il sera décrit par $H_i = \neg AN(C_j)$.En revanche, si le composant est défaillant avec un défaut D, il sera décrit par l'hypothèse suivante : $H_i = D(C_j)$.

	¬AN(C1)	*¬AN(C2)*	*D(C2)*	*¬AN(C3)*	*¬AN(C4)*	*¬AN(C5)*	*¬AN(C6)*	*¬AN(C7)*	*¬AN(C8)*
T_1	1	1		1			1	1	
T_3				1	1	1		1	1
T_4	1	1		1	1	1	1		1
T_2	1		1	1			1	1	
T_5	1		1	1	1	1	1		1

Table 4.17 - Composition des tests Orientés Bon fonctionnement et Mauvais fonctionnement

Pour cette application nous avons choisi trois stratégies possibles (voir figure 4.8) de diagnostic parmi d'autres. Le choix de la stratégie la plus performante dépend essentiellement des indicateurs de performances définies précédemment. Grâce à ces estimateurs, l'ingénieur de maintenance décrit sa procédure de diagnostic en tenant compte de la rapidité de l'exécution des tests de détection ainsi que la robustesse de la stratégie choisie et l'exactitude du diagnostic attendue.

Les tests T_1, T_3 et T_4 sont orientés bon fonctionnement car ils ne font pas intervenir l'hypothèse du mauvais comportement du composant C_2. Par ailleurs, les tests T_2 et T_5 sont des tests de mauvais fonctionnement car ils font référence à l'hypothèse du défaut $D(C_2)$ qui concerne le composant C_2.

	TCN	TDN_0	TDN_1	$TDNI_0$	$TDNI_1$
Strategie 1	2	255/255	254/254	186/255	0/255
Strategie 2	1	254/255	254/255	31.75/255	0/255
Strategie 3	3	255/255	-	217,75/255	-

Table 4.18 – Les estimateurs

Le TDN_0 examine l'ensemble des stratégies qui fournissent le maximum d'information concernant les défauts.

La stratégie 1 et la stratégie 3 permettent la détection de tous les états anormaux du système des bacs. Autrement dit, le défaut sur le composant C_7 n'est pas détecté par la stratégie 2.

Le TDN_1 confirme que la stratégie 1 est meilleure que la stratégie 2 et 3 en l'absence de défauts, parce que cette approche permet la détection d'autres composants défectueux du système.

La 2ème stratégie est la plus rapide. Cependant, elle ne détecte pas tous les états anormaux, la 3ème a une meilleure capacité de détection. La 3ème stratégie a aussi une meilleure moyenne de taux de détection que la 2ème stratégie dans le cas où il y a présence de grandes incertitudes. La 3ème stratégie est plus lente que les deux autres stratégies (1 et 2) mais elle détecte tous états anormaux. En plus, elle a de meilleurs résultats que les deux autres en présence de grandes incertitudes.

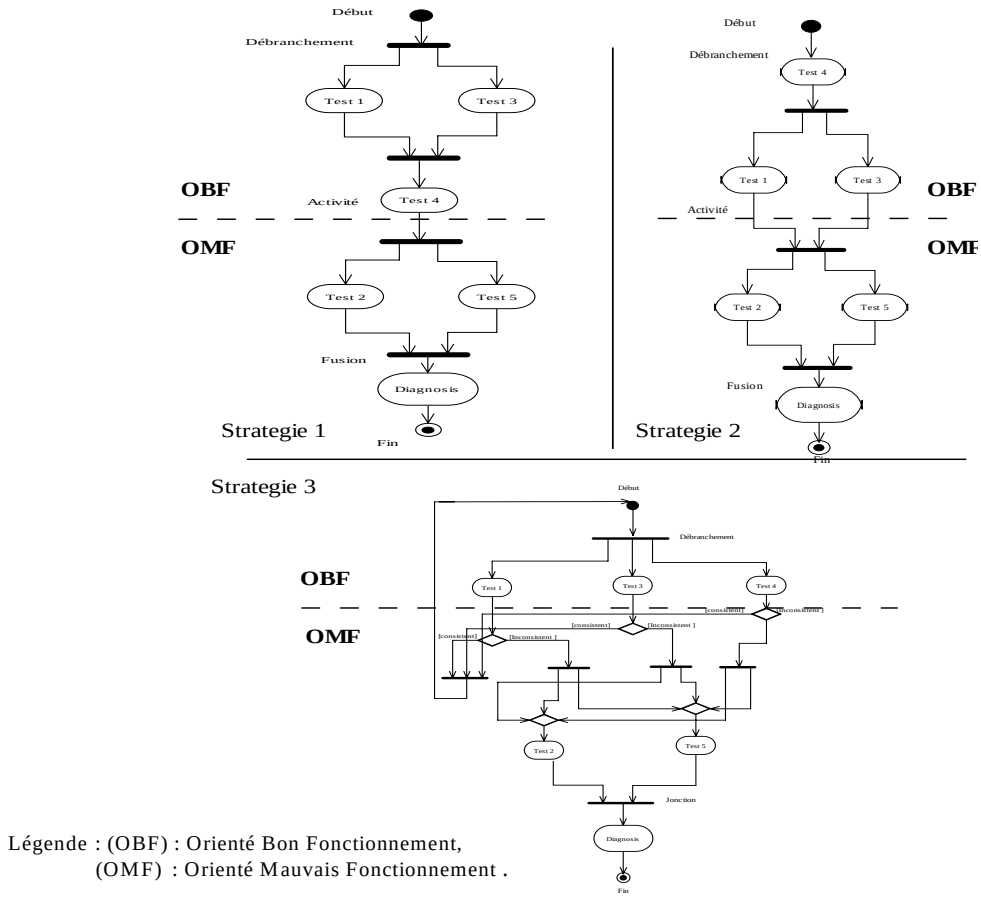


Figure 4.8 –Différentes stratégies de diagnostic

IV.9. CONCLUSION

L'approche présentée dans ce chapitre propose une méthode générique de diagnostic de défauts fondée sur un raisonnement diagnostique logique. L'approche bénéficie à la fois des outils développés par la communauté DX, notamment l'analyse diagnostique avec un algorithme systématique pour trouver les diagnostics minimaux et de ceux développés par la communauté FDI, c'est-à-dire la puissance des algorithmes de détection qui permettent de détecter des défauts sur des systèmes complexes pouvant intégrer des équations différentielles.

Elle se différencie des autres approches existantes par plusieurs points (voir tableau 4.19). Du point de vue de la modélisation des systèmes complexes et dynamiques, notre approche, dans la description du système, englobe le bon et le mauvais fonctionnement simultanément. Elle explicite pour chaque modèle élémentaire, l'état du composant modélisé. Elle intègre les conditions de validité des modèles. En effet, l'approche proposée a besoin de formaliser clairement toutes les hypothèses liées à l'état des composants du système à diagnostiqué. C'est donc plus difficile et consomme plus de temps de calcul que la technique classique utilisant la table des signatures. En revanche, le diagnostic qu'elle produit est exhaustif et précis. Il fournit plusieurs diagnostics possibles, y compris des diagnostics multiples. De plus, nous avons proposé une décomposition de la procédure de diagnostic en trois phases principales : la phase de génération de symptômes par le biais des tests de détection, puis la phase de l'analyse diagnostique, appelée la phase de la localisation des défauts et finalement la phase stratégie de diagnostic. En effet, la stratégie de diagnostic a pour souci l'organisation des tâches de génération de symptômes et de raisonnement

diagnostique pour être conforme à des exigences de précision des résultats et de rapidité de la procédure de diagnostic. Des indicateurs de performance permettent de choisir la meilleure stratégie de diagnostic. Par ailleurs, notre approche, permet de fournir un ensemble de diagnostics exhaustif sous la forme d’état de composant suivant un ordonnancement par ordre de vraisemblance.

Dans le but de développer une application générique et distribuée fondée sur des systèmes multi-agents alors la prise en compte des imprécisions qui affectent les symptômes a été étudiée. Le système multi-agent permettra d’implémenter les tâches principales d’une procédure de diagnostic à savoir la génération de symptômes, le raisonnement diagnostique, et la stratégie de diagnostic. Nous avons mis en exergue les limites des approches de décision binaire pour introduire les principaux outils théoriques permettant d’aborder les incertitudes de la décision des tests de détection. Nous avons établi certaines propriétés qui nous semblent utiles à une application au diagnostic distribué. Les résultats ont été déduits dans les deux cas de figure, le cas de l’analyse structurelle qui vient de la communauté FDI et dans l’autre cas dans l’analyse logique qui vient de la communauté DX. L'algorithme proposé gère aussi les incertitudes de décision en transposant la logique nette du raisonnement formel en logique floue. En effet, les résultats présentés sont une inférence de la définition d’un test de détection avec la logique des prédicats en utilisant les propriétés de la logique floue. Les résultats fournis dans ce chapitre nous permettent de gérer la situation où le défaut n'est pas complètement confirmé.

PARTIE II

SOLUTION TECHNOLOGIQUE POUR LE DIAGNOSTIC DISTRIBUE

Chapitre 5

Les systèmes multi-agents

V.1. INTRODUCTION

Le chapitre 1 dans la première partie de ce document nous a permis de mettre en évidence la problématique du diagnostic des systèmes complexes dans un contexte distribué et nous avons vu l'intérêt d'avoir un système de diagnostic distribué par rapport à un système centralisé. Les systèmes multi-agents, faisant l'objet d'intérêts croissants, semblent particulièrement bien adaptés à la problématique (cf. chapitre 1). L'idée est de multiplier les entités intelligentes et de les doter de facultés de communication afin qu'elles aient le moyen de communiquer entre elles pour que se construise par coopération une solution globale à un problème donné. Dans notre cas, le problème est de diagnostiquer un système physique. Pour résoudre des problèmes complexes, il peut être nécessaire et naturel de faire intervenir plusieurs points de vue, c'est à dire de faire coopérer différents experts utilisant des stratégies de résolution et des sources de connaissances distinctes. Chacun d'eux n'a qu'une vue partielle du problème et tend à atteindre un but qui peut être cohérent avec l'objectif général à atteindre.

Les approches multi-agents permettent de mettre en œuvre des solutions calquées sur l'organisation humaine, plutôt que d'envisager une seule entité omnipotente et omnisciente. De ce fait, les solutions d'organisation sociétale complexe mises au point au fur et à mesure des années par les hommes peuvent être adaptées à la résolution de problèmes. Le premier avantage est de conduire à des systèmes modulaires et ouverts où le fait d'ajouter un agent ou de modifier la structure d'un système n'induit pas une re-conception d'une solution mais converge automatiquement, à la suite d'un processus d'apprentissage, vers une nouvelle solution globale. L'autre grand avantage est de permettre de résoudre des problèmes complexes en les décomposant en une multitude de problèmes plus élémentaires, en construisant des entités autonomes qui pourront, en coopérant, participer à la construction d'une solution globale.

Ce chapitre donne un panorama sur les systèmes multi-agents. Nous décrivons les propriétés et lois qui émergent d'un système multi-agent. Nous commençons par décrire l'aspect individuel (le comportement d'un agent) ensuite par étudier l'aspect collectif (comportement d'une société d'agents). Les interactions entre ces différentes entités intelligentes peuvent faire apparaître des phénomènes nouveaux comme l'émergence. En effet, chaque agent peut être spécialiste dans son domaine de compétence dans lequel il peut intervenir sans pour autant trouver la solution globale. En revanche, l'interaction entre les différents agents du système fait qu'il y a émergence de la solution globale.

V.2. AGENTS ET SYSTEMES MULTI-AGENTS

V.2.1. Définition d'un agent

Il n'y a pas une définition acceptée à l'unanimité pour la notion d'agent. Dans ce qui suit, on présente les définitions les plus courantes.

- « Un agent est une entité qui perçoit son environnement et agit sur celui-ci » [Russell, 1997].
- Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome pour atteindre les objectifs (buts) pour lesquels il a été conçu [Wooldrige et Jennings, 1995].
- « Les agents intelligents sont des entités logiciels qui réalisent des opérations à la place d'un utilisateur ou d'un autre programme, avec une sorte d'indépendance ou d'autonomie, et pour faire cela ils utilisent une sorte de connaissance ou de représentation des buts ou des désirs de l'utilisateur » [L'agent IBM].
- « Un agent est une entité qui fonctionne continuellement et de manière autonome dans un environnement où d'autres processus se déroulent et d'autres agents existent.» [Shoham, 1993].
- Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents [Ferber, 1995].

En partant de l'ouvrage de [Wooldrige et Jennings, 1995], et des définitions citées, on peut identifier les caractéristiques suivantes pour la notion d'agent :

Situé – l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement ;

Autonome – l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne ;

Proactif – l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment ;

Capable de répondre à temps – l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis ;

Social – l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

Intentionnalité - un agent intentionnel est un agent guidé par ses buts, une intention est la déclaration explicite des buts et des moyens d'y parvenir. Elle exprime donc la volonté d'un agent d'atteindre un but ou d'effectuer une action.

Rationalité - un agent rationnel est un agent qui suit le principe suivant "Si un agent sait qu'une de ses actions lui permet d'atteindre un de ses buts, il la sélectionne". La notion de rationalité se rapporte au comportement cognitif de l'agent. Ce terme qualifie l'utilisation efficace des ressources par l'agent.

Engagement - La notion d'engagement est une qualité essentielle des agents coopératifs. Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents. En construisant un plan pour atteindre un but, l'agent se donne les moyens d'y parvenir et donc

s'engage à accomplir les actions qui satisfont ce but : l'agent croit qu'il est en mesure d'exécuter tout le plan qu'il a élaboré, ce qui le conduit (ainsi que les autres agents) à agir en conséquence.

Adaptabilité - Un agent adaptatif est un agent capable de contrôler ses aptitudes (communicationnelles, comportementales) selon l'agent avec qui il interagit. Un agent adaptatif est un agent d'un haut niveau de flexibilité.

Intelligence -Un agent intelligent est un agent cognitif, rationnel, intentionnel et adaptatif.

V.3. DIFFERENTS TYPES D'AGENTS

La conception du raisonnement d'un agent introduit automatiquement des approches de traitement des connaissances. Pour simplifier notre présentation, nous décrirons des raisonnements « réflexes » ou « réactifs » et des raisonnements de plus haut niveau (appelés raisonnements cognitifs). Le choix du type de raisonnement proposé par le concepteur est basé sur un large éventail de possibilités. Certes, il est évident que plus le raisonnement des agents est simple, plus les interactions entre agents devront être performantes afin de maîtriser la cohérence globale du système.

V.3.1. Agents réactifs

Les agents réactifs**,** sont des entités extrêmement simples ne disposant de presque aucune capacité de raisonnement, n'ayant pas de mémoire, n'étant pas capables de se représenter l'environnement, sont de plus bas niveau, ne disposent que d'un protocole et d'un langage de communication réduit, ne se comportant que sur le mode stimulus–réponse, mode situé très localement. Un agent réactif est un agent de faible granularité. Les premiers travaux relatifs à cette approche ont été réalisés au MIT en 1986 par R.BROOKS [Herceau et Ferbert, 1991].

Les agents réactifs sont souvent qualifiés de non-« intelligents ». Ils sont des composantes très simples qui perçoivent l'environnement et sont capables d'agir sur celui-ci. Ils n'ont pas une représentation symbolique de l'environnement ou des connaissances et ils ne possèdent pas de croyances, pas de mécanisme d'envoi de messages. Leurs capacités répondent uniquement au mode stimulus/action qui peut être considéré comme une forme de communication.

Un SMA constitué d'agents réactifs possède généralement un grand nombre d'agents et présente un comportement global intelligent. Les agents réactifs sont considérés intelligents au niveau du groupe. En conséquence, l'intelligence est distribuée entre beaucoup d'agents réactifs. Le comportement intelligent devrait émerger de l'interaction entre ces agents réactifs et l'environnement qui les entoure. L'architecture réactive la plus connue et la plus influente est celle proposée par Rodney Brooks ; elle s'appelle architecture de subsomption, en anglais "*subsumption architecture*". Une architecture de subsomption comporte plusieurs modules, chaque module étant responsable de la réalisation d'une tâche simple. Ces modules correspondent à des comportements spécifiques pour accomplir une tâche particulière, et s'appellent modules de compétence. La figure 5.1 montre une telle architecture.

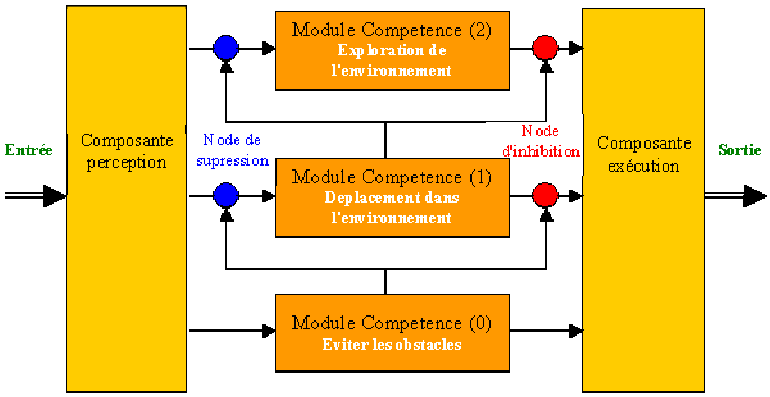


Figure 5.1- Architecture réactive de subsomption

Les architectures réactives ont l'avantage de la simplicité et de l'efficacité de calcul. Pourtant, elles présentent plusieurs limitations, ce qui fait que ces architectures ne peuvent pas être utilisées dans de nombreuses classes d'applications. Les principales objections contre l'utilisation d'architectures réactives sont les suivantes : Les agents ont une vision de courte durée sur la résolution du problème, et ils ne peuvent pas toujours choisir la meilleure action à exécuter à un certain moment ; comme les agents réactifs ne maintiennent pas une représentation de l'environnement, ils ne peuvent pas avoir des buts ; dans ce cas, même le grand avantage de cette architecture, l'émergence de l'intelligence par interaction, peut devenir un désavantage, car des interactions indésirables peuvent émerger.

V.3.2. Agents cognitifs

A l'opposé, les autres agents, dits cognitifs**,** sont plus 'intelligents'. Plus complexes, ils peuvent se construire une représentation de l'environnement, ils possèdent une base de connaissances et ils sont aptes à mener un raisonnement étayé sur cette dernière, des capacités de raisonnement, ils peuvent tenir compte de leur passé et ainsi anticiper sur l'avenir pour planifier leurs actions. Ils peuvent aussi connaître les états mentaux des autres agents. Comme exemple de système multi-agent cognitifs nous citerons la plate forme de développement de système multi-agent SARA (système multi-agent pour le filtrage d'alarmes) [Khoualdi, 1994], CONDOR (système multi-agent pour le support des activités de conception de produits) [Iffencker, 1992].

C'est l'école cognitive qui, jusqu'à maintenant, a donné lieu aux applications les plus avancées. Toutefois, pour avoir le meilleur des deux solutions, à savoir architectures cognitives et réactives, les chercheurs ont conçu des architectures hybrides qui combinent les caractéristiques de ces deux architectures. Il est possible de concevoir des systèmes hétérogènes comportant les deux types d'agents. Il est aussi possible de doter les agents cognitifs de capacités de réactions aux événements : on parlera alors d'agents hybrides.

De nombreux travaux de recherche peuvent se classer dans ce type d'approche, notamment l'approche BDI qui repose sur le modèle "Croyance-Désir-Intention", en anglais "*Belief-Desire-Intention*". Dans ce qui suit, on va présenter d'une manière informelle, intuitive, la signification de ces trois éléments dans un modèle BDI.

- ***Le B = Belief (Croyance)*** **-** Les croyances d'un agent sont les informations que l'agent possède sur l'environnement et sur d'autres agents qui existent dans le même environnement. Les croyances peuvent être incorrectes, incomplètes ou incertaines et, de ce fait, elles sont différentes des connaissances de l'agent, qui sont des informations toujours vraies. Les croyances peuvent changer au fur et à mesure que l'agent, par sa capacité de perception ou par l'interaction avec d'autres agents, recueille plus d'information.

- ***Le D = Desire (Désir)*** **-** Les désirs d'un agent représentent les états de l'environnement, et parfois de lui-même, que l'agent aimerait voir réalisés. Un agent peut avoir des désirs contradictoires ; dans ce cas, il doit choisir parmi ses désirs un sous-ensemble qui soit consistant. Ce sous-ensemble consistant de ses désirs est parfois identifié avec les buts de l'agent.

- ***Le I = Intention (Intention)*** **-** Les intentions d'un agent sont les désirs que l'agent a décidé d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs. Même si tous les désirs d'un agent sont consistants, l'agent peut ne pas être capable d'accomplir tous ses désirs à la fois.

L'exemple suivant va apporter plus de clarification à ce modèle. L'agent Pierre croit que, si quelqu'un passe son temps à étudier, cette personne peut faire une thèse de doctorat. En plus, Pierre a le désir de voyager beaucoup, de faire une thèse de doctorat et d'obtenir un poste d'assistant à l'université. Le désir de voyager beaucoup n'est pas consistant avec les deux autres et Pierre, après réflexion, décide de choisir, parmi ces désirs inconsistants, les deux derniers. Comme il se rend compte qu'il ne peut pas réaliser ses deux désirs à la fois, il décide de faire d'abord une thèse de doctorat. En ce moment Pierre a l'intention de faire une thèse et, normalement, il va utiliser tous ses moyens pour y parvenir. Il serait irrationnel de la part de Pierre, une fois sa décision prise, d'utiliser son temps et son énergie, notamment ses moyens, pour voyager autour du monde. En fixant ces intentions, Pierre a moins de choix à considérer car il a renoncé à faire le tour des agences de voyage pour trouver l'offre de voyage qui le satisferait au mieux.

La figure suivante présente les composantes principales d'une architecture BDI.

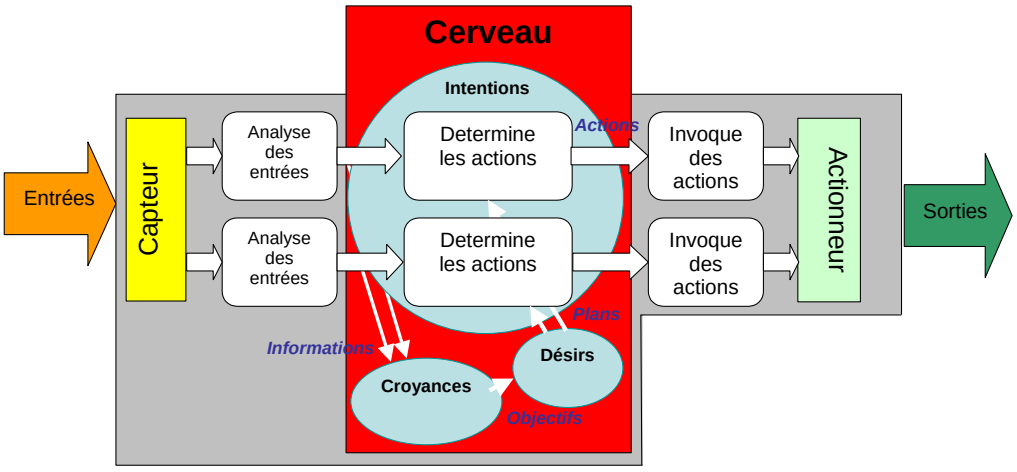


Figure 5.2- Architecture BDI d'un agent

L'agent a une représentation explicite de ses croyances, désirs et intentions. Les ensembles des croyances de l'agent, ses désirs et ses intentions peuvent être représentés au moyen de divers

modèles de représentation de connaissances, par exemple en utilisant la logique des prédicats du premier ordre, une logique d'ordre supérieur, le modèle des règles de production, ou bien comme de simples structures de données.

Le tableau 5.1 résume les différences entre les agents cognitifs et les agents réactifs.

Système d'agents cognitifs	**Système d'agents réactifs**
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement Stimulus/Réponse
Petit nombre d'agents	Grand nombre d'agents

Tableau 5.1-Différences entre agents cognitifs et agents réactifs

V.3.3. Fonctionnement d'un agent

L'architecture fonctionnelle d'un agent cognitif est illustrée par la (figure 5.3). Les agents sont immergés dans un environnement dans lequel et avec lequel ils interagissent. Leur structure axée autour de trois fonctions principales : percevoir, décider et agir.

Un agent a la possibilité d'acquérir des connaissances sur son environnement (perception). Il a aussi des capacités d'interaction avec les autres agents (communication). En fonction des connaissances et des croyances dont il dispose, et des buts qu'il se fixe suite à une perception ou à une interaction avec le monde extérieur ; l'agent doit élaborer un plan d'action. Pour cela, il doit décider du but à retenir et à satisfaire en premier, planifier en fonction de ce but et passer à l'exécution [Demazeau et Muller, 1990] [Labidi et Lejouad, 1993].

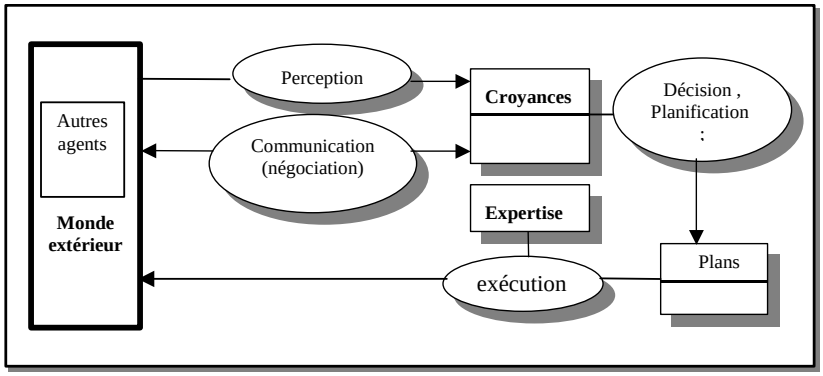


Figure 5.3 - Structure de Fonctionnement de l'agent

Les différents modules représentés par la figure 2.3 sont présentés ci-dessous :

Perception - Les connaissances d'un agent ont plusieurs origines :

- Le savoir initial de l'agent ;
- La perception du soi et du monde (ses accointances et son environnement) ;
- La communication avec les autres agents.

Les informations issues de la perception et du savoir initial de l'agent sont considérées comme des connaissances certaines puisqu'elles n'ont subi aucune mise à jour, alors que les

connaissances provenant des autres agents sont considérées comme incertaines, puisqu'elles évoluent sans que l'agent en soit forcément informé. On peut pour cela associer, à chaque connaissance, son origine afin d'en évaluer la crédibilité et d'en permettre la vérification.

Prise de décision - Durant son exécution, un agent se fixe un certain nombre de buts, suite à ses observations et à ses interactions avec le monde (perception, communication, négociation). Il se trouve donc confronté au problème de la sélection du but à satisfaire en premier et pour chaque but, de l'action (ou des actions) qui permet de l'atteindre. La prise de décision est une des caractéristiques des agents rationnels, l'agent tiendra compte de ses croyances pour faire son choix.

Planification - La planification est le processus qui permet de déterminer la séquence d'actions permettant d'atteindre un but ; on parle de plan d'actions. Ce plan est constitué d'intentions, d'engagements, et de croyances sur l'univers qui l'entoure.

V.4. DEFINITION D'UN SYSTEME MULTI-AGENT

L'agent décrit ci-dessus est destiné à agir au sein d'un groupe d'autres agents, dénommé système multi-agent (SMA). Un système multi-agent est « un ensemble d'entités qui coordonnent leurs connaissances, buts, expériences et plans pour agir ou résoudre des problèmes, incluant le problème de la coordination inter-agent lui-même » [Mandiau et al, 2000 (a)]. Un système multi-agent peut donc être défini comme une entité logicielle composée de plusieurs agents intelligents (cf. figure 5.4).

Les agents composent un « système » car ils constituent un ensemble cohérent autour d'un « objet » commun. L'objet qui les relie peut prendre diverses formes : ce peut être un objectif commun (par exemple, diagnostiquer un système physique), ou un langage commun, le minimum étant un environnement commun. L'ensemble des agents forme un groupe parce qu'ils interagissent. Selon les objectifs qu'ils poursuivent et leurs capacités à les atteindre, l'interaction entre les membres du groupe peut être de type coopératif, conflictuel, ou tout autre type de relation intermédiaire entre ces deux extrêmes.

Alors que la description d'un agent donne une vision locale, la description du SMA (système multi-agent) offre une vue globale et externe. Pour obtenir une vision complète, il faut ajouter la description intermédiaire qui concerne l'aspect « social », interne au système, des interactions entre les agents. C'est par l'intermédiaire de cette couche d'interactions que les comportements individuels interfèrent et composent le comportement global du système. Inversement, le comportement global influe sur les comportements individuels via cette couche sociale.

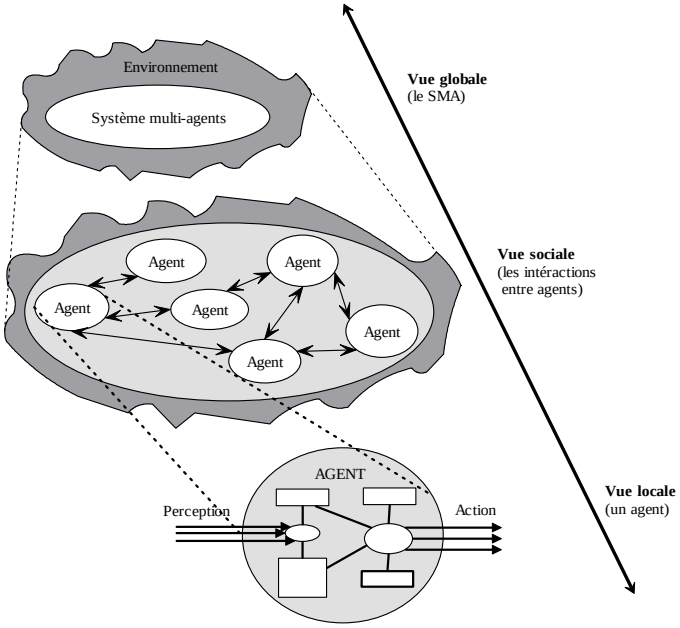


Figure 5.4 - Système multi-agent vu selon différents niveaux de détail

Un système multi-agent est donc constitué d'un ensemble de processus informatiques se déroulant en même temps, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communicant entre eux. Le principal problème à résoudre lors de la conception des systèmes multi-agents réside dans la formulation de la coordination entre les agents. La recherche sur les agents s'intéresse aux mécanismes de décision, de coordination et de communication entre agents.

L'organisation des agents est un cadre organisationnel qui fixe les rôles des agents dans le groupe et donc les règles d'interactions entre eux. Deux modèles d'organisation fondamentalement différente sont souvent distingués dans la littérature.

- *Organisation émergente* - Un premier modèle concerne l'émergence d'organisations. En effet, graduellement, une forme d'organisation se met en place ou, plus précisément, un observateur a l'impression de voir apparaître (émerger) un comportement structuré parmi les agents. Chaque entité, par son action locale, participe au développement de l'ensemble sans qu'aucune d'entre elles ne dirige les autres. Ces systèmes mettent généralement en jeu un nombre assez important d'individus. Leur nombre, ajouté au fait qu'ils n'ont pas conscience de leur appartenance à un groupe, conduit à une grande souplesse du système. C'est l'adaptabilité de chaque agent qui se répercute au niveau supérieur. Il suffit que les conditions environnementales varient pour que chaque individu réagisse différemment et que, par conséquent, des changements soient observables au niveau du groupe.

- *Organisation support d'activités* - Les agents sont intégrés dans une structure organisationnelle. Cette structure est plus ou moins flexible et autoadaptative [Mandiau et al, 2000 (b)], mais les comportements des agents vis-à-vis de leurs congénères sont délimités (déterminés) par leurs rôles respectifs. Ce modèle se rapproche plus des organisations humaines car chaque agent est autonome tout en ayant connaissance de ce qu'il doit faire et comment se comporter avec tel ou tel agent en fonction de leurs rôles respectifs. Ces modes de comportement induisent des organisations moins flexibles que celles proposées par le modèle d'émergence. En revanche, ils

ont l'avantage de représenter de manière explicite les liens entre les agents ou rôles joués par ceux-ci.

V.5. MODELES DE SYSTEMES MULTI-AGENTS

Nous distinguons trois classes de systèmes multi-agents [Haton, 1991] les systèmes à tableaux noirs, les systèmes d'acteurs et les systèmes physiquement distribués.

V.5.1. Les systèmes à tableaux noirs

Dans de tels systèmes, le tableau noir est le seul moyen de coopération [Harbouche, 1993]. La coopération est alors implicite et les agents s'ignorent totalement. Les agents, appelés aussi « sources de connaissances », sont des modules interdépendants renfermant chacun une partie de la connaissance du domaine d'application traité. Dans de tels systèmes, un agent n'a connaissance ni du problème complet, ni de l'état global de la solution. Il réagit uniquement aux informations auxquelles il a accès et ne peut pas raisonner sur sa coopération avec les autres. La communication entres ces différents agents se fait à travers une zone de données commune, appelée **tableau noir (*blackboard*)**. Le système de contrôle est totalement centralisé.

Tout système basé sur le modèle de tableau noir est constitué de trois entités principales : le tableau noir, les sources de connaissances et le mécanisme de contrôle (cf. figure 5.5)

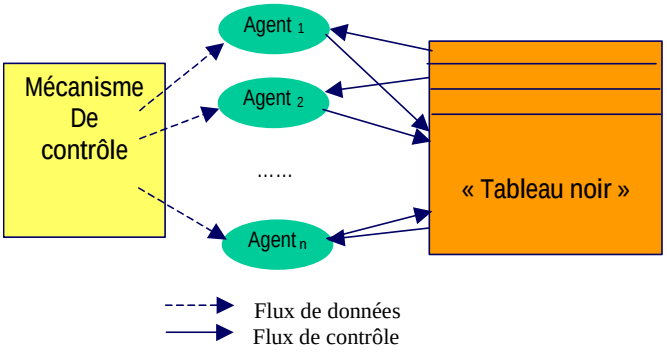


Figure 5.5 - Structure d'un système à tableau noir

- **Les agents** [Delouis, 1991] stockent leurs hypothèses dans un tableau noir. Ces agents représentent des modules indépendants. La coopération entre ces agents s'effectue par des ajouts, des suppressions ou des modifications des éléments du tableau noir. Sous certaines conditions, les agents peuvent reconnaître les situations dans lesquelles ils peuvent intervenir et ainsi faire progresser la solution. En revanche, lorsque l'agent est sélectionné, il peut modifier une certaine zone du tableau. Chaque agent permet de traiter une étape dans la résolution du problème, en manipulant des connaissances au niveau adapté. Un agent est une association (contexte/traitement) ; la partie contexte décrit les conditions sous lesquelles l'agent peut être activé, la partie traitement décrit, le plus souvent sous forme d'une procédure ou d'un ensemble de règles, comment l'agent doit être exécuté.

- **Le tableau noir** contient une description de l'état de la résolution sous formes d'entités appelées souvent faits, hypothèses ou nœuds. Il est décomposé en niveaux qui permettent une

décomposition de la solution en niveaux d'abstraction. Le tableau noir contient donc les données du problème, les solutions partielles et la solution globale du problème à résoudre.

- **Le mécanisme de contrôle** - Les agents doivent répondre de façon opportuniste aux changements du tableau noir. Pour cela, il existe un ensemble de modules de contrôle qui gèrent les opérations réalisées sur le tableau noir et décident de la prochaine action à entreprendre. Leur rôle consiste à déterminer soit un agent à activer soit une région du tableau noir à étudier, soit une combinaison des deux [Delouis, 1991].

Dans ce modèle des SMA, l'expertise est distribuée entre les sources de connaissances tandis que le contrôle de la résolution est centralisé. En général, les avantages reconnus des systèmes basés sur un tableau noir sont essentiellement la possibilité de gérer une très grande quantité d'informations et de décomposer les problèmes, lors de leurs différentes étapes de résolution, en niveaux d'abstractions pour une application donnée. Les systèmes basés sur un tableau noir sont caractérisés par une communication indirecte puisque les différents agents communiquent par l'intermédiaire du mécanisme de contrôle. Ils sont aussi caractérisés par une communication centralisée, puisque ce mécanisme de contrôle active et désactive les différents agents. Cependant, les recherches récentes ont souligné des goulets d'étranglement liés à ce type de mécanisme [Mandiau et al, 2000 (a)].

V.5.2. Les systèmes d'acteurs

A l'inverse des systèmes à tableau noir, les systèmes d'acteurs relèvent d'une distribution totale des connaissances et du contrôle [Ferber, 1995]. Ces systèmes se caractérisent par :

- Un traitement local qui veut dire qu'un agent ne peut manipuler que sa base de connaissance locale, envoyer des messages aux autres agents qu'il connaît (accointances) et créer de nouveaux agents. De ce fait, les connaissances et le comportement sont distribués parmi les agents, qui effectuent des tâches en parallèle et de manière indépendante.
- La seule structure de contrôle est l'envoi de message.

Le modèle acteur est une extension du modèle objet par ajout de la notion d'activité - Chaque objet, appelé ***acteur***, est un agent actif, autonome, communiquant librement avec ses semblables. Les acteurs évoluent dans un univers dynamique, où des activités se créent, se déroulent et s'achèvent. Ils participent à ces activités, et communiquent entre eux pour se confier ou se déléguer des tâches et se transmettre des résultats [Gerald, 1990].

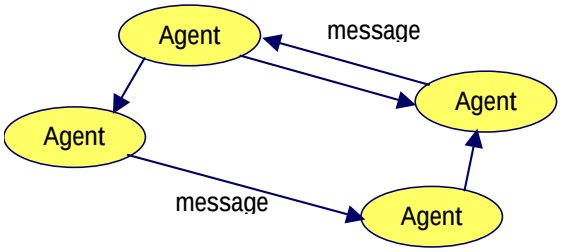


Figure 5.6 –Les systèmes d'acteurs

V.5.3. Les systèmes distribués

Les systèmes distribués sont composés d'agents plus complexes que précédemment, dotés de capacités de raisonnement élaborées [Gerald, 1990]. Généralement, ces agents sont localisés sur des sites physiquement différents. Cela implique la mise en œuvre de stratégies de coordination très élaborées entres les agents. Ces derniers doivent en effet être capables de planifier intelligemment leurs activités locales en fonction de celles des autres.

V.6. COMMUNICATION DANS LES SYSTEMES MULTI-AGENTS

Les communications dans les SMA sont à la base des interactions et de l'organisation. Une communication peut être définie comme une forme d'action locale d'un agent vers d'autres agents. Les questions abordées par un modèle de communication peuvent être résumées par l'interrogation suivante : qui communique quoi, à qui, quand, pourquoi, et comment ?

- *Pourquoi les agents communiquent-ils* ? la communication permet l'interaction entre agents et par conséquent la coopération et la coordination d'actions.
- *Quand les agents communiquent-ils* ? les agents sont souvent confrontés à des situations où ils ont besoin d'interagir avec d'autres agents pour atteindre leurs buts locaux ou globaux. La difficulté réside dans l'identification de ces situations. Par exemple, une communication peut être sollicitée suite à une demande explicite par un autre agent.
- *Avec qui les agents communiquent-ils* ? les communications peuvent être sélectives sur un nombre restreint d'agents ou diffusées à l'ensemble des agents. Le choix de l'interlocuteur dépend essentiellement des accointances de l'agent (connaissances qu'a l'agent sur les autres agents).
- *Comment les agents communiquent-ils* ? la mise en œuvre de la communication nécessite un langage de communication compréhensible et commun à tous les agents. Il faut identifier les différents types de communications et définir les moyens permettant non seulement l'envoi et la réception des données mais aussi le transfert des connaissances avec une sémantique appropriée à chaque type de message.

V.6.1. Comment communiquer ?

Pour coordonner l'activité d'un ensemble hétérogène d'agents autonomes, il faut que les agents communiquent dans un langage compréhensible par les autres. On observe que dans un système ouvert un tel langage peut constituer une interface entre les agents. L'utilisation d'un langage commun implique que tous les agents comprennent son vocabulaire sous tous ses aspects concernant :

- La syntaxe – qui précise le mode de structuration des symboles ;
- La pragmatique – qui permet d'interpréter les symboles ;
- L'ontologie – qui permet d'utiliser les mêmes mots d'un vocabulaire commun.

Pour que la communication s'établisse entre les agents, un Langage de Communication Agent (ACL de l'anglais *Agent Communication Language*) doit être conçu comme un langage de haut niveau qui assure en premier lieu l'échange d'états mentaux et le sens du vocabulaire [Boissier, 2001]. Le format utilisé pour l'échange des connaissances est donné par un langage de contenu, indépendant du langage ACL (exemple : XML). Le vocabulaire commun concerne les définitions précisées dans une ontologie. Ces composants sont représentés dans la figure 5.7.

Figure 5.7 - Modèle des Langages de Communication entre Agents

Si on regarde l'évolution des langages de communication, on constate une tendance à assurer un partage d'information de plus en plus complexe. Au début on peut remarquer que le partage concernait les objets (*Remote Method Invocation, CORBA*); ensuite il a concerné les connaissances (faits, règles, procédures de traitement des connaissances). Les ACL's du présent concernent le partage des états mentaux (croyances, désirs, intentions). On peut citer ici les langages KQML [Labrou et Finin, 1998] [Labrou et al, 1999] FIPA-ACL [FIPA, 1997]. Le développement des spécifications des ACL's tire profit des recherches effectuées sur les langues naturelles, sur la pragmatique conversationnelle et la théorie des actes de langage [Austin, 1962].

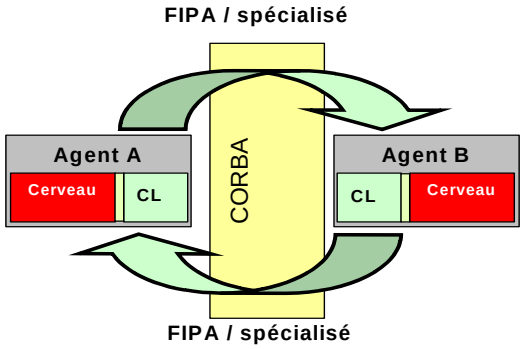


Figure 5.8 - Communication entre deux agents via Corba

V.6.2. Modèles de communication

Les modes et protocoles de communication établissent les moyens et les styles de communication entre agent. Il existe deux formes de communication : Communication par envoi de message et la communication par partage d'information. Ces deux modèles ont été combinés pour donner naissance à un modèle hybride [Labidi et Lejouad, 1993] [Khoualdi, 1994] [Baujard, 1992**]**.

V.6.2.1. Communication par partage d'information

Les agents ne sont pas en liaison directe mais communiquent via une structure de données partagée, ou on trouve les connaissances relatives à la résolution (état courant du problème) qui évolue durant les processus d'exécution. Ce mode de communication est utilisé quand il y a recouvrement des domaines d'expertise. Le meilleur exemple qui utilise ce type de communication est les systèmes à tableau noir (cf. figure 5.9).

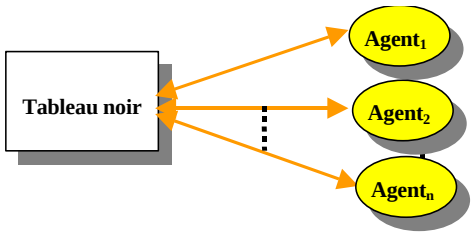


Figure 5.9 - Communication par partage d'informations.

V.6.2.2. Communication par envoi de message

Les agents sont en liaison directe et envoient leurs messages directement et explicitement au destinataire (cf. figure 5.10). Ce mode de communication est utilisé dans les systèmes d'acteurs. La transmission se fait suivant deux modes :

- Transmission point à point - l'agent émetteur connaît et précise l'adresse de ou des agent(s) destinataire(s).
- Transmission par diffusion - le message est envoyé à tous les agents du système.

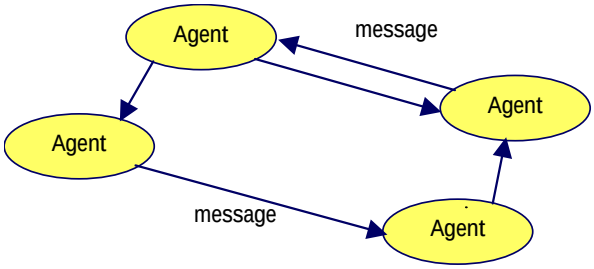


Figure 5.10 - Communication par envoi de messages

- **L'envoi de message** - L'univers des acteurs est complètement homogène. L'acteur est l'unique entité. Un seul type d'événement peut se produire : l'envoi d'un message par un acteur à un autre. Plus précisément, un acteur émetteur transmet à un acteur récepteur un acteur message dont une des accointances (donnée locale) est le message à transmettre [Gerald, 1990]. Ceci se fait d'une manière asynchrone. Ce message peut être une requête, une information ou une suggestion, etc.
- **Envoi de message et continuation** - La notion de continuation est l'un des aspects les plus caractéristiques des modèles d'acteurs [Haton, 1991]. Le message contient une continuation qui désigne l'acteur auquel doit être transmis le résultat du message. Par défaut, la continuation est l'acteur qui a émis le message. L'acteur ayant émis le message n'est pas bloqué et peut continuer son activité.
- **Délégation** - A la réception d'un message, l'agent vérifie si l'expéditeur du message fait bien partie de ses accointances. Si l'expéditeur est inconnu, l'agent ne va pas traiter le message mais

va le déléguer à un autre susceptible de s'y intéresser. Pour cela, il consulte les informations qu'il possède sur le modèle des autres agents [Iffencker, 1992].

V.6.2.3. Modèle hybride

Le principe est de concevoir des systèmes d'agents à base de tableau noir qui communiquent par le mode d'envoi de message [Baujard, 1992].

V.7. LANGAGES DE COMMUNICATION ENTRE AGENTS

Ces langages se focalisent essentiellement sur la manière de décrire exhaustivement des actes de communication d'un point de vue syntaxique et sémantique supportant un langage de représentation des connaissances. Toutefois, l'aspect ontologique et l'utilisation de conventions garantissant un comportement collectif cohérent du système et l'aspect conversationnel n'est pas facile à garantir. Plusieurs tentatives de normalisation de la communication inter-agents ont été effectuées au sein de la communauté multi-agent ces dernières années.

V.7.1. a) Langage de communication KQML (Knowledge Query and Manipulation Language)

KQML est fondé sur la théorie des actes de langage [Austin, 1962] et a été développé au Etats-Unis [Labrou et Fenin, 1997] dans le but de permettre aux agents cognitifs de coopérer. KQML est un langage de communication et un protocole de haut niveau pour l'échange d'information, orienté messages et indépendant de la syntaxe du contenu et de l'ontologie du contenu des messages, indépendant aussi du mécanisme de transport (TCP/IP, e-mail, objets CORBA etc.) et du langage utilisé pour coder le contenu des messages (par exemple : Prolog, STEP, SQL, KIF etc.).

Une ontologie est une spécification ou une vue simplifiée et abstraite du domaine qui sera représenté. En d'autres termes, une ontologie définit le vocabulaire dans un domaine donné pour que les agents puissent se comprendre.

KQML est un acronyme de «*Knowledge Query and Manipulation Language* », conçu comme un format standard de message. Ce format de message autorise un protocole de communication temps réel supportant le partage d'informations entre les agents. Un message KQML peut être vu comme un objet, défini par l'information clé, la «performative ». Ce langage est formé de trois couches :

- Une couche ***communication*** décrivant les paramètres de communication de bas niveau (tels que l'émetteur, le destinataire) ;
- Une couche ***message*** (indiquant un acte communicatif ainsi que son protocole d'interprétation).
- Une couche ***contenu*** (des informations pertinentes pour le message transmis).

La 'performative' est « ask-all » qui signifie que l'agent E (émetteur) désire que tous les agents R (récepteur) répondent à sa question. Les attributs de cette performative sont :

- *content* - le contenu du message (l'information transportée par la performative)
- *reply-with* - identificateur unique du message, en vue d'une référence ultérieure
- *ontology* - précise le nom de l'ontologie utilisé dans contenu
- *language* - le nom du langage utilisé dans le contenu du message (content)

D'autres attributs, largement utilisés, sont :

- *sender* - identifie l'agent émetteur (le nom de l'agent qui envoie la performative)
- *receiver* - identifie le destinataire du message (nom de l'agent qui reçoit la performative)
- *in-reply-to* - référence à un message auquel l'agent est en train de répondre (dans une réponse c'est le symbole précisé par l'attribut *reply-with* de l'émetteur)
- *force* - l'émetteur ne contredira jamais le contenu du message

Les messages de KQML ne concernent pas seulement des phrases dans un langage quelconque, mais sont enrichis d'une attitude sur le contenu (affirmation, désengagement, requête, question, etc.). Ce langage propose une description d'un nombre important de performatives (32 performatives) permettant les conversations entre agents mais manque de spécifications et de formalisation (cf. les critiques de [Cohen et Levesque, 1995]).

Les facilitateurs représentent une classe spécifique d'agents qui distribuent des meta-informations sur les autres agents et offrent des services de communication tels que les suivants :

- retransmission et distribution des messages
- découverte des ressources
- routage basé sur le contenu du message

Les performatives concernant les facilitateurs sont : *advertise, broker, recruit, recommend, forward, broadcast*. Les facilitateurs peuvent être des agents intelligents ou simplement des "pages jaunes" [Boissier, 2001].

Figure 5.11- Schéma des messages échangés entre les agents A et C en utilisant les services du facilitateur B dans le cas de la performative *broker*

On peut supposer que la performative *advertise(ask(P))* a été exécutée par l'Agent C avant que l'Agent A n'ait lancé *broker(ask(P))*. Dans le cas de la performative *recommend* le facilitateur B retransmet vers l'agent A une éventuelle performative *advertise(ask(P))* reçue antérieurement de la part de l'Agent C.

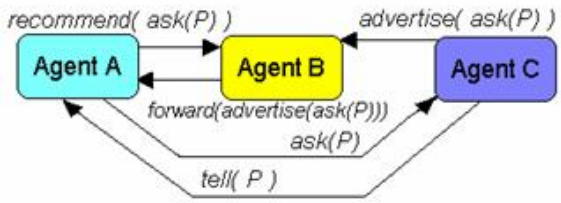


Figure 5.12- Exemple d'exécution d'une performative *recommend*
à l'aide du facilitateur B

On voit dans la figure 5.12 qu'après la réception du message *forward(advertise(ask(P)))* émis par le facilitateur B, l'Agent A ouvre un dialogue direct avec l'Agent C.

V.7.2. a) Langage de communication FIPA-ACL (Foundation for Intelligent Physical Agents)

L'organisation *Foundation for Intelligent Physical Agents* (FIPA) a été créée en 1996 et a permis de spécifier des standards dans la technologie agent et vise à favoriser l'interopérabilité des applications, des services et des équipements informatiques basés sur le paradigme agent. Ils ont défini un certain nombre de spécifications principales d'agents. Notamment, un standard de langage de communication agent ACL (*Agent Communication Language*) [FIPA, 1997] a été proposé et spécifié. Comme KQML, ce dernier est basé sur la théorie des actes de langages [Austin, 1962] : les messages sont des actes communicatifs, car ils sont prévus pour effectuer une certaine action en vertu de l'envoi. Les spécifications de FIPA-ACL se composent d'un ensemble de types de message et de la description. Les spécifications décrivent chaque acte communicatif avec une forme narrative et une sémantique formelle basées sur la logique modale [Levy, 2001]. En effet, FIPA considère que l'interaction est basée sur des actions communicatives mutuellement échangées entre les agents et qui sont initiées par des intentions, ce qui justifie l'utilisation d'une logique pour raisonner sur les intentions. Elles fournissent également la description normative d'un ensemble de protocoles d'interactions de haut niveau, y compris la demande d'action, l'établissement de contrat (contract net) et plusieurs mécanismes d'enchère.

FIPA-ACL est très semblable à KQML. Sa syntaxe est identique à celle de KQML exceptés différents noms pour quelques primitifs réservés. FIPA-ACL possède 21 actes communicatifs, exprimés par des performatives. En FIPA-ACL, il n'existe pas de primitives de gestion ni de facilitation. Les actes communicatifs peuvent être primitifs ou composés. Les actes communicatifs primitifs sont définis de façon atomique, c'est-à-dire qu'ils ne sont pas définis à partir d'autres actes. En revanche, les actes communicatifs composés sont définis à partir d'autres actes.

Un message comprend plusieurs éléments qui sont présentés ci-dessous.

- *Performative*- type de l'acte communicatif
- *Sender*- l'émetteur du message
- *Receiver* - le destinataire du message
- *reply-to* - participant à l'acte de communication
- *content* - le contenu du message (l'information transportée par la performative)
- *language*- le langage dans lequel le contenu est représenté
- *encoding*- décrit le mode d'encodage du contenu du message
- *ontology*- le nom de l'ontologie utilisée pour donner un sens aux termes utilisés dans le content
- *protocol*- contrôle la conversation
- *conversation-id* - identificateur de la conversation
- *reply-with* - identificateur unique du message, en vue d'une référence ultérieure
- *in-reply-to* - référence à un message auquel l'agent est en train de répondre (précisé par l'attribut reply-with de l'émetteur)
- *reply-by* - impose un délai pour la réponse

Les documents de référence de FIPA-ACL et KQML ne spécifient aucune implantation par l'utilisation d'un langage de programmation ni plate-forme particulière. Ce champ de développement est propre à chaque équipe désirant utiliser un tel langage. La seule obligation est que l'implémentation soit conforme aux spécifications du langage. Toutefois, il existe plusieurs implémentations, dont les principales sont JATLite, KAPI et COBALT proposées par l'équipe IRIT/SIEIRA de Toulouse.

V.8. LA COOPERATION DANS LES SYSTEMES MULTI-AGENTS

La coopération est la forme générale la plus étudiée dans les SMA. En effet la résolution distribuée d'un problème est le résultat d'interactions coopératives entre un ensemble d'agents. La coopération peut être vue comme la détermination de qui fait quoi, quand, où, avec quel moyen, de quelle manière et avec qui. L'objectif de la coopération est d'améliorer le mode de travail des agents en termes :

- D'augmenter le taux de l'achèvement de la tâche à travers le parallélisme.
- D'augmenter l'ensemble ou étendue des tâches réalisables en partageant des ressources
- D'augmenter la probabilité de compléter des tâches en entreprenant d'autres tâches.
- De diminuer l'intervention entre tâches en évitant des interactions malfaisantes.

V.8.1. Coopération et structure d'organisation

Il existe deux types d'organisation pour les sociétés d'agents :

V.8.1.1. Structure horizontale

Les agents de cette société sont au même niveau, il n'y a pas d'agent maître qui joue le rôle du superviseur et des agents esclaves qui jouent le rôle des agents exécutants qui traitent des sous tâches. Tous les agents peuvent communiquer entre eux (cf. figure 5.13 (a)).

V.8.1.2. Structure verticale

A l'inverse de la structure précédente les agents sont structurés par niveau. Dans un même niveau, on retrouve localement une structure horizontale (cf. figure 5.13(b)). Dans une telle structure, l'agent reçoit le problème à résoudre, il le décompose en :

- Des sous problèmes auxquels il peut répondre localement ;
- Des sous problèmes qu'il pourrait résoudre en coopérant avec les autres agents du même niveau que lui ;
- Des sous problèmes qu'il fait suivre aux agents du niveau inférieur dans la hiérarchie.

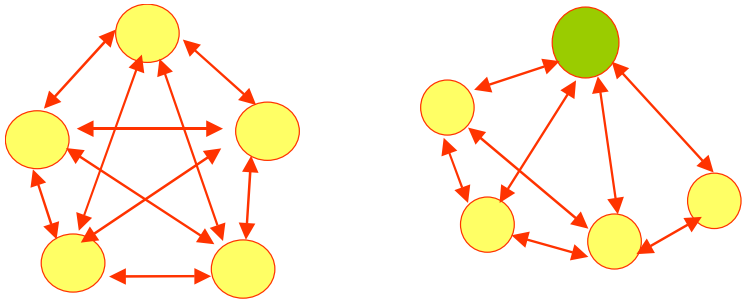


Figure 5.13 - (a) structure verticale (b) structure horizontale

V.9. MODES DE COOPERATION

Il y a deux manières de coopérer entre les agents. La première est due à l'existence d'une hiérarchie entre les agents, la seconde au contraire, suppose qu'il n'y a pas de hiérarchie entre les agents.

Cas ou il existe une hiérarchie entre les agents (cf. figure 5.13 (b))

Dans ce cas, il existe trois modèles de coopération :

V.9.1. Le mode « COMMAND »

Dans ce mode, un agent superviseur décompose un problème en sous problèmes qu'il repartit entre les agents. Ceux-ci les résolvent et renvoient les solutions possibles à l'agent superviseur.

V.9.2. Le mode « APPEL D'OFFRE »

Dans ce mode, un agent superviseur décompose un problème en sous problèmes, dont il diffuse la liste aux agents. Chaque agent qui le souhaite envoie une offre. L'agent superviseur choisi parmi celles-ci et distribue les sous problèmes. Le système fonctionne ensuite en mode commande.

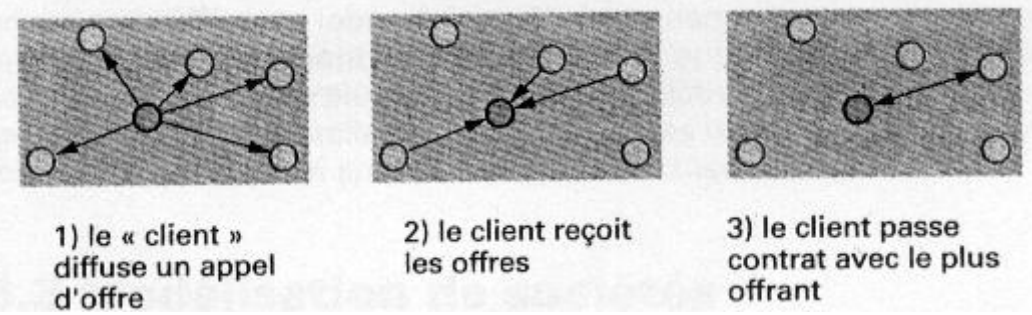


Figure 5.14 - Le mode appel d'offre

V.9.3. Le mode « COMPETITION

Dans ce mode un agent superviseur décompose un problème en sous problèmes, dont il diffuse la liste aux agents, comme dans le mode appel d'offre. Chaque agent résout un ou plusieurs sous problèmes et envoie les résultats correspondants à l'agent superviseur qui à son tour fait le tri.

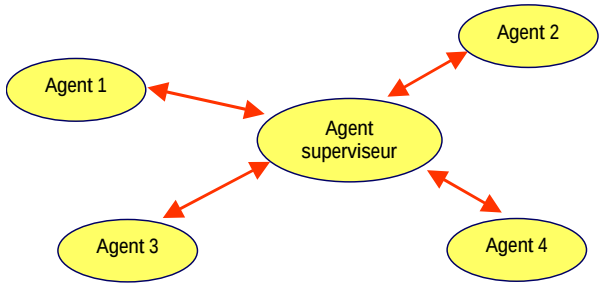


Figure 5.15- Coopération par commande, appel d'offres ou compétition

Cas ou il n'existe pas de hiérarchie entre les agents (cf. figure 5.13 (a))

V.9.4. Coopération par partage de tâches

Le problème est distribué entre les différents agents. Les agents travaillent indépendamment les uns des autres. Chaque agent dispose de ressources et de compétences nécessaires pour accomplir la tâche qui lui a été assignée. Le contrôle est dirigé par les buts, et les agents sont représentés par les tâches qu'ils se sont engagés à exécuter.

V.9.5. Coopération par partage des résultats

Les agents ne peuvent accomplir leurs tâches de manière indépendante. Ils sont appelés à se transmettre des résultats partiels. Le contrôle est dirigé par les buts, et les agents sont représentés par des sources de connaissances, la problématique réside dans la communication des résultats partiels.

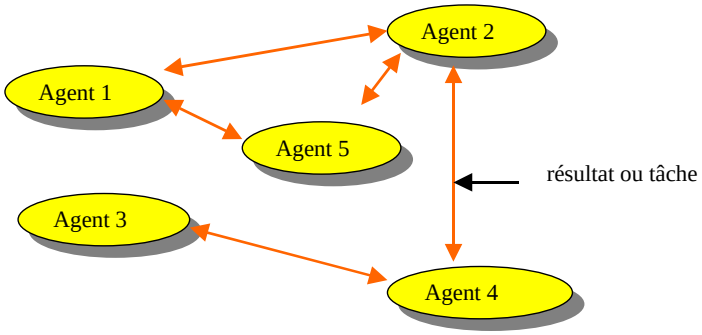


Figure 5.16- Coopération par partage de taches ou de résultats

V.10. RESOLUTION DES CONFLITS DANS LES SYSTEMES MULTI-AGENTS

Dans un système multi-agent, les agents interagissent en vue de réaliser des tâches ou d'atteindre des buts. L'interaction a lieu, d'habitude dans un environnement commun où les agents ont diverses zones d'influence (cf. figure 5.17), notamment diverses parties de l'environnement sur lesquelles ils peuvent agir. Ces zones peuvent être disjointes mais, dans la plupart des cas, elles se superposent - l'environnement est partagé par les agents. En interagissant dans un tel environnement partagé, les agents doivent coordonner leurs actions et avoir des mécanismes pour la résolution des conflits. La coordination et la résolution des conflits sont surtout nécessaires dans le cas des agents égocentrés (des agents ayant leurs propres buts, désirs, préférences, etc.) ou compétitifs mais aussi bien, parfois, dans le cas des agents coopératifs pour la communication des changements des plans ou l'allocation des tâches.

Le mécanisme favori pour la résolution des conflits et la coordination, inspiré du modèle des humains, est la négociation.

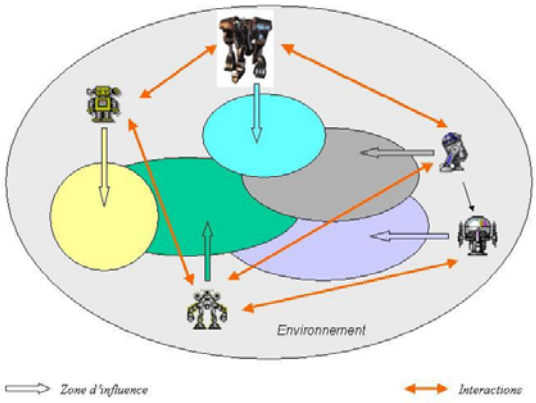


Figure 5.17- Interaction et zones d'influence dans un système multi-agent.

V.10.1.Coordination

Les agents travaillent sur des problèmes dont les solutions sont utiles pour les autres agents. Leur travail doit donc être temporairement coordonné. La coordination [Labidi et Lejouad, 1993] permet aux agents d'une part de considérer toutes les tâches (aucune tâche n'est ignorée). D'autre part la coordination évite aux agents de dupliquer le travail, c'est-à-dire de faire deux fois la même

tâche. La coordination des actions est liée à la planification et à la résolution des conflits car c'est à ce niveau qu'on tient compte des actions (plan) des différents agents.

Dans les SMA, la coordination des actions des agents peut s'organiser suivant deux schémas principaux : Soit une coordination au moyen d'un système capable de déterminer et de planifier (globalement) les actions des différents agents, ou bien à l'inverse, on décide de donner une totale autonomie aux agents qui, à leur tour, identifient les conflits pour les résoudre localement.

On peut distinguer deux types de coordination : la coordination due à la gêne (problème de navigation : les agents doivent coordonner leurs plans de navigations pour s'éviter mutuellement) et la coordination due à l'aide (la manutention : dans un environnement multi-robots, les agents doivent synchroniser leurs actions pour pouvoir agir efficacement et transporter un objet).

V.10.2.Négociation

Dans le cas des systèmes multi-agents intelligents, la négociation est une composante de base de l'interaction surtout parce que les agents sont autonomes [Jenning e.a. 2000] ; il n'y a pas de solution imposée à l'avance et les agents doivent arriver à trouver des solutions dynamiquement, pendant qu'ils résolvent les problèmes. On peut citer plusieurs définitions pour la négociation. David et Smith (1980) disent que: "*Par négociation, on entend une discussion dans laquelle des individus intéressés échangent des informations et arrivent à un accord en commun*." Pruitts (1981) donne une définition qui s'appuie sur des considérations psychologiques et pour laquelle le conflit est l'élément de base. "*La négociation est le processus par lequel plusieurs individus prennent une décision commune. Les participants expriment d'abord des demandes contradictoires, puis ils essaient de trouver un accord par concession ou par la recherche de nouvelles alternatives*."

Si on regarde les deux définitions citées, on peut identifier deux aspects essentiels de la négociation : la communication et la prise de décisions. Pour modéliser la négociation dans un logiciel multi-agent, il faut alors prendre en compte les aspects suivants :

- **Le langage de négociation** - le langage utilisé par les agents pour échanger des informations pendant la négociation ; le langage de négociation est composé d'un ensemble de primitives de communication.
- **Le protocole de négociation** - l'ensemble des règles qui régit la négociation (cf. paragraphe II.10 sur les modes de coopération) : les participants possibles dans la négociation, les propositions légales que les participants peuvent faire, les états de la négociation (par exemple l'état initial où commence la négociation, l'état où on accepte des soumissions ou la fin de la négociation) et une règle pour déterminer quand on est arrivé à un accord ou quand il faut s'arrêter parce qu'aucun accord n'a pu être trouvé.
- **L'objet de négociation** - un objet abstrait qui comprend les attributs qu'on veut négocier ; il peut s'agir de négocier un prix ou plus généralement plusieurs attributs comme le temps nécessaire pour satisfaire une commande, la qualité des produits, etc.
- **Le processus de décision** - le modèle que l'agent utilise pour prendre des décisions pendant la négociation. La partie la plus importante de la prise des décisions dans ce cas est la stratégie de négociation qui permet de déterminer quelle primitive de négociation l'agent doit choisir à un

certain moment. Le processus de décision revient à répondre à la question : "Que dois-je faire maintenant ?". Pour prendre une décision adéquate, un agent doit être capable de développer un raisonnement stratégique, notamment, de raisonner en tenant compte de ce que font/décident les autres agents et, s'il parvient à le savoir ou à le supposer, quel est le modèle de décision des autres agents.

Le protocole établit les règles de négociation. Par exemple, une négociation peut avoir lieu en un seul tour, comme l'enchère premier-prix offre-cachée, ou en plusieurs tours avec les participants faisant des offres (soumission) à chaque tour.

Le nombre de participants et les interactions possibles peuvent aussi varier :

- **Négociation un-à-un :** un agent négocie avec un autre, par exemple dans le cas où on essaie de négocier le prix d'achat d'une maison avec le représentant d'une agence immobilière
- **Négociation un-à-plusieurs :** un seul agent négocie avec plusieurs autres agents, par exemple les enchères où un agent veut vendre un objet
- **Négociation plusieurs-à-plusieurs :** plusieurs agents négocient avec plusieurs autres agents en même temps, par exemple, les participants à des enchères électroniques.

Il existe plusieurs méthodes de négociation pour l'allocation de tâches (le réseau contractuel).

Négociation centralisée

Le processus de négociation centralisée suppose que l'agent superviseur dispose d'une vue globale du problème. Il détecte les conflits entre les agents grâce à sa vue globale du problème, et il prend en charge la résolution de ces conflits entre les agents.

Négociation distribuée

Le processus de détection et de résolution de conflits est totalement distribué entre les agents. Les stratégies de résolution de conflits peuvent être communes ou locales aux agents.

Réseaux contractuels (Contract Nets)

C'est une technique d'allocation de tâches dédiée à la résolution distribuée de problèmes proposée par Davis et Smith [Davis et Smith, 1983]. Il s'agit d'un ensemble d'agents qui peuvent passer des contrats selon un protocole fixe qui a été imaginé à partir du protocole d'appel d'offres des contrats publics (cf. figure 5.18).

- L'agent ayant des tâches à accomplir (maître d'ouvrage), fait un appel d'offres (annonce de tâches). L'annonce est diffusée à l'ensemble des agents.
- Les autres agents évaluent ces tâches. Ceux pouvant en accomplir une font des propositions (offre) et deviennent alors des contractants potentiels.
- Le manager évalue ces offres et décide de passer un contrat avec l'un de ces agents.

Dans le protocole réseau contractuel, les agents peuvent prendre deux rôles : gestionnaire et contractant. L'agent qui doit exécuter une tâche (le maître d'ouvrage) commence par décomposer cette tâche en plusieurs sous-tâches. Le maître d'ouvrage annonce chaque sous-tâche sur un réseau d'agents (les contractants). Les agents qui reçoivent une annonce de tâches à accomplir évaluent l'annonce. Les agents qui ont les ressources appropriées, l'expertise ou l'information requise pour accomplir la tâche, envoient au maître d'ouvrage des soumissions ("bids" en anglais) qui indiquent leurs capacités à réaliser la tâche. Le maître d'ouvrage rassemble toutes les propositions qu'il a reçues et alloue la tâche à l'agent qui a fait la meilleure proposition. Ensuite, le maître d'ouvrage et les contractants échangent les informations nécessaires durant l'accomplissement des tâches. Par exemple, le contractant annoncera au maître d'ouvrage quand l'exécution de la tâche sera terminée. Dans des cas exceptionnels, un maître d'ouvrage peut annuler le contrat : annoncer au contractant qu'il faut abandonner l'exécution de la tâche.

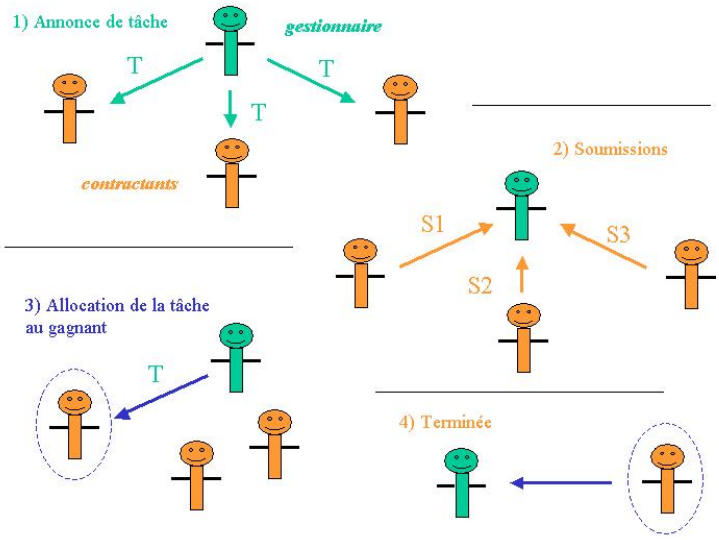


Figure 5.18- Etapes du protocole réseau contractuel

Les rôles des agents ne sont pas spécifiés à l'avance. N'importe quel agent peut agir comme un maître d'ouvrage et lancer un appel d'offres ; n'importe quel autre agent peut être un contractant pour une annonce qu'on a faite. Cette flexibilité permet de nouvelles décompositions des tâches : un contractant pour une tâche spécifique peut agir comme un gestionnaire en décomposant sa tâche et en annonçant les sous-tâches à d'autres agents. Les liens entre les gestionnaires et les contractants pour diverses tâches et sous-tâches forment un réseau contractuel hiérarchique qui permet la division des tâches et la synthèse des résultats. Le langage de communication FIPA (présenté dans la section 1.4) fournit, en plus d'une série de primitives de communication et d'une sémantique associée, une série de protocoles d'interaction.

Le protocole réseau contractuel a été utilisé pour la résolution coopérative des problèmes, c'est-à-dire la résolution des problèmes par des agents coopératifs.

Négociation aux enchères

Les enchères étaient assez restreintes auparavant, car elles s'organisaient à des moments bien définis, et avec un nombre limité de participants. Les enchères (en anglais "auctions") sont des mécanismes d'interaction assez simples ; pourtant il y a beaucoup de problèmes à étudier, concernant principalement le choix du protocole et de la stratégie à utiliser. Une enchère comprend, d'habitude, un initiateur (en anglais "actioneer") qui lance un appel d'offres, et plusieurs participants (en anglais "bidders") qui sont intéressés par contractualiser certaines tâches.

V.11. PROGRAMMATION ORIENTEE AGENTS

L'architecture de l'agent une fois décidée, on peut utiliser n'importe quel langage de programmation pour implémenter l'agent. Il est toutefois intéressant de disposer d'un langage de programmation qui offre des modèles de conception spécialisés pour bâtir nos agents et qui permette, de cette manière, de développer plus vite nos applications. Les langages de programmation agents ont été conçus dans ce but. Pourtant il est bien difficile de concevoir un langage qui offre les modèles adéquats pour implémenter n'importe quel agent. On a vu dans les sections précédentes qu'il y a une grande diversité d'architectures possibles pour les agents et que les architectures et les modes de coopération peuvent être très différents. Les chercheurs ont proposé de nombreux langages pour la programmation des agents, chaque langage favorisant une vision particulière de la notion d'agent intelligent. On ne peut pas dire qu'il y ait actuellement un langage générique. Une place particulière dans le panorama est occupée par le langage de programmation qui a introduit l'approche de la programmation orientée agents.

La programmation orientée agents a été proposée par Yoav Shoham en 1993 comme un nouveau paradigme de programmation, que l'on peut voir comme une spécialisation de la programmation orientée objets. Dans cette approche, les agents sont les éléments centraux du langage comme les objets sont centraux pour les langages orientés objets. La perspective sur les agents est cognitive : les agents sont caractérisés par des notions mentales comme leurs croyances, leurs décisions et leurs obligations. De plus, à chaque agent est associé un ensemble d'habiletés qui représentent ce que l'agent sait faire. En même temps, la programmation orientée agents suppose qu'on va développer des programmes dans lesquels plusieurs agents interagissent, ce qui met l'accent sur la dimension sociale des agents. Le langage de programmation proposé par Shoham comme démonstration de ce nouveau paradigme s'appelle AGENT0.

La différence principale entre un tel langage et un langage de programmation classique que l'on pourrait utiliser pour développer des agents, vient du fait que les notions mentales qui caractérisent les agents apparaissent dans le langage lui-même, et que la sémantique du langage est intimement liée à la sémantique de ces notions mentales. La programmation orientée agents peut être vue comme une spécialisation de celle orientée objets parce que les modules du programme sont maintenant des agents, c'est-à-dire des objets avec un état qui définit les notions mentales associées, et que les messages entre objets sont remplacés par des messages entre agents. En quoi les messages entre agents différent-ils de ceux entre objets ? Premièrement, parce que ces messages

sont modélisés en partant de la théorie des actes de langage, qui s'intéresse aux actions de communication comme informer, demander, offrir, accepter, rejeter et d'autres (pour la théorie des actes de langage). Deuxièmement, puisque les agents sont autonomes et dotés de capacités mentales, ils ont la liberté de décider s'ils vont ou non exécuter l'action spécifiée dans le message. Par contraste, un objet recevant un message va toujours exécuter l'action spécifiée dans le message. Cette spécialisation même fait que la programmation orientée agents est différente de celle orientée objets comme le montre le tableau ci-dessous, tableau de différences établi par Shoham.

	POO	**POA**
Unité de base	objet	agent
Paramètres définissant l'état de l'unité de base	pas de contraintes	croyances, décisions, obligations, habiletés
Processus de calcul	envoi de messages et méthodes pour la réponse	envoi de messages et méthodes pour la réponse
Types de messages	pas de contraintes	informer, demander, offrir, promettre, accepter, rejeter, …
Contraintes sur les méthodes	pas de contraintes	consistance, vérité, …

Tableau 5.2 - Programmation Orientée Objets versus Programmation Orientée Agents

Avec l'évolution des théories concernant les agents et des langages de programmation associés, on peut également souligner d'autres différences entre POO et POA :

- Les agents sont autonomes alors que les objets ne le sont pas ; un agent va décider par son propre processus de décision s'il exécute ou non une action requise ;
- Les agents ont leurs propres buts et ils agissent d'une manière pro-active pour atteindre leurs buts (par exemple, ils saisissent des opportunités) alors que les objets ne le font pas ;
- Les agents sont capables d'un comportement social : ils peuvent s'engager dans des interactions complexes, par exemple coopération, compétition, négociation, avec d'autres agents ; ce n'est pas le cas des objets ;

La figure 5.19 illustre l'évolution de la programmation structurée vers la programmation orienté agent en passant par les différentes phases d'évolution (programmation orienté objet, les objets distribués en utilisant des standards comme CORBA, programmation composants qui a fait son apparition avec le développement en langage java).

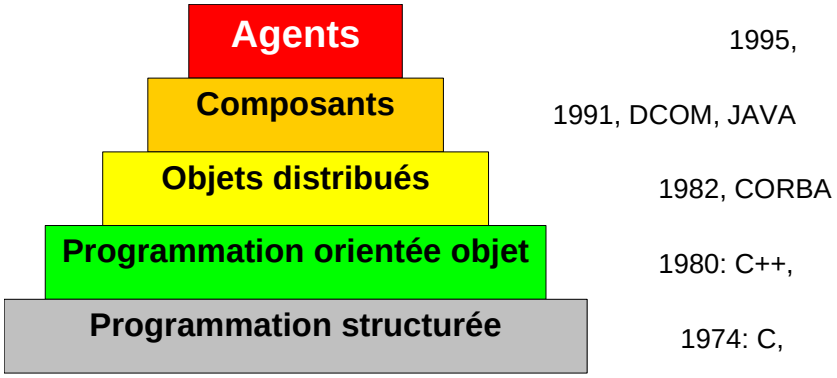


Figure 5.19 - L'évolution vers la programmation orientée agent

V.12. CONCLUSION

Les technologies agent attirent beaucoup l'attention des industriels de nos jours, particulièrement pour des applications largement distribuées et ouvertes. Ces applications présupposent l'interopérabilité des agents issus de différentes infrastructures. Les concepts multi-agents peuvent désormais apporter des solutions à des problèmes industriels.

Ce chapitre a porté sur le principe des SMA et des interactions dans la résolution collective de problèmes. Les concepts de base permettant de mettre en place l'échange d'information, et d'interaction ont été étudiés. La communication constitue donc un concept fondamental du paradigme multi-agent et la brique de base de cette notion d'interaction. Les ACLs permettent la communication de haut niveau calquée sur la théorie des actes de langage. Les agents communiquent des manières bien plus sophistiquées que les objets. Contrairement aux objets, les agents n'ont pas nécessairement de comportement déterministe (ils peuvent dire « non »).

La notion d'interactions, qui est fortement liée à la notion de construction de stratégies de résolution coopérative des problèmes, offre un cadre de développement et d'analyse très intéressant car ce sont les possibilités d'échanges et d'influences que les interactions créent entre les agents. Les interactions dans un système d'agents cognitifs se traduisent donc par des schémas de conversations construits à partir des actes de communication et des prises de décision locales des agents (issus du caractère autonome et du comportement interne des agents) ou sous la forme de protocoles d'interaction pré-définis qui concernent le potentiel de tous les agents du système.

Si les systèmes multi-agents semblent être les outils les mieux adaptés à la complexité des problèmes qui se posent pour le diagnostic des systèmes physiques, il reste à concevoir les algorithmes de chaque agent et le mode de coordination entre les agents. En effet, la modélisation d'une procédure de diagnostic distribuée est une des applications possibles des systèmes multi-agents. Une architecture logiciel orientée agent doit être aussi naturelle que possible c'est-à-dire, que chaque agent doit être identifié clairement par rapport à sa sémantique, son rôle et sa responsabilité. Les études basées sur les systèmes multi-agents se proposent d'analyser le problème en terme d'interactions entre les agents.

Utiliser le paradigme multi-agent possède deux objectifs :

- Développer une architecture robuste d'une procédure de diagnostic permettant le partitionnement et la séparation des entités favorisant la réutilisation des sources, et la maintenance des logiciels.

- Spécifier un système de diagnostic distribué, favorisant la coopération et la communication entre les différents agents.

Toutefois, le lecteur se référera au chapitre 6 dans lequel on décrit un système de diagnostic dédié aux systèmes complexes intitulé MAGIC. Nous allons détailler l'utilisation des agents dans la construction d'un système de diagnostic. Ce travail est intéressant car il fait apparaître comment les agents servent à la fois à modéliser l'organisation et à spécifier de manière simple et « naturelle » un système distribué dérivé de cette modélisation.

Chapitre 6

Un système de diagnostic multi-agent : MAGIC

VI.1. INTRODUCTION

Ce chapitre présente une approche de diagnostic distribué fondée sur une architecture multi-agent et dédiée aux systèmes complexes, produit d'un projet européen intitulé MAGIC (*Multi-Agent-based Diagnostic Data Acquisition and Management in Complex Systems*). Le système MAGIC intègre diverses méthodes de diagnostic, adaptées à différentes situations, et tous les outils nécessaires pour les adapter aux besoins du client, les configurer et les employer pour la surveillance en temps réel. Le but de MAGIC est de fournir à l'opérateur une information claire, aussi détaillée que nécessaire et facilement compréhensible sur l'état courant d'un système à surveiller. En plus de ces informations, le système est capable de suggérer des actions correctives appropriées.

MAGIC est un système d'aide à la décision pour la supervision de procédés complexes basé sur une architecture multi-agent. La procédure de diagnostic a été distribuée à différents niveaux et dans différents agents. Cette architecture a permis d'utiliser simultanément diverses techniques de détection de défauts et d'aide à l'opérateur [Garcia-Beltran, 2004]. En effet, les méthodes de détection utilisées sont aussi diverses que des techniques à base de traitement du signal (méthodes statistiques), des techniques à base de connaissances (graphes causaux), des réseaux de neurones, des observateurs d'état.

La deuxième partie de ce chapitre est consacrée à la description précise de la manière dont l'agent responsable de l'analyse diagnostique, nommé Agent de Décision Diagnostique (DD-Agent), est construit. Son rôle est de prendre une décision finale sur le ou les états possibles du système à diagnostiquer en s'appuyant sur les résultats des algorithmes de détection. La difficulté principale à laquelle l'Agent de Décision Diagnostique doit faire face est que les symptômes proviennent de différents sous systèmes testables qui ne sont pas toujours indépendants les uns des autres. Un ou plusieurs composants peuvent être communs à deux sous-systèmes différents. Un sous-système peut inclure un autre sous système partiellement. Des symptômes peuvent être obtenus grâce à des algorithmes de détection différents appliqués à un même sous-système. Or, tous les symptômes doivent être analysés ensemble afin d'obtenir des diagnostics fiables.

Les méthodes développées et les outils implémentés ont été testés sur un exemple industriel, l'Enrouleur Hydraulique, proposé par un des partenaires industriels du projet MAGIC. Les méthodes d'analyse diagnostique logique et à base de tables de signatures ont été utilisées pour la localisation de défauts. Les algorithmes ont été tout d'abord évalués avec des scénarios de défauts provenant d'un simulateur de l'enrouleur hydraulique. Ensuite, des données issues d'un véritable Enrouleur Hydraulique ont été utilisées comme scénario de test.

VI.2. L'ARCHITECTURE MAGIC

L'extrême variété des méthodes de détection utilisées montre que, malheureusement, aucune d'entre elles ne peut prétendre être meilleure que les autres. Elles sont plutôt dédiées à différents types de problèmes et se basent sur différentes connaissances du système à diagnostiquer. Il est

donc naturel que pour diagnostiquer un système complexe, différentes méthodes soient utilisées. Une amélioration significative des applications industrielles peut avoir lieu seulement quand les problèmes du FDI sont résolus dans une structure qui utilise et qui intègre différentes technologies FDI. Chaque algorithme de détection est encapsulé dans un agent de diagnostic appelé (Diagnostic-Agent).

Utiliser des méthodes de détection différentes ne suffit pas à l'obtention d'un diagnostic correct pour l'ensemble du système physique. Étudier la collaboration possible entre plusieurs méthodes est l'un des défis scientifiques de MAGIC. Les différents Agents de détection sont nommés Diagnostic Agents ou D-Agents. L'Agent de Décision Diagnostique *(Diagnostic Decision Agent*) doit prendre une décision définitive et globale au sujet de l'état plausible du système dans sa globalité. L'Agent de Décision Diagnostique est donc consacré à l'isolation du défaut. La principale difficulté pour cette tâche est que les différents symptômes des Diagnostics Agents peuvent concerner soit différents sous-systèmes qui ne sont pas indépendants mais physiquement liés entre eux, soit le même sous-système, mais avec des résultats de test de détection différents. Prenons des exemples. Les capteurs peuvent être communs à 2 sous-systèmes différents. Un sous-système peut inclure un autre sous système physique. Les symptômes peuvent être obtenus grâce à des algorithmes de détection différents tels que l'approche à base d'observateur ou l'approche à base de traitement de signal, appliquées au même sous-système. Chaque symptôme est obtenu par l'application d'une méthode particulière à un sous-système particulier dans des circonstances particulières. Cette information doit aider à évaluer leur pertinence au diagnostic global.

Une première remarque importante est qu'un symptôme ne peut pas être analysé sans savoir les conditions dans lesquelles il a été obtenu. En effet, avec les méthodes à base de modèle, les symptômes comptent sur la qualité du modèle, et évidemment, le domaine de validité du modèle doit être prise en considération.

Un autre point important est comment relier des symptômes aux composants. Une deuxième remarque est que quelques modèles peuvent représenter le comportement normal d'un composant ou différents types de comportements défectueux. Donc, un modèle de composant comporte une hypothèse sur l'état du composant. Ces hypothèses sont essentielles pour le raisonnement diagnostique parce qu'elles fournissent une façon d'interpréter les symptômes. Un symptôme révèle une inconsistance entre les observations et le modèle de référence. Donc si un modèle du comportement normal est utilisé, il peut être déduit que l'état du composant est contradictoire avec l'hypothèse du comportement normal et donc le composant est défectueux. En revanche, si un modèle d'un mauvais fonctionnement est utilisé alors, il peut être déduit que l'état du composant est contradictoire avec le défaut pris en compte par le modèle.

Une première conséquence de cette analyse est que les résultats fournis par les agents de détection (D-Agents) ne se limitent pas seulement à des symptômes. Nous définissons dans ce qui suit l'information nécessaire à transmettre par les agents de diagnostic à l'agent de décision diagnostique (DDA) : les symptômes, leur niveau de confiance, information sur la validité du symptôme, des informations sur le support des tests de détection.

L'architecture distribuée de MAGIC est fondée sur une architecture en couches, Figure 6.1. Ces fonctionnalités sont distribuées sur 5 couches [Köppen-Seliger, Marcu et al. 2003] :

- Niveau 1 : spécification du procédé
- Niveau 2 : acquisition de l'information
- Niveau 3 : détection
- Niveau 4 : analyse diagnostique
- Niveau 5 : aide à l'opérateur
- Niveau de gestion du système

En référence au concept Multi-agent Multi-Couche (MAML), illustré à la figure 6.1, les différents agents dans MAGIC auraient les tâches suivantes :

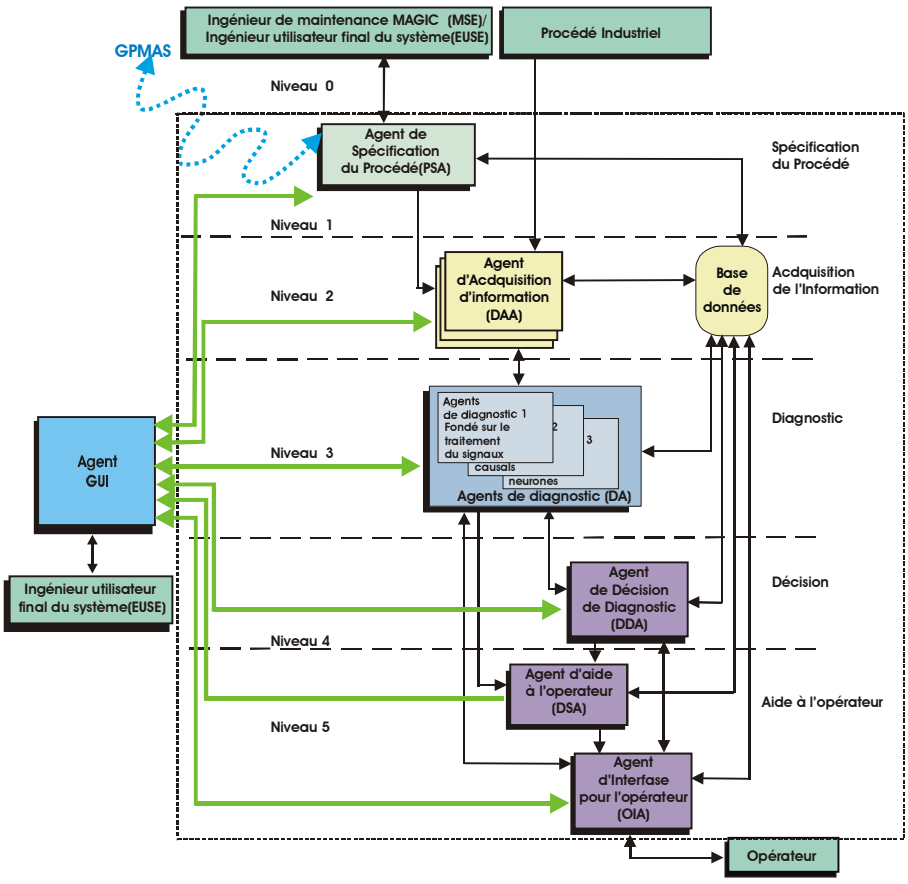


Figure 6.1-Architecture générale de MAGIC : Architecture Multi-agent-Multi-Layer (MAML)

VI.2.1.1. Niveau 1 - Spécification du procédé

- **PSA -** l'Agent de Spécification de Procédé (PSA) a pour tâches d'adapter le système de surveillance au système à surveiller et à diagnostiquer et de configurer les divers agents du système MAGIC.

- **GPMAS -** Rigoureusement GPMAS (Serveur des Algorithmes Mathématiques d'objectif Général) n'est pas un agent dans la structure MAGIC car cet élément n'utilise aucune méthode

de communication inter-agent. Le but principal du GPMAS est de fournir les outils d'aide à la modélisation des comportements des composants et des unités physiques à surveiller.

VI.2.1.2. Niveau 2 - Acquisition de l'information

- **DAA (Diagnostic Acquisition Agent) -** Dans le cadre de MAGIC, chaque agent d'acquisition de données (DAA) est l'interface entre le système MAGIC et l'application industrielle. Il est responsable de la gestion des données du processus récupérées à partir des capteurs et des actionneurs (par l'intermédiaire du système (SCADA). Il est aussi responsable de réaliser les pré-traitements des données (filtrage, mise en format, validation, etc).

- **Base de Données -** La base de données du système MAGIC est utilisée pour stocker et rechercher toute information nécessaire pour configurer correctement le système MAGIC. Cette base de données permet aussi de stocker toute information générée pendant le fonctionnement des agents du système MAGIC. La base de données stocke des signaux venant du procédé par le biais de l'agent d'acquisition de données (DAA), toute l'information de configuration pour chacun des agents de MAGIC, les messages de communication inter-agent, la description structurelle du procédé et les messages générés et reçus pour l'opérateur.

VI.2.1.3. Niveau 3 - Diagnostic

- **D-Agent (Diagnostic Agent) -** Les agents de détection (D-Agent) participent au processus de diagnostic global en effectuant des "tests de détection" et en produisant des symptômes qui seront examinés plus tard par l'agent de prise de décision diagnostique (Diagnostic Decision Agent). Plusieurs agents de détection peuvent fonctionner en parallèle. Ils sont déclenchés par le DDA suivant une stratégie de diagnostic définie à priori par l'ingénieur système.

Divers algorithmes de détection ont été choisis pour leur intégration dans le système MAGIC, cette sélection s'est appuyée sur une étude comparative de différentes méthodes existantes [Atlas, Lacatusu et al., 2003]. Néanmoins, d'autres algorithmes peuvent aussi être intégrés si les structures d'information de MAGIC sont respectées. Actuellement les algorithmes mis en oeuvre sont les suivants :

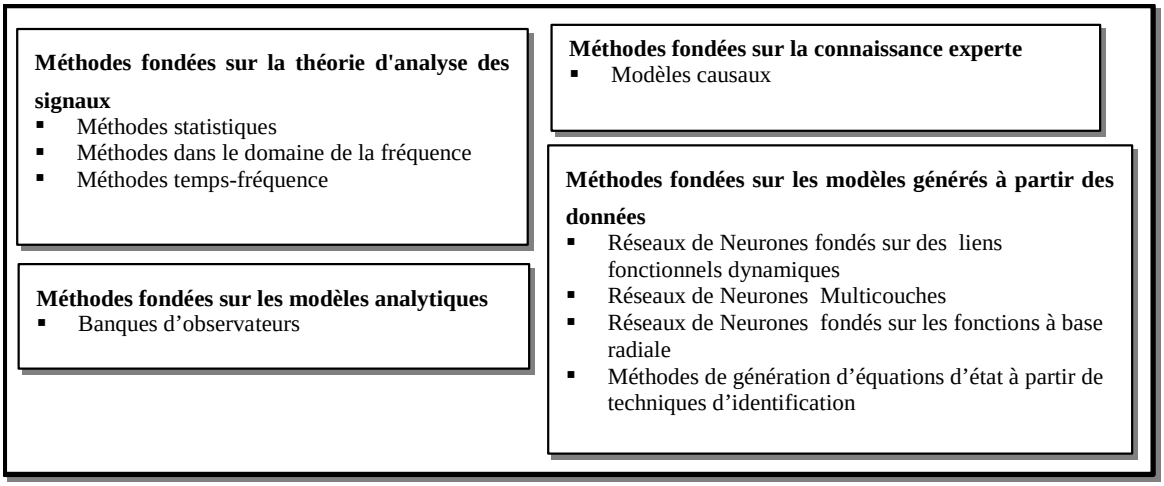


Figure 6.2 – Les différentes méthodes de diagnostic

VI.2.1.4. Niveau 4 – La prise de décision

- **DDA (Diagnostic Decision Agent)** - C'est le niveau de l'analyse diagnostique qui conduit à une décision globale tenant compte des résultats obtenus par les tests de détection.

VI.2.1.5. Niveau 5 - Aide à l'opérateur

- **DSA (Diagnostic Support Agent) -** La responsabilité de l'agent d'aide à la décision (DSA) est de fournir des conseils à l'opérateur afin de lui apporter une assistance dans les tâches de maintenance, où d'action de reprise de contrôle en cas de défaut.

- **OIA (Operator Interface Agent) -** Le système MAGIC est équipé d'une interface Homme-Machine implantée dans l'agent d'interface à l'opérateur (OIA). Cette interface permet de visualiser les diagnostics fournis par l'agent DDA ainsi que les différents symptômes et messages collectés sur le système. Il permet aussi d'interagir sur le système MAGIC en permettant de désactiver des tests de détection.

VI.2.1.6. Niveau de gestion du système

- **GUIA (Graphic User Interface Agent)-** Hors de la hiérarchie fonctionnelle de MAGIC, l'agent GUI (Graphic User Interface) est en charge de la gestion du système au niveau informatique. L'agent GUI a la capacité de déclencher, d'arrêter et d'interroger un agent de son état actuel. Le GUIA est principalement utilisé durant le démarrage du système MAGIC.

Cette décomposition est originale pour les deux communautés Automatique et d'Intelligence Artificielle. Dans la communauté Automatique, la plupart des algorithmes combinent ces rôles. Par exemple, la détection et la localisation (les tests de consistance et l'analyse des symptômes) sont exécutées en même temps quand on utilise un banc d'observateurs d'état [Franck, 1990]. En revanche, dans la communauté IA, le raisonnement logique utilisé pour l'analyse des symptômes sépare la partie génération de conflit et la partie analyse des conflits pour déduire les diagnostics. Il y a plusieurs façons d'exécuter un test de détection et il y a aussi plusieurs façons de programmer les tests de détections. Chaque communauté utilise sa propre méthode pour analyser des symptômes : le raisonnement logique développé par la communauté IA peut donner plusieurs diagnostics plausibles, en se basant sur une hypothèse d'exonération [Dague, 2001] et d'autres fois on ne tient pas compte de cette hypothèse d'exonération [Leyval, 1994], le raisonnement FDI a été développé par la communauté automatique.

VI.3. LES PHASES DE FONCTIONNEMENT DE MAGIC

Pendant les étapes de configuration, seules les fonctionnalités implémentées dans les niveaux 1 et 2 sont utilisées, tandis que dans l'étape de fonctionnement normal les fonctionnalités implémentées dans les niveaux 2, 3, 4, 5 et de gestion sont utilisées.

VI.3.1. Les modes de configuration

Les phases de configuration permettent de sélectionner les fonctionnalités adaptées à l'application industrielle.

- *Les phases 1 et 2* couvrent l'identification initiale de la structure du procédé industriel, et l'adoption d'une stratégie de diagnostic et de support à l'opérateur.
- *La phase 3* couvre tous l'identification des modèles, des paramètres et de la configuration des valeurs opérationnelles.

Grâce à la modularité de MAGIC, l'utilisateur final peut choisir entre les divers types d'agents proposés, ceux qui s'adaptent au mieux aux demandes du procédé [Atlas, Lacatusu et al, 2003]. En général, les tâches de configuration de la phase 1 peuvent être classées en trois rubriques :

1. *Caractérisation du procédé*

La description structurelle de l'application industrielle : cette tâche a pour objectif de décrire la structure du procédé industriel sans une forme hiérarchique. Un procédé industriel est composé d'unités physiques et chaque unité physique est constituée de composants physiques. Un composant physique est conçu comme une partie indivisible de l'application industrielle.

La description des variables physiques dans l'application industrielle : l'implémentation d'un système de diagnostic automatique a besoin d'avoir des mesures des diverses variables du procédé pour la génération de symptômes. Par conséquent, la caractérisation des variables physiques permet d'établir quelles sont les variables physiques disponibles, sur quels composants, avec quelles caractéristiques (bruit, précision de capteurs, temps d'échantillonnage etc.).

La description des états physiques : les différents états modélisés des composants y sont décrits. L'état le plus fréquent est celui qui correspond à l'état normal des composants, mais il est aussi possible de décrire le mauvais fonctionnement associé à des défauts spécifiques. Un état est généralement lié à un ou plusieurs modèles de comportement. Les domaines de validité des modèles sont décrits durant cette étape.

2. *Sélectionner la stratégie globale de diagnostic et la méthode de diagnostic à utiliser.*

3. *Mettre en œuvre des tests de détection associés aux agents de détection.*

➥ La génération de modèles de référence : cette tâche a pour objectif de générer des modèles de référence et/ou d'établir des gabarits et des seuils de décision pour les algorithmes de détection. Cette fonction est accomplie en collaboration avec l'agent de configuration (PSA), le serveur des algorithmes mathématiques (GPMAS) (outil d'aide à la modélisation des comportements des composants et des unités physiques à surveiller) et les divers agents d'acquisition de données (DAA) (voir figure 6.3). L'agent de configuration (PSA) fait office de "client" et les deux autres agents de "serveur". D'un coté, l'agent de configuration (PSA) demande aux agents d'acquisition (DAA), les signaux nécessaires pour la construction des modèles et de l'autre, l'agent de configuration (PSA) demande au GPMAS la construction des modèles.

Le serveur des algorithmes mathématiques (GPMAS) supporte les activités d'identification des modèles, leur validation et leur mise à l'essai. Finalement, les modèles sont récupérés par l'agent de configuration (PSA) qui les stocke dans la base de données.

La capture de l'information pour la configuration de l'agent d'analyse diagnostique (DDA), l'agent d'aide à la décision (DSA) et l'agent interface à l'opérateur (OIA) : Cette tâche a comme objectif la capture de diverses informations nécessaires l'opération continue de MAGIC. Les exemples de cette information sont les règles d'activation des tests de détection, les conseils de maintenance associée aux composants, etc.

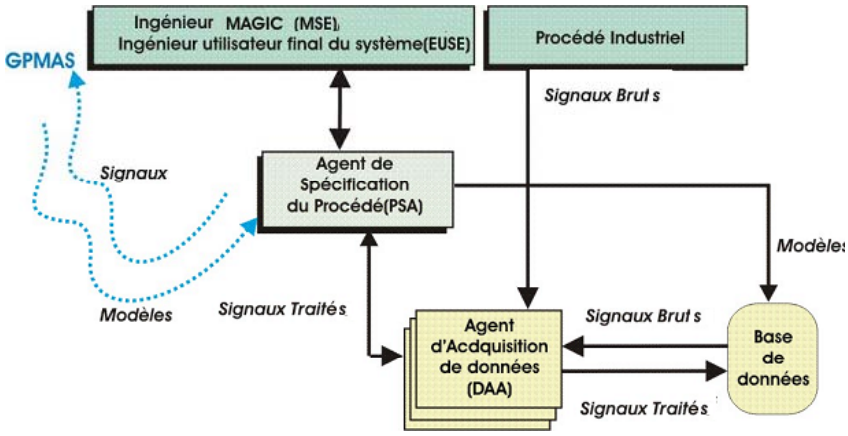


Figure 6.3- Agents actifs durant l'étape de configuration

VI.3.2. Le mode d'opération continue

VI.3.2.1. Acquisition de données

Le processus de diagnostic commence avec l'acquisition de données, les différents DAA récupèrent les données brutes via le système SCADA et réalisent le prétraitement prévu (filtrage, mise en format, validation, etc.). Les données prétraitées sont stockées dans la base de données et les autres agents sont informés de leur disponibilité.

VI.3.2.2. Diagnostic

Génération de symptômes : après l'acquisition des données à partir du procédé, les symptômes doivent être extraits à partir du système physique à diagnostiquer. Ici, la consistance entre le comportement réel du système physique et les modèles de référence doit être vérifiée par les

différents tests de détection par exemple : causal [Garcia-Beltran, Lesecq et al, 2003], neuronal [Marcu et al, 2003], signal [Lesecq et al, 2003].

Raisonnement diagnostique : l'agent de décision diagnostique (DDA) analyse les symptômes fournis par les agents diagnostics (D-Agents) courants pour déterminer l'état global du procédé. Si les symptômes concernent le même sous-système, cet agent de décision diagnostique doit résoudre les conflits possibles. De plus, il agrège les symptômes dans une décision globale pour décrire l'état des sous-systèmes. Si les symptômes concernent différents sous-systèmes reliés, cet agent doit employer des informations sur la structure du procédé afin de remonter aux composants incriminés. Deux méthodes différentes de raisonnement pour l'analyse de défauts sont employées afin de résoudre des résultats contradictoires possibles venant de différents D-Agents; la première méthode localisation est basée sur les tables de signatures et la deuxième méthode est basée sur le raisonnement logique [Ploix et al, 2003].

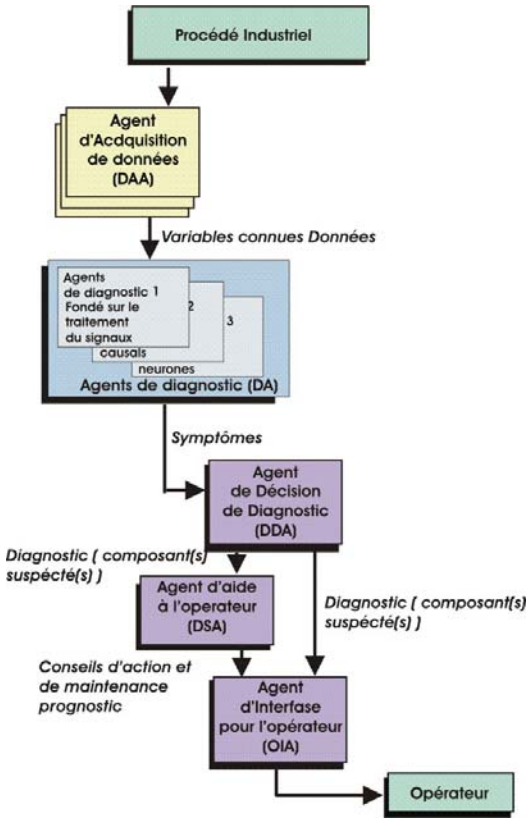


Figure 6.4- Les agents actifs en mode d'opération continue

VI.3.2.3. Support à l'opérateur

Génération de conseils d'action et de maintenance à l'opérateur : à partir des résultats de diagnostic générés par l'agent de décision diagnostique (DDA), l'agent d'aide à la décision (DSA) essaye de faire la prédiction du comportement du système, et de générer les conseils d'action et de maintenance associées aux composantes suspectées [Garcia-Beltran, 2004].

Interface opérateur : par l'intermédiaire de l'interface graphique de l'OIA, l'opérateur est capable d'accéder aux diagnostics, aux messages et aux conseils générés par les différents agents. L'opérateur reçoit l'information sur les différents symptômes et variables, la liste des composantes suspects (diagnostics), le comportement futur possible du procédé.

VI.4. LES MECANISMES DE COMMUNICATION DANS MAGIC

Les différents agents de MAGIC accomplissent des tâches très spécialisées qui constituent seulement une petite partie du processus global d'une procédure de diagnostic. Afin d'atteindre le but global du système MAGIC qui est le diagnostic et l'aide à l'opérateur, les agents doivent échanger des informations. En raison de sa standardisation, CORBA (*Common Object Request Broker*) a été sélectionné comme *middleware*[5] de base pour toutes les tâches de communication (Figure 6.5). Ainsi, CORBA établit un raccord fiable entre les objets de type « serveur » et les objets de type « client » et permet de masquer l'hétérogénéité des systèmes matériels et logiciels sous-jacents, de rendre la répartition transparente et de fournir des services répartis d'usage courant.

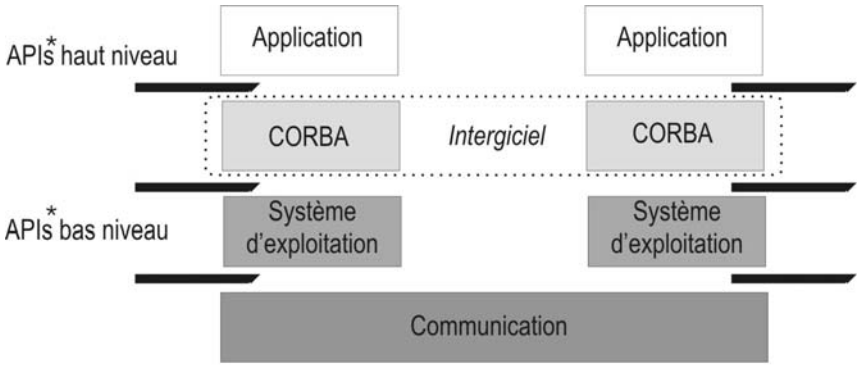


Figure 6.5– La position de l'intergiciel dans les applications répartis

La transparence offerte par CORBA permet de parler d'un niveau de communication plus élevé, placé dans une couche logiquement supérieure dans le modèle OSI à celle des protocoles de transfert de données (intergiciel + système d'exploitation + TCP/IP, HTTP, IIOP), et adressé au niveau intentionnel et social des agents.

Un agent MAGIC est capable de communiquer avec les autres agents de diverses manières [Albert, Langle, 2004] :

- en utilisant le langage de communication d'agent FIPA,
- en employant une notification d'événements,
- en employant la notification d'événements en combinaison avec une technique de transfert d'information via un fournisseur de données.

VI.4.1. Communication via le langage de communication d'agent FIPA

Les langages de communication d'agents appartiennent au niveau de communication social et intentionnel des agents, un d'entre eux, l'ACL (*Agent Communication Langage*) spécifié par la

[5] « Le *middleware* (intergiciel) est la couche logicielle située entre les couches basses (système d'exploitation, protocoles de communication) et les applications dans un système informatique réparti. Son but est de faciliter le développement de ces applications, en masquant l'hétérogénéité des systèmes sous-jacents et les détails de leurs mécanismes, et en fournissant des interfaces normalisées de haut niveau.» Krakowiak, S. (2002). Enseignement d'option en Master d'Informatique : Middleware adaptable. IMAG-LSR and INRIA.

FIPA (*the Foundation for Intelligent Physical Agents*, [FIPA, 2003] a été pris pratiquement comme standard dans les développements des systèmes multi-agents. Le but de l'ACL- FIPA est d'interagir entre les agents quel que soit le protocole qu'ils utilisent. Pour que les agents puissent utiliser un nouveau protocole, il suffit d'implémenter une nouvelle interface.

Dans l'ACL de FIPA le message minimum contient l'information suivante :

- le type du message envoyé.
- l'expéditeur du message
- le destinataire du message
- le contenu du message

Cependant, ces informations minimales ne suffisent pas toujours pour communiquer : il est parfois nécessaire, pour la compréhension du message et pour la rapidité de traitement du message, d'indiquer d'autres informations tels que le langage utilisé dans le contenu du message, le protocole, l'ontologie auquel le message se rattache, la référence d'un message antérieur auquel le message actuel se rattache. Ce langage offre une grande flexibilité de communication grâce a) au fait que le contenu du message peut être encodé dans des langages différents b) aux différents protocoles de transmission et c) la possibilité d'avoir différentes réponses potentielles de l'agent de réception. Ce type de communication est employé dans MAGIC dans tous les cas où un agent veut communiquer avec un autre agent. Cette communication sera employée pour toutes les interactions qui ne font pas partie du processus régulier de diagnostic, par exemple :

- L'agent interface à l'opérateur (OIA) fait une demande à l'agent de décision diagnostique (DDA) de désactivation d'un test de détection.
- L'agent interface à l'opérateur (OIA) fait une demande à l'agent d'aide à la décision (DSA) d'un pronostic d'une variable.

VI.4.2. Communication via notification d'événements et fournisseur de données

Dans les systèmes répartis à grande échelle, la notification d'événements est un système de communication très rapide. Contrairement aux autres modèles de communication, ce modèle permet à chaque agent émetteur de continuer sa tâche après les envois de notification d'événement, l'agent émetteur n'a pas besoin de savoir quels agents sont intéressés par la notification, au contraire, ce sont les agents récepteurs qui doivent s'abonner au service de notification d'événements. Néanmoins, puisque la notification est transmise seulement quand il y a un changement détecté par un agent et qu'il n'y a pas de réponse de la part des agents récepteurs, chaque message devient important. La communication fondée sur la notification d'événements présente le risque que l'information envoyée à un canal d'événement ne soit pas reçue par les agents concernés. Afin d'éviter cette perte d'information, tous les événements envoyés par un fournisseur d'événements contiennent un compteur. Le consommateur peut comparer le compteur des événements successifs et peut détecter si un événement n'a pas été reçu correctement. Cette approche ne garantit pas que tous les événements soient fournis correctement, mais elle permet d'identifier l'absence d'un

événement précédent. Si un consommateur détecte qu'un événement est absent, il doit informer le système MAGIC à ce sujet en envoyant un événement d'avertissement.

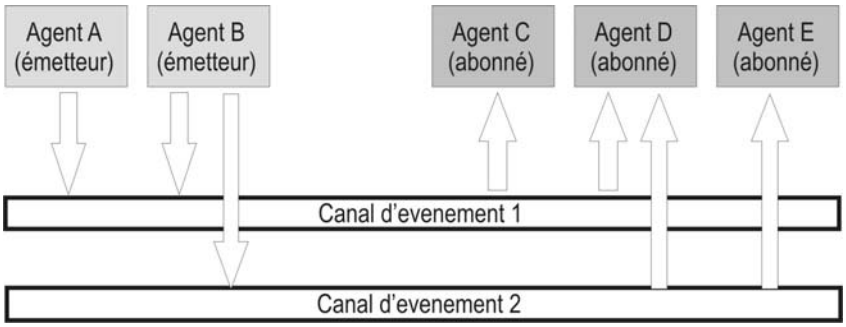


Figure 6.6–Abonnement des agents aux canaux d'événement.

La notification d'événement est employée de deux façons différentes : soit pour distribuer l'information liée aux tâches administratives du système soit pour informer les autres agents que des nouvelles données sont disponibles dans le Fournisseur de Données de l'agent qui a envoyé l'événement. Les paragraphes suivants seront dédiés au dernier cas.

Le modèle de communication appelé Fournisseur de Données fournit une communication synchrone via CORBA entre les différents agents qui coopèrent. Les données sont stockées dans une mémoire tampon qui agit en tant que serveur de données. Ensuite, l'agent notifie un événement à tous les autres agents dans MAGIC. Cet événement contient les informations sur le type des données qui sont disponibles et le lien à l'agent serveur qui peut fournir les données. Un agent intéressé par cette information peut accéder aux données en employant une requête CORBA.

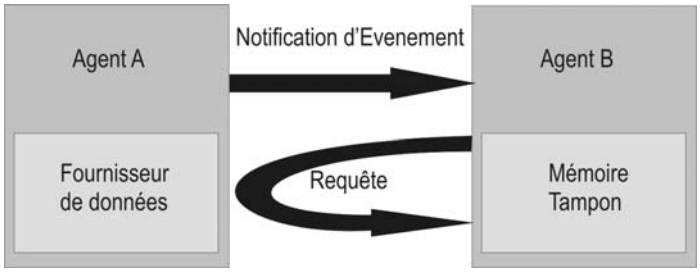


Figure 6.7 – Communication via la notification d'événements et le fournisseur de données

Ce type d'échange de données est employé toutes les fois qu'un flot prédéfini d'informations dans le processus de diagnostic est utilisé :

- Transférer des mesures à partir du DAA vers les DA.
- Transfert des symptômes créés par les DA vers le DDA et l'OIA.
- Transfert de diagnostic à partir du DDA vers le DSA et l'OIA.
- Transfert de conseils d'action et de maintenance et des prédictions à partir de DSA vers l'OIA.

VI.4.3. Communication intra-agent

En plus de la communication entre différents agents, les différentes couches d'un agent MAGIC (les couches de l'agent d'analyse diagnostique sont présentées par le paragraphe 6.2.1) responsables chacune d'une fonction spécifique de l'agent, doivent également échanger des

informations. Cette interface est réalisée sur un modèle client-serveur. Chaque couche peut être client ou serveur, selon la méthode spécifique qui est appelée. L'échange de données peut être réalisé par une méthode d'appel "à sens unique" sans valeur de retour ou par une méthode d'appel synchrone qui permet de renvoyer une valeur de retour à l'objet client.

VI.5. L'ONTOLOGIE DE MAGIC

Tout système intelligent a besoin d'une quantité considérable de connaissance sur un domaine pour être utile dans ce domaine. Dans le cas particulier du système MAGIC, l'organisation de la connaissance associée à la supervision de procédés permet d'avoir un sens unique des concepts d'automatique, de traitement de signal, d'informatique, d'ingénierie de procédé qui sont utilisés.

VI.5.1. Définition de l'Ontologie

Une **ontologie** est une conceptualisation d'un domaine à laquelle sont associés un ou plusieurs vocabulaires de termes. Les concepts se structurent en un système et participent à la signification des termes. Une ontologie est définie pour un objectif donné et exprime un point de vue partagé par une communauté. Une ontologie s'exprime dans un langage (représentation) qui repose sur une théorie (sémantique) garante des propriétés de l'ontologie en termes de consensus, cohérence, réutilisation et partage. En d'autre terme, l'ontologie est l'organisation hiérarchique de la connaissance sur un ensemble d'objets par leur regroupement en sous-catégories suivant leurs caractéristiques essentielles.

L'ontologie de MAGIC est la base de la conception du système de communication de haut niveau puisqu'elle décrit la sémantique des messages circulants. Le développement du système autour des ontologies permet la cohérence des différents types d'information existants. Dans MAGIC il est possible de distinguer trois grandes rubriques de connaissances (Figure 6.8) :

1) L'ontologie du procédé.
2) L'ontologie du diagnostic.
3) L'ontologie d'aide à l'opérateur

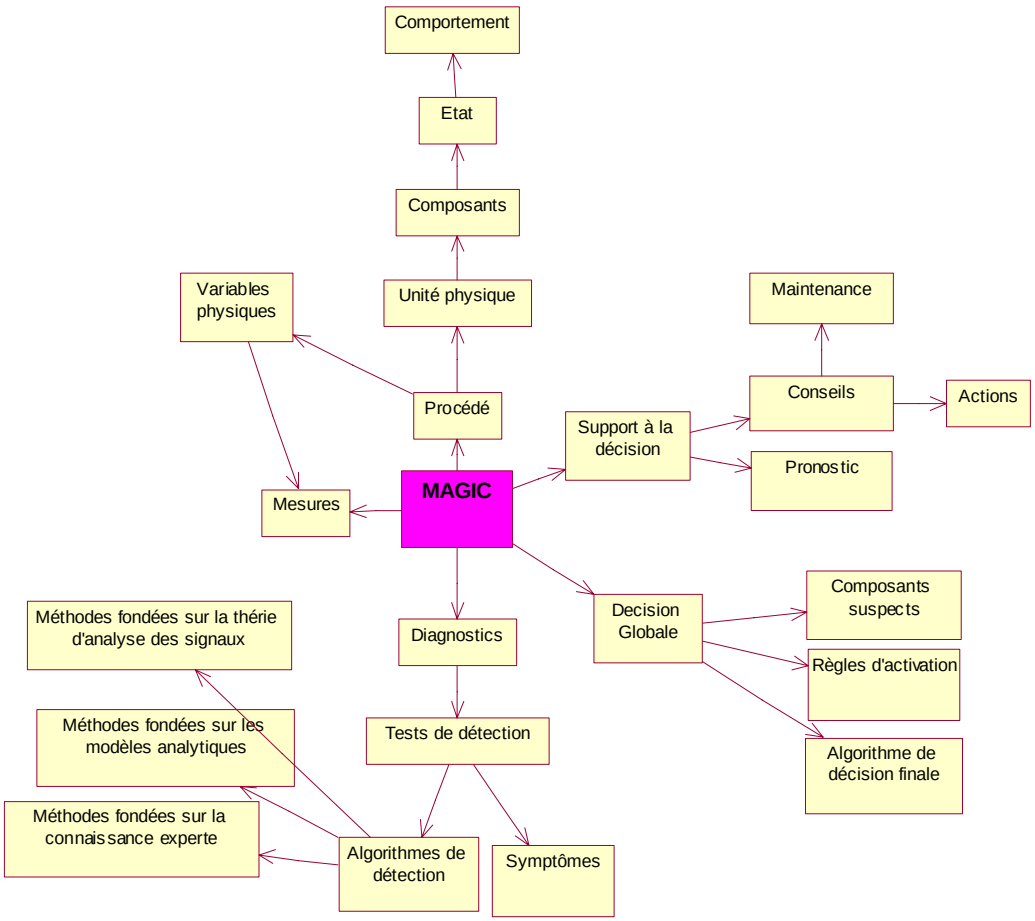


Figure 6.8 - Un diagramme des classes UML décrit une partie des éléments de l'ontologie du système MAGIC

Unité physique	Est un ensemble de composants ou d'autres unités physiques.
Composant	Est une entité physique considérée comme indivisible
État du composant	Un des états de chaque composant est "normal". Cet élément permet de discriminer/ distinguer entre l'état normal d'un composant et les autres qui peuvent être associés à des modes de défaillance particulière.
Comportement du composant	Cet élément permet d'établir la manière dont le composant doit se comporter dans un mode donné. Dans le meilleur des cas, le comportement d'un composant peut être défini par un ensemble d'équations algèbro-différentielles obtenues à partir des lois physiques qui régissent le comportement du composant. Chaque mode est décrit par un comportement différent, bien évidement.
Variable physique	Une variable physique représente une certaine quantité physique, mesurée ou pas.

Tableau 6.2- Ontologie partielle du procédé.

.

Mesures	C'est la valeur d'une variable physique obtenue avec le système d'acquisition de données du procédé.
Test de détection	C'est l'ensemble des vérifications exécutées afin d'établir si le comportement du composant ou composants particuliers est conforme à un modèle ou gabarit particulier.
Algorithmes de détection	Pour exécuter un test de détection, le système utilise diverses méthodes numériques qui permettent au système de créer les références de comportement, de comparer avec le comportement réel et de décider avec certain niveau de croyance si le comportement réel est conforme ou non à la référence.
Symptômes	Les symptômes sont les résultats des tests de détection, c'est-à-dire, ils indiquent si l'état du comportement du composant est conforme à un modèle ou gabarit particulier.
Décision de localisation	En tenant compte des divers symptômes, la décision de diagnostic représente les algorithmes utilisés pour la localisation du composant défectueux.

Tableau 6.3- Ontologie partielle du diagnostic

Diagnostic	C'est l'ensemble des composants qui sont dans un état anormal.
Conseils	C'est l'ensemble des recommandations afin d'éviter la propagation d'un défaut. Il s'agit de conseils pour la conduite du procédé ou de conseils de maintenance des composants.
Pronostic	Il s'agit des avis sur l'évolution possible du procédé sous l'effet des défauts.

Tableau 6.4- Ontologie partielle d'aide à l'opérateur

VI.5.2. Mise en application de l'ontologie

Dans MAGIC, les concepts impliqués dans l'ontologie ont été transférés dans les tableaux d'une base de données relationnelle (dans ce cas une base de données ORACLE).

Beaucoup de concepts concernant le fonctionnement des agents ont été ajoutés. Ils n'ont pas été mentionnés dans la section précédente du fait qu'ils sont seulement utilisés au niveau informatique et non au niveau conceptuel. Des exemples sont : la taille de la mémoire tampon pour la transmission, les références aux données pour la validation d'agents, etc.

Ce type d'implémentation est la plus standard au niveau industriel, voir par exemple le projet [Diamond, 2001], Protégé-2000 [Musen, Fergerson et al., (in press)]. Elle est de réutilisation facile (un des buts de toute ontologie) au moyen de méthodes d'exportation et d'importation de tableaux. De plus, elle est compatible avec les protocoles de communication déjà présentés.

VI.6. AGENT DE PRISE DE DECISION DIAGNOSTIQUE (DDA)

Le processus de diagnostic se décompose en trois parties principales. Premièrement, les différentes mesures et observations sont collectées par les capteurs via l'agent d'acquisition de données (DAA). Ces derniers récupèrent les signaux appropriés pour le diagnostic, c'est-à-dire qu'ils obtiennent les observations du système d'acquisition en temps réel, puis ils les envoient vers les agents diagnostiques (DA) après traitement (filtrage, linéarisation de caractéristiques, élimination de points aberrants, etc. [Lacatusu, Stücher et al, 2002]. Cette étape est appelée **étape d'acquisition de données**. Deuxièmement, tous les symptômes doivent être extraits du système physique à diagnostiquer. Les tests de détection encapsulés dans les agents de (DA) vérifient la consistance entre les observations du comportement réel du système physique et ses modèles de

référence. Cette étape est appelée **l'étape de génération de symptôme** ou **l'étape de la détection**. Finalement, le **raisonnement diagnostique** analyse les symptômes disponibles fournis par les Agents Diagnostic afin de déterminer les états plausibles d'un système physique. Il mène à la détermination des défauts possibles, suivant deux approches possibles : approche à base de table de signatures (voir chapitre 2), et approche logique proposée et détaillée dans le chapitre 4.

L'Agent de Décision Diagnostique (DDA) a pour but de prendre une décision globale et définitive sur l'état du système physique dans sa totalité. La difficulté principale est que les symptômes fournis par les agents DA peuvent avoir des origines différentes, c'est-à-dire des sous systèmes physiques différents. Ces derniers peuvent ne pas être indépendants, c'est-à-dire qu'ils peuvent avoir des composants en commun. Par exemple, un capteur peut être commun à plusieurs sous systèmes testés. De plus, plusieurs algorithmes de détection peuvent être assignés à un seul sous système (les symptômes sont obtenus grâce à différents algorithmes de détection tel que les observateurs d'état ou à base de traitement de signal, détection en utilisant les graphes causaux,…). Cependant, tous les symptômes doivent être analysés ensemble afin d'obtenir un diagnostic fiable.

La responsabilité du DDA est d'analyser les symptômes disponibles fournis par les agents DA afin de déterminer les états possibles anormaux du système physique. Différents types de raisonnement diagnostique sont possibles : raisonnement en utilisant les *tables de signatures* ou le *raisonnement logique* pur qui s'inspire de l'algorithme de Reiter. Les symptômes fournis par les agents de détection (D-Agents) et collectés par l'agent d'analyse diagnostique (DDA), contiennent les informations suivantes :

- La décision (une valeur qui appartient à l'intervalle [0,1]) : 0 signifie que la décision n'est pas fiable, et 1 signifie que la décision est complètement fiable. Les valeurs intermédiaires représentent l'incertitude dans la décision ;

- La validité du modèle (une valeur qui appartient à l'intervalle [0,1]) : 0 signifie que le modèle est invalide dans le contexte observé, c'est-à-dire le modèle ne peut pas être utilisé, et 1 signifie que le modèle est parfaitement applicable dans le contexte observé.

- La date de détection : c'est la date de détection du symptôme ;

- La référence du test : c'est un identificateur unique du test qui a fourni le symptôme ;

- Une référence de l'Agent de détection (D-Agents) : c'est un identificateur unique du D-Agent qui implémente le test de détection qui a fourni le symptôme ;

L'agent de décision diagnostique (DDA) reçoit des symptômes des différents tests de détection actifs. Ces tests sont encapsulés dans des agents diagnostiques. L'agent de décision diagnostique (DDA) fusionne la validité du modèle avec la décision du test avec des outils de la logique floue (voir chapitre 5) afin de pouvoir gérer l'incertitude de la décision prise par les tests et ceci dans le but d'obtenir un symptôme final. Les symptômes finaux sont internes au DDA. Ils contiennent les informations suivantes :

- La décision finale : une valeur qui appartient à l'intervalle [0, 1]. Elle est le résultat de la fusion entre la validité du modèle sur lequel repose le test de détection et la décision du test.
- La date de détection finale : Elle correspond à la date où la première détection est apparue.
- La référence au test et la référence à l'agent diagnostique sont les mêmes identificateurs générés par les agents de détection (D-Agents).

Chaque diagnostic global fourni par l'agent de décision diagnostique est exprimé en terme de composants défaillants avec un certain niveau de confiance : il reflète un état plausible du système physique à diagnostiquer. L'analyse diagnostique donne habituellement plusieurs diagnostics globaux possibles à un instant donné (en garantissant que la solution appartient à cet ensemble si les tests sont justes). Chaque diagnostic est composé de :

- un niveau de confiance ;
- une date de détection correspondant à la première fois où le diagnostic a été trouvé ;
- une liste de composants plausibles qui expriment l'état du système ;
- une méthode de diagnostic utilisée (table de la signature ou raisonnement logique).

VI.6.1. L'architecture de l'agent de décision diagnostique (DDA)

La structure proposée pour l'agent de décision diagnostique, comme tous les agents de MAGIC, est composée de trois couches principales (voir figure 6.9). La "couche supérieure" est connectée aux deux couches inférieures via CORBA. L'interface est spécifiée au moyen d'un descripteur IDL (Interface Definition Language). L'IDL définit les méthodes qui doivent être appelées par chacune des deux couches interconnectées de la structure de l'agent DDA. Les trois couches sont :

- La **couche spécifique du DDA** implémente les algorithmes de décision diagnostique responsables de l'analyse diagnostique, c'est-à-dire, analyse les symptômes fournis par les agents de détection (D-Agents) afin de produire des diagnostics globaux. Cette couche est connectée par le bas à deux autres couches (interface de communication) via l'interface CORBA. La couche spécifique de l'agent de décision diagnostique transmettra ces résultats diagnostiques via l'interface CORBA vers les couches inférieures qui permettent de les communiquer vers d'autres agents.
- La **couche de communication spécifique du DDA** fournit les services de communication spécifiques et nécessaires pour la couche « supérieure » (couche spécifique du DDA), et qui ne peuvent pas être directement fournis par la couche inférieure de communication. Cette couche joue un rôle intermédiaire entre la couche spécifique du DDA et la couche de communication.
- La **couche de communication générique** manipule la communication avec d'autres agents et inclut les fonctionnalités « génériques » d'accès (communes à tous les agents), tel que l'accès à la base de données, et permet au DDA d'être intégré dans l'infrastructure du système MAGIC et ainsi pouvoir communiquer avec les autres agents de MAGIC.

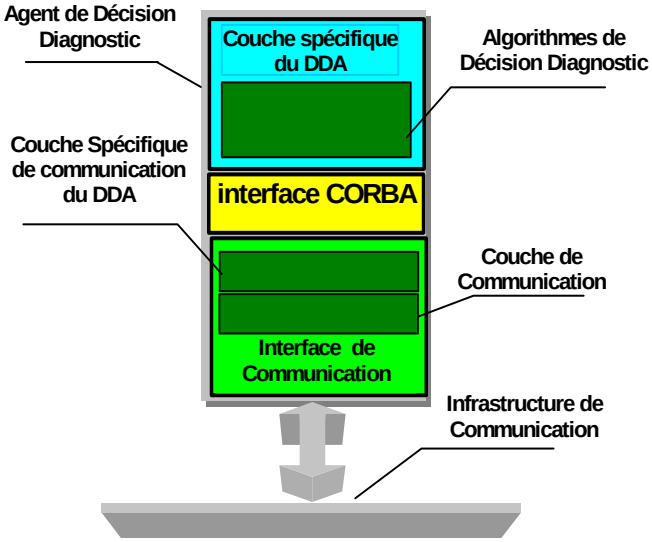


Figure 6.9- Structure interne de l'agent DDA

VI.6.2. Les fonctionnalités générales du DDA

L'agent de Décision Diagnostique (DDA) interagit avec d'autres agents du système MAGIC. Le DDA doit fournir diverses fonctionnalités. Parmi l'ensemble de ces fonctionnalités, on présentera ci-dessous seulement celles implantées dans la couche spécifique du DDA (c'est-à-dire la partie cerveau de l'agent). En revanche, les fonctionnalités liées à la communication ont été déjà présentées de façon générale précédemment. L'agent de Décision Diagnostique doit réaliser les fonctions suivantes :

- Analyser les symptômes fournis par les différents agents de détection (DA) et cela afin de prendre une décision globale et donner l'information sur l'état réel du système (fournir l'ensemble des diagnostics) ;
- Estimer le niveau de confiance de chaque diagnostic global grâce à la logique floue et les proposer à l'utilisateur du système MAGIC par ordre de plausibilité ;
- Sauvegarder l'historique des diagnostics dans la base de données ;
- Gérer le processus de diagnostic en activant les tests de détection pertinents dans les agents DA suivant une stratégie prédéterminée ;
- Détecter les tests de détection qui mènent à des diagnostics erronés.

Pour que l'agent DDA réalise ces fonctionnalités, l'agent a besoin d'échanger les informations suivantes :

L'agent de Décision Diagnostique (DDA) a besoin des informations suivantes en provenance des agents de détection (D-Agents) :

- L'ensemble des composants appartenant à chacun des supports des tests de détection avec les états modélisés (cette information est collectée par l'agent PSA (*Process Specification Agent*) et stockée dans la base de données). L'agent de décision diagnostique récupère cette information directement de la base de données.
- Les symptômes fournis par les agents diagnostiques (D-Agents) qui incluent décisions et validités des tests. Chacun des agents de détection (D-agents) doit décider si un test de détection est vrai ou faux. Les symptômes peuvent venir d'un processus de diagnostic en mode permanent ou périodique ou bien sur demande qui couvre une fenêtre de temps spécifiée.

L'agent de Décision Diagnostique a besoin des informations suivantes en provenance de l'agent de spécification du procédé (PSA) :

- La probabilité a priori pour qu'un composant tombe en panne (cette information est saisie par l'agent de spécification du procédé et stockée dans la base de données),
- Les hypothèses sur l'état des composants impliqués dans la construction des tests de détection dans les agents D-Agents.
- La stratégie de déclenchement des tests de détection définie a priori (une seule stratégie possible pour le mode permanent ou périodique en revanche, plusieurs stratégies peuvent être définies pour le mode sur demande).

Les besoins généraux du DDA sont résumés avec le schéma UML appelé « diagramme des cas d'utilisation » (voir figure 6.10). Ce diagramme illustre les aspects utilitaires du système, et identifie ses acteurs (entités externes réagissant au DDA, les accointances de l'agent DDA). Chaque fonctionnalité ou cas d'utilisation est représenté par un cercle. Les acteurs représentent les autres agents du système MAGIC. Les arcs pleins indiquent, selon leur direction, les agents qui peuvent

demander à ou bénéficier des services de l'agent DDA. En revanche, la position des arcs pointillés porte des inclusions de fonctionnalités.

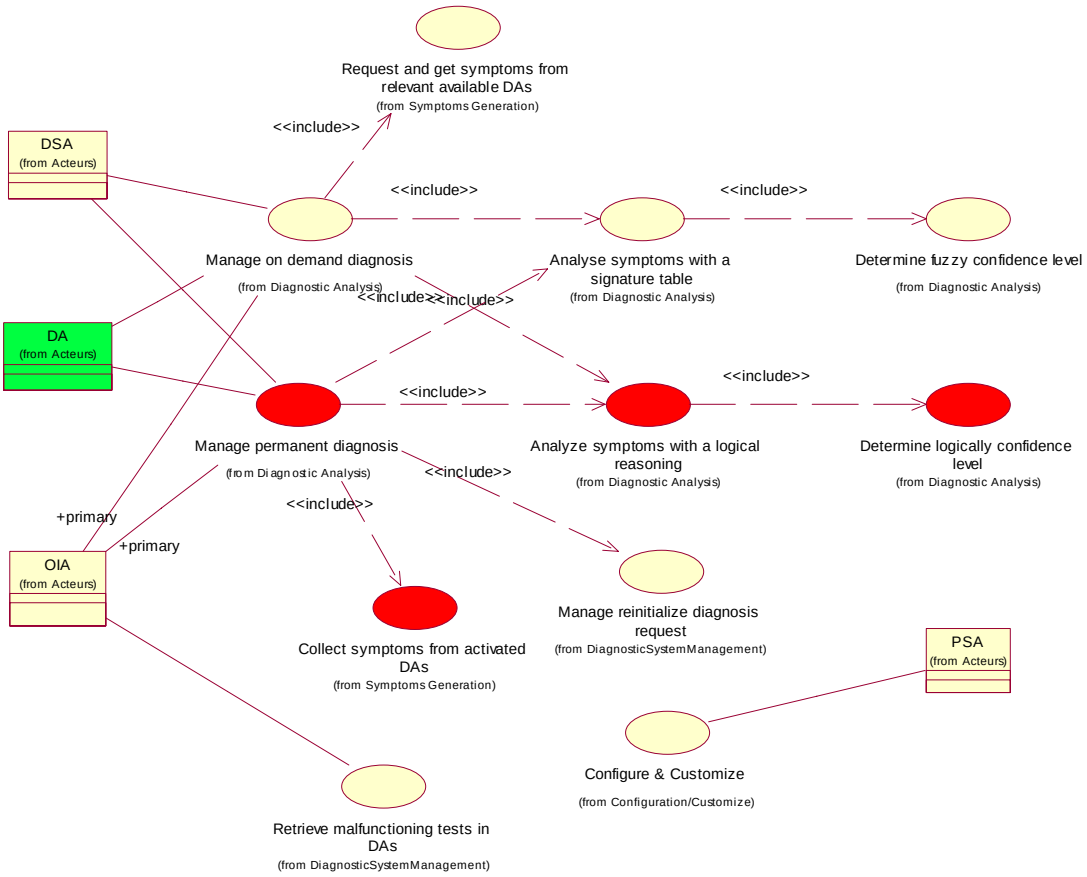


Figure 6.10- Fonctionnalité générale du DDA (Diagramme UML de cas d'utilisation fonctionnelle)

Cas d'utilisation	Description
Configurer	Collecter l'information concernant la stratégie de diagnostic choisie (règles d'activation), les tests de détection disponibles, méthode de diagnostic à utiliser lors de l'analyse diagnostique (table de signature, approche logique).
Diagnostic sur demande	Le diagnostic sur demande peut être demandé par le EUSE afin de réaliser une analyse diagnostique sur une période de temps passée choisie. Le diagnostic sur demande permet de choisir la méthode d'analyse diagnostique.
Déterminer les agents diagnostiques pertinents	Déterminer les agents pertinents qui devront être activés
Demande et obtention des symptômes des agents diagnostiques actifs	Le DDA demande et collecte les symptômes des agents diagnostiques actifs sur une fenêtre temporelle.
Analyser les symptômes	Détermination des diagnostics possibles (i.e. les états possibles du système).
Utiliser le raisonnement par table de signatures	Utilisation du raisonnement par table de signatures afin de déterminer les diagnostics
Utiliser le raisonnement logique	Utilisation du raisonnement logique afin de déterminer les diagnostics
Déterminer le niveau de confiance	Utiliser l'approche de la logique floue afin d'évaluer le niveau de confiance des diagnostics.
Diagnostic permanent ou périodique	La procédure de diagnostic peut être soit permanente soit périodique. Mais, dans les deux cas, les symptômes ne sont pas demandés par l'agent DDA mais automatiquement fournis par les agents de détection (DA) au DDA.
Collecter les symptômes des agents diagnostiques	Le DDA reçoit différents symptômes des différents agents diagnostiques. Les symptômes reçus sont collectés et analysés tous ensemble pour donner un diagnostic fiable.
Retrouver les tests qui donnent des fausses alarmes dans les agents diagnostiques	Si le système MAGIC donne un faux diagnostic et que l'opérateur indique le vrai état du système à l'agent de décision diagnostique (DDA) via l'agent interface opérateur (OIA) alors le DDA doit être capable de retrouver le test de détection qui a mal fonctionné et d'en informer l'opérateur.
Gérer l'arrêt du diagnostic sur demande	Pouvoir arrêter l'analyse diagnostique sur demande de l'utilisateur.
Gérer la réinitialisation du diagnostic sur demande	Après que l'ingénieur utilisateur final du système MAGIC répare le défaut sur le système diagnostiqué alors l'agent de décision diagnostique (DDA) permet la réinitialisation des symptômes qui ont servis pour l'analyse diagnostique.

Table 6.4 – La description des cas d'utilisation fonctionnelle du DDA

VI.6.3. Fonctionnement du DDA

La figure 6.11 illustre les parties principales implémentées dans la couche spécifique du DDA. La structure de la couche spécifique du DDA est divisée en 4 parties :

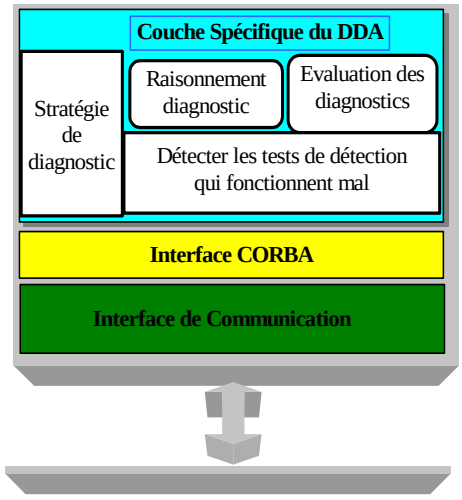


Figure 6.11 - L'agent de décision diagnostique (DDA)

- **Le raisonnement diagnostique** : cette partie contient deux types d'algorithmes d'analyse diagnostique : l'approche à base de table de signatures (communauté FDI), l'approche à base de la consistance (approche logique) (communauté DX). L'approche logique analyse des symptômes incompatibles et qui peuvent être conflictuels (raisonnement diagnostique) en utilisant les modèles de bon et de mauvais fonctionnement (voir chapitre 4). Cette approche a besoin d'une description du système assez précise (voir chapitre 3).

- **Niveau de confiance des diagnostics** : cette partie est dédiée à l'estimation des plausibilités des diagnostics (circonstancielles, a priori et formelles). Ces estimations se font grâce aux outils de la logique floue (chapitre 4). Les niveaux de confiance des résultats des tests de détection fournis par les agents DA, le taux de défaillance a priori des composants, la compatibilité du diagnostic global avec les tests vrais sont pris en compte par cette partie du système afin de pouvoir évaluer chaque diagnostic et les classer par ordre de plausibilité.

- **Détecter les agents qui ont mal fonctionné** : cela revient à retrouver les tests de détection qui ont mal fonctionné et qui ont donné des erreurs de diagnostic. Cette partie peut se faire une fois que le véritable diagnostic est retourné par l'opérateur. Si l'ingénieur utilisateur final du système MAGIC intervient sur le système à diagnostiquer et connaît le véritable diagnostic qui n'a pas été trouvé par le système MAGIC, une exception doit être lancée afin de retrouver le test de détection qui a mal fonctionné.

- **Stratégie de diagnostic :** Elle gère la politique de déclenchement des tests de détection, c'est-à-dire les règles d'activation des tests de détection encapsulées dans les agents DA. En effet, le DDA est capable d'interpréter la demande de l'utilisateur concernant le choix de la stratégie de diagnostic la plus pertinente à déclencher, c'est-à-dire, la meilleure stratégie pour lancer les tests de détection. La stratégie de diagnostic gère la génération de symptômes ainsi que le raisonnement diagnostique suivant les spécifications de l'utilisateur final telle que la précision du diagnostic final et la vitesse du processus de diagnostic.

Le diagramme d'activité UML de la figure 6.12 résume les opérations ainsi que les scénarios typiques de l'agent DDA.

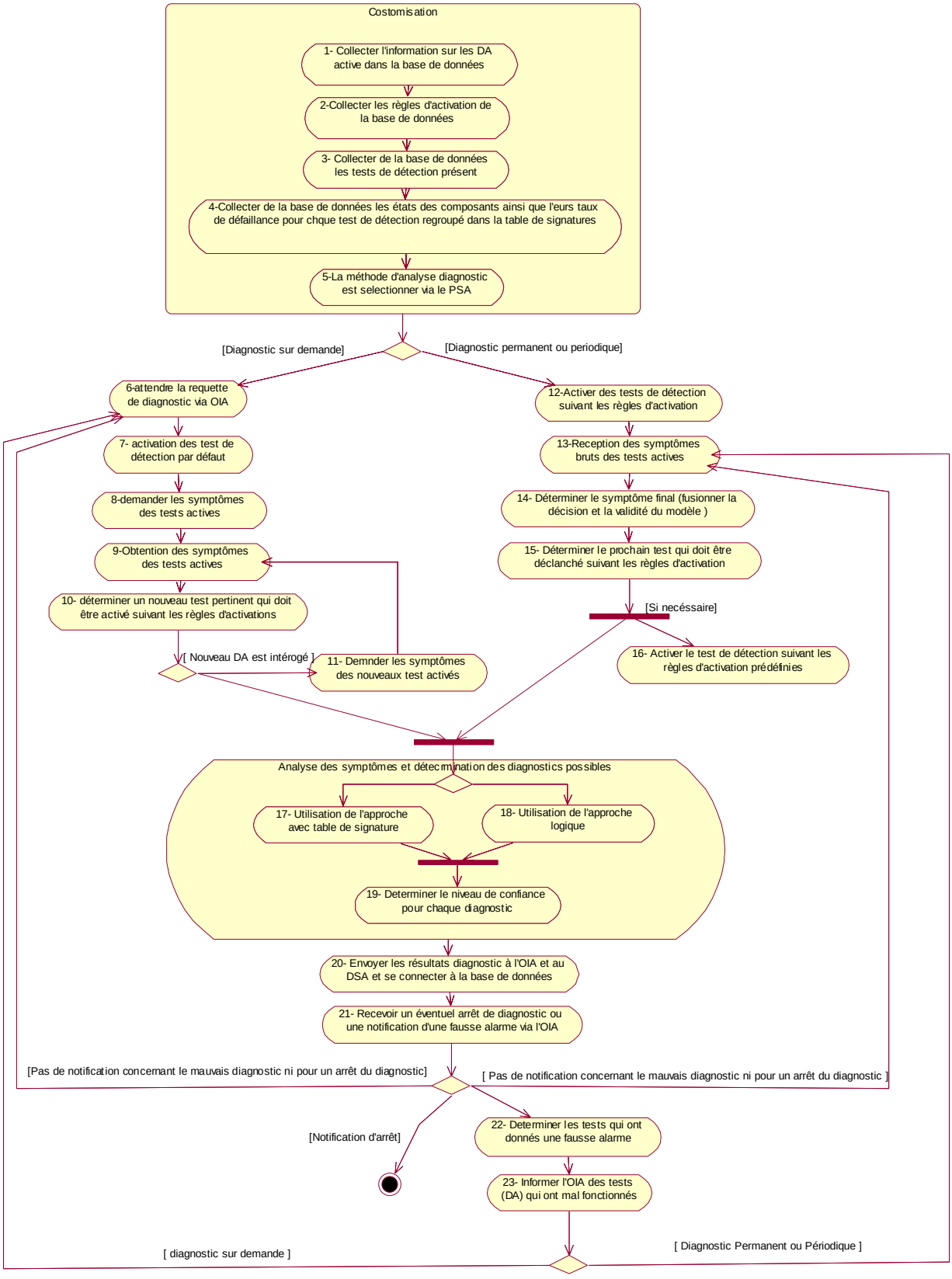


Figure 6.12– Diagramme UML d'activité de l'agent DDA

VI.6.4. La communication et la coopération

La figure 6.13 illustre l'interaction entre l'agent DDA et les autres agents du système MAGIC. La couche de communication du DDA est responsable de la gestion de cette communication en utilisant l'Infrastructure de communication MAGIC.

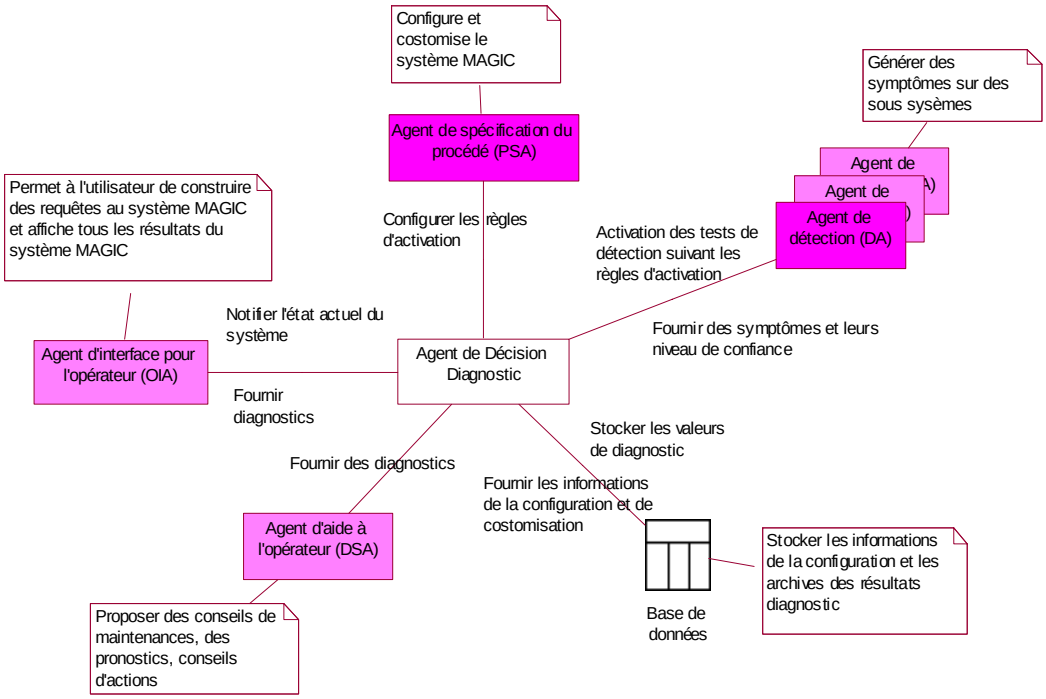


Figure 6.13– Principales relations entre l'agent de décision diagnostique (DDA) et les autres agents MAGIC

Une description courte des relations fonctionnelles est donnée par la table 6.5. Les actions facultatives dépendent de la conception et des choix effectués lors de la configuration.

Description de la communication	Description
DDA ↔ DA	Cette communication est bidirectionnelle, les différents agents DA fournissent les symptômes. Le résultat est formalisé en terme de : symptômes et leurs niveaux de confiance, Le DDA peut communiquer avec les agents de détection (DA) en les activant suivant les règles d'activations prédéfinies a priori.
DDA→DSA	La communication est unidirectionnelle, le DDA envoie à l'agent d'aide à l'opérateur (DSA) un ensemble de diagnostics à exprimer en terme de composants défaillants avec un degré de plausibilité.
DDA↔OIA	Cette communication est bidirectionnelle. L'agent de décision diagnostique (DDA) envoie à l'agent d'interface pour l'opérateur (OIA) un ensemble de diagnostics avec leurs niveaux de confiance. Quand le système fonctionne en mode « sur demande », l'OIA peut demander au DDA de lui faire un diagnostic sur une période de temps spécifique. L'ingénieur utilisateur final du système MAGIC avertit le DDA via l'agent d'interface pour l'opérateur qu'un diagnostic est faux et qu'il s'agit probablement d'une fausse alarme fournie par un test du système MAGIC: le vrai état du système doit être fourni par l'ingénieur utilisateur final du système MAGIC afin que le DDA puisse trouver le test de détection qui a mal fonctionné.
DDA↔Base de données	Les diagnostics sont stockés dans la base de données pour chaque composant.

Table 6.5 – Description des relations fonctionnelles du DDA dans le système MAGIC

VI.7. APPLICATION INDUSTRIELLE

Afin de valider les aspects théoriques développés dans les chapitres 3 et 4, le raisonnement diagnostique logique a été appliqué à la détection de défauts d'un enrouleur hydraulique[6] [Stücher et al, 2004], composant utilisé dans le procédé de laminage à chaud. Ce composant est en contact direct avec le métal chauffé au rouge ; il est nécessaire au maintien de la bande de métal à une tension constante. Cette application représente un premier test industriel des algorithmes de diagnostics développé dans le projet MAGIC.

VI.7.1. Description du procédé de laminage à chaud

Le principe de base du laminage par roulement est la modification de la forme du métal par déformation plastique obtenue par l'action d'outils de rotation appelés les cylindres de travail. Il est habituel de traiter le métal en plusieurs étapes et par différents cylindres en réduisant progressivement les dimensions du produit.

Le laminage commence par une phase à chaud afin d'effectuer un premier traitement des pièces de métal de grande section ; cette phase est généralement précédée par le coulage du métal. La chaleur donne au métal l'élasticité nécessaire pour produire les premières grandes déformations. Après que le métal ait été traité dans le stand de dégrossissage pour le ramener à l'épaisseur

[6] Le nom en anglais utilisé pour ce composant est *hydraulic looper*. Ce nom vient du fait que pour compenser les différences de vitesse de la bande de métal entre deux rouleaux, le «looper » crée un demi boucle (loop) avec la bande de métal.

nominale exigée, il traverse une série de stands de finition où il est laminé jusqu'à son épaisseur finale avec certaines propriétés mécaniques et de qualité extérieure. Cette phase garantit la qualité du produit [Wakamiya et Nitta, 2000][Montmitonnet, 2002]. Le traitement entier est montré sur la figure 6.1.

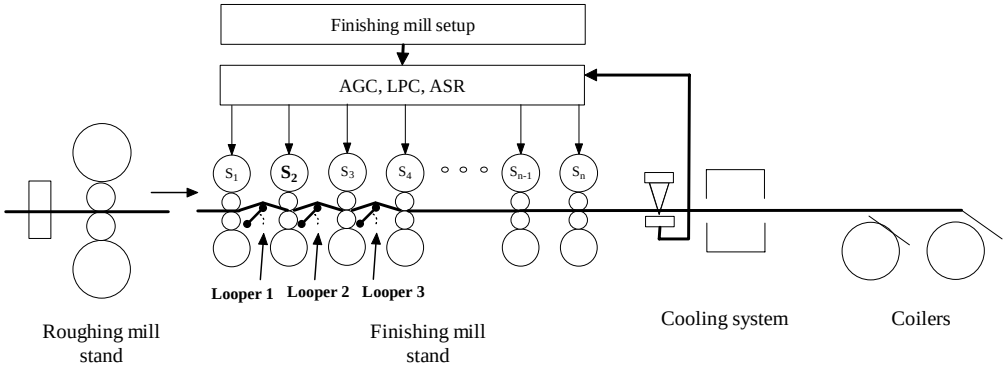


Figure 6.14 - Procédé de laminage à chaud

VI.7.2. Description de l'enrouleur hydraulique et son rôle dans le laminoir

Les enrouleurs hydrauliques sont situés typiquement entre deux stands dans le laminoir. Ils mesurent et éliminent les perturbations dans la circulation de la masse, ils maintiennent aussi la tension de bande afin de la maintenir aussi constante que possible. L'enrouleur se décompose en trois parties :

- Un contrôle électrique ;
- Une transmission hydraulique, composée de tuyaux, de servo-valves et de cylindres hydrauliques
- Un système mécanique

La liste des composants qui apparaît sur la figure 6.15 est la suivante : C_1 : La servo-vanne ; C_2 : Le tuyau hydraulique du côté de la tige ; C_3 : Le tuyau hydraulique du côté du piston ; C_4 : Le capteur de pression du côté du piston; C_5 : Le capteur de pression du côté de la tige; C_6 : Le cylindre hydraulique ; C_7 : L'articulation mécanique; C_8 : Le capteur de position angulaire.

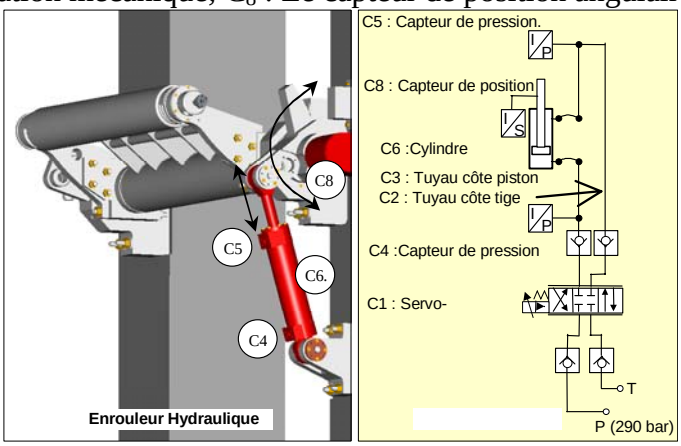


Figure 6.15- Schéma mécanique et hydraulique de l'enrouleur hydraulique

VI.7.3. L'analyse diagnostique

Dans l'implantation industrielle effectuée par l'entreprise SMS-DEMAG, le système MAGIC fonctionne en combinaison avec un système commercial de surveillance « actif » appelé Système de Surveillance du Système et du Procédé (*PPMS = Plant and Process Monitoring System*). Ce système fournit des séquences spéciales de test pour différentes composantes du procédé de laminage [Deckers, 2003]. Les tests effectués par le PPMS sur l'enrouleur hydraulique sont toujours exécutés après un changement de cylindres de travail d'un stand de laminage, c'est-à-dire, en dehors du processus de laminage. Les tests sont réalisés généralement quatre fois par jour mais sans une période fixe. Ces tests ont été décrits dans [Stücher et al, 2003] et [Garcia-Beltran, 2004]. Quatre types d'essai sont réalisés :

- ***Test de position maximale*** - Dans une configuration de commande de position en boucle fermée, l'enrouleur hydraulique est déplacé jusqu'à sa position maximale ; un interrupteur de position indique au système de commande que l'enrouleur est dans la position angulaire maximale. Dans cet état, les valeurs ponctuelles de la position angulaire $x_{meas}^{\max}$ de l'enrouleur et de la force $f^{\max}$ (mesurée ou calculée à partir des pressions) nécessaires pour amener l'enrouleur à cette position, sont stockées.

- ***Test de position minimale*** - Dans une configuration de commande de position en boucle fermée, l'enrouleur hydraulique est déplacé jusqu'a sa position minimale ; un interrupteur de position indique au système de commande que l'enrouleur est dans la position angulaire minimale. Dans cet état, les valeurs ponctuelles de la position angulaire $x_{meas}^{\min}$ de l'enrouleur et de la force $f^{\min}$ (mesurée ou calculée à partir des pressions) nécessaires pour faire arriver l'enrouleur à cette position, sont stockées.

- ***Test de réponse à un échelon de consigne de force*** - L'enrouleur est positionné à l'extrémité supérieure. Dans cette position, et dans une configuration de régulation de la force, une consigne en échelon positif puis en échelon négatif est appliquée au système. L'enrouleur agit contre un élément plastique de basse élasticité situé à l'extrémité fixe de l'enrouleur, ce qui bloque son déplacement. La trajectoire de la réponse de la force (mesurée ou calculée à partir des pressions), la variable de commande ainsi que la consigne de force sont enregistrées.

- ***Test de réponse à une rampe de consigne de position (test d'hystérésis)*** - l'enrouleur est déplacé entre une position inférieure de 10° et une position supérieure de 70° en suivant une consigne de position croissante/décroissante. Les trajectoires de la position angulaire de l'enrouleur et de la force (mesurée ou calculée à partir des pressions) sont enregistrées. Avec ces deux signaux, la courbe d'hystérésis de la force en fonction de la position est déduite.

D'après la description du système par son modèle et les mesures disponibles, tous les sous-systèmes testables possibles ont été calculés grâce à l'algorithme de génération automatique de sous-systèmes testables présenté dans le chapitre 3.

La table 6.6 résume la composition des trois tests de détection $test_1$, $test_2$ et le $test_3$. Chaque test de détection t_i se base sur la conjonction d'état de composants donnée par $\bigwedge_j \neg AN(C_j)$, c'est-à-

dire, le test de détection *test₁* vérifie si les composants C_1, C_2, C_3, C_4, C_5 et C_6 sont dans un état normal. Par conséquent, le test "$test_1$" est sensible aux défauts qui peuvent apparaître sur le composant C_1 mais insensible aux défauts qui peuvent apparaître sur les composants C_7 et C_8.

¬AN(.)	**C_1**	**C_2**	**C_3**	**C_4**	**C_5**	**C_6**	**C_7**	**C_8**
Test1	1	1	1	1	1	1		
Test2				1	1	1	1	1
Test3	1	1		1	1	1	1	

Table 6.6 – Table de signatures

Les tests "$test_1$" et "$test_2$" ont été réalisés par des agents de détection qui implémentent des tests à base d'un algorithme causal [Garcia-Beltran et al, 2003]. En revanche, le troisième test "$test_3$" est réalisé par un agent de détection qui implémente un algorithme basé sur un réseau de neurones [Marcu etal, 2003]. Les résultats fournis par les tests de détection (t_1, t_2, t_3) correspondent à des décisions brutes données par une fonction d'appartenance du résultat des tests $(\mu_{\neg\sigma}) = (1, 0.2, 0.75)$ et une fonction d'appartenance de la validité des tests $(\mu_{\upsilon}) = (0.6, 1, 0.8)$.

En appliquant la formule (3.11) développée dans le chapitre 3 pour fusionner ces résultats en symptômes finaux, on obtient :

$$\left(\mu_{\neg\hat{\pi}(t_1)}, \mu_{\neg\hat{\pi}(t_2)}, \mu_{\neg\hat{\pi}(t3)}\right) = (0.8, 0.2, 0.7) \qquad (6.1)$$

La plausibilité circonstancielle a été calculée d'après la formule (4.13) et la plausibilité formelle a été calculée d'après la formule (4.27). Ces résultats sont résumés dans la table 6.7

Diagnostics	AN(C1)	AN(C2)	AN(C3)	AN(C4)	AN(C5)	AN(C6)	AN(C7)	AN(C8)
Plausibilité Formelle	0.8	0.8	0.8	0.8	0.8	0.8	0.7	0.2
Plausibilité Circonstancielle	0.77	0.77	0.63	0.57	0.57	0.57	0.37	0.23

Table 6.7 – Les diagnostics logiques

Parce qu'aucune des décisions n'est sûre, les diagnostics sont plutôt des pré-diagnostics car ce n'est pas certain qu'un défaut soit véritablement présent dans le système. Dans ce contexte, chaque diagnostic possible implique seulement un composant (chapitre 4, paragraphe 4.7.2). La table 6.7 montre formellement que, chaque composant de l'enrouleur hydraulique est suspecté, bien que des défauts au niveau des composants C_8 (capteur de position angulaire) et C_7 (bras mécanique) soient moins vraisemblable. Parmi ces composants, le score circonstanciel donne une plus grande importance au défaut sur C_1 (valve) ou sur C_2 (Le tuyau hydraulique du côté de la tige) et ceci parce que leurs signatures sont plus proches des symptômes donnés par (6.1). Ils doivent être vérifiés par l'opérateur en premier lieu. On note aussi que les composants C_1 et C_2 sont indistinguables autant que les composants C_4, C_5 et C_6 parce qu'ils ont la même signature dans table de signatures donnée par la table 6.6. Par conséquent, ils obtiennent toujours les mêmes scores.

Considérons la situation où les trois tests de détection (t_1, t_2, t_3) donnent les résultats bruts suivant $(\mu_{\neg\sigma}) = (1, 0.25, 1)$ et $(\mu_{\upsilon}) = (1, 0.8, 1)$

Ces résultats bruts mènent aux symptômes suivants :

$$\left(\mu_{\hat{\vec{r}}(t_1)}, \mu_{\hat{\vec{r}}(t_2)}, \mu_{\hat{\vec{r}}(t3)}\right) = (1, 0.3, 1)$$

Les diagnostics sont résumés dans la table 6.8 ci-dessous :

Diagnostics	AN(C_1)	AN(C_2)	AN(C_4)	AN(C_5)	AN(C_6)	AN(C_3) ^ AN(C_7)
Plausibilité Formelle	1	1	1	1	1	1
Plausibilité Circumstantielle	0.9	0.9	0.77	0.77	0.77	0.77

Table 6.8 – Les diagnostics logiques

Les symptômes révèlent avec certitude une anomalie (décision égale à 1). Les diagnostics ne sont donc plus des pré-diagnostics. Ils ont une plausibilité formelle égale à 1. Les tests 1 et 3 sont faux à 100% alors que le test 2 ne l'est qu'à 30%. Les diagnostics minimaux correspondent à ceux fournis par l'algorithme présenté dans le chapitre 4. Les résultats obtenus pour la plausibilité circonstancielle (formule 4.13 décrite dans le chapitre 4) privilégient les défauts sur les composants C_1 ou C_2 (plausibilité de 90%) indiscernables. Les autres composants n'ont une plausibilité qui ne vaut que 77%.

VI.8. CONCLUSION

Le système MAGIC offre non seulement une aide au diagnostic lorsqu'un défaut s'est produit mais il permet aussi de détecter des défauts en voie de développement dans les systèmes complexes. Le calcul de pré-diagnostics permet d'envisager des opérations de maintenance préventive et ainsi de maximiser la disponibilité et l'efficacité des installations [Aström et al., 2001].

Une architecture basée sur le paradigme des systèmes multi-agents vient d'être présentée, cette architecture a été conçue dans le cadre d'un projet européen intitulé MAGIC (*Multi-Agent-based Diagnostic Data Acquisition and Management in Complex Systems*). Les agents exécutent eux mêmes les tâches du processus de diagnostic comme l'acquisition et la gestion des données, le choix des outils diagnostiques pour la détection, la gestion des conflits et la prise de décision, ainsi que l'information et l'aide à l'opérateur. Les « agents » conduisent à des composants logiciels, qui peuvent s'exécuter sur différents environnements. Chaque agent coopère avec ses accointances et a une ou plusieurs responsabilités spécifiques. Ils forment une architecture spatialement distribuée avec des éléments modulaires et extensibles qui sont fonctionnellement et sémantiquement bien identifiés.

Un des objectifs principaux de MAGIC est la détection et le diagnostic des défauts en voie de développement en temps réel dans les systèmes complexes. La détection des défauts, avant qu'ils se produisent complètement, permet ainsi soit à l'opérateur de planifier ces opérations sur le système, soit au système de contrôle automatique de compenser le défaut. Cela permet d'avoir une production continue et d'assurer ainsi une qualité constante du produit. L'identification précoce des défauts potentiels sur le système fournit des informations clés pour l'application d'un régime de maintenance prédictive. Par conséquent, le temps d'interruption des machines est minimisé, et la production est ainsi maintenue, et l'usage du personnel d'entretient est optimisé, de cette façon on

maximise la disponibilité et l'efficacité de nos machines. Le système MAGIC sera un outil de grand usage pour les ingénieurs du procédé et maintenance.

Notre travail a porté sur la conception et la réalisation d'un agent de décision diagnostique (Diagnostic Decision Agent) dans le système MAGIC. Cet agent a pour rôle principal de calculer des diagnostics globaux en tenant compte de tous les tests de détection disponibles, aussi divers soient-ils. L'algorithme de diagnostic proposé s'appuie sur une logique formelle qui a été étendue à la logique floue pour permettre d'appréhender des incertitudes de décision. Ces incertitudes sont en effet courantes car il est rare qu'un test parvienne à détecter avec certitude une anomalie. Les mécanismes de seuillage au niveau des algorithmes de détection permettent d'ajuster la sensibilité du système de diagnostic. Par ailleurs, le DDA permet aussi de remonter au test de détection qui a donné des fausses alarmes.

Nous avons pu réalisé un test du système MAGIC sur un système physique qui est « Un Enrouleur Hydraulique ». Bien que les tests de détection construits pour ce composant soient très simples, les algorithmes d'analyse diagnostique pourraient être utilisés avec des systèmes plus complexes sans perte de généralité. L'Enrouleur Hydraulique n'était toutefois pas assez complexe pour permettre de mettre en valeur correctement l'intérêt de l'analyse diagnostique logique.

Conclusion et perspectives

CONCLUSION

Les systèmes industriels sont devenus de plus en plus sophistiqués avec d'une part les systèmes de pilotages numériques et d'autre part les solutions technologiques pour l'informatique distribuée et hiérarchisée. Du fait de ces évolutions, l'implantation d'algorithmes de diagnostic au sein d'installations industrielles est devenue possible. Pourtant, si la recherche en Automatique s'est beaucoup intéressée aux systèmes de faible complexité (moins d'une dizaine de variables), les systèmes industriels sont généralement complexes et requiert des garanties de résultats. Lorsqu'il a été demandé à un industriel allemand participant au projet européen MAGIC, s'il était acceptable qu'un système de diagnostic puisse fournir un mauvais diagnostic, sa réponse fut positive, mais d'ajouter : prouver que l'erreur ne se reproduise plus. Les systèmes de diagnostic doivent par conséquent offrir des garanties de fonctionnement pour intéresser l'industrie pour qu'en cas de mauvais diagnostic, il soit possible de remonter aux tests ayant fournis des résultats erronés et être ainsi capable de prendre des mesures correctives.

Pour que les méthodes de diagnostic puissent s'appliquer à des systèmes complexes, il est crucial de concevoir des méthodologies et des technologies adaptées. Les systèmes de diagnostic doivent être conçus pour supporter de nombreuses procédures de test se déclenchant les unes les autres et fonctionnant souvent en parallèle et en des lieux différents. A un niveau supérieur, les résultats des tests de détection doivent être analysés pour conduire à un diagnostic fiable tenant compte du passé du système surveillé.

Cette thèse propose une méthodologie pour la conception de systèmes de diagnostic fiables permettant d'appréhender les systèmes dynamiques complexes et spatialement distribués. Les résultats proposés s'appuient d'une part sur des techniques d'analyse diagnostique formelle ou à base de consistance, qui permettent de garantir la justesse de l'analyse diagnostique, et d'autre part, sur le paradigme multi-agent pour concevoir une procédure de diagnostic distribuée. Les algorithmes proposés permettent de déduire tous les défauts possibles pour un comportement observé en les classant suivant différents critères de vraisemblance.

Notre contribution consiste à proposer une méthode de diagnostic qui distingue la phase de génération de symptômes qui peut se faire à base de techniques variées et hétérogènes et parfois très sophistiquées (observateur d'état, relation de parité, traitement de signal, causal…), de la phase de localisation ou d'analyse diagnostique, qui doit permettre de garantir ce qui peut l'être et d'analyser toutes les informations disponibles pour en déduire le diagnostic le plus juste et complet possible. Pour ce faire, la première phase de détection nous l'avons basée sur des approches FDI (communauté Automatique) qui sont et les mieux adaptées aux systèmes dynamiques et complexes. En revanche, la deuxième phase se construit en utilisant des algorithmes formels d'analyses diagnostic de même type que ceux qu'on trouve dans la communauté (DX) (communauté

d'Intelligence Artificielle). Ces algorithmes ont été adaptés de telle sorte qu'ils soient capables d'appréhender des tests de détections hétérogènes.

Un test de détection repose nécessairement sur un sous-système testable, il faut adjoindre à ce sous-système un domaine de validité correspondant à l'intersection des domaines de validité de chacune des relations comportementales élémentaires. Le domaine de validité d'un modèle testable est important car le test de détection en hérite : un résultat n'est valide que si l'état du système étudié appartient au domaine de validité qu'il est crucial de formaliser.

Nous avons montré qu'il était possible d'appréhender des incertitudes de décision au niveau des tests de détection. Tous les tests de détection ne conduisent pas toujours à des décisions sûres. L'incertitude se caractérise par l'incapacité de savoir si un test de détection est vrai ou faux. Les tests de détection mise en œuvre dans le cadre du projet MAGIC fournissent des décisions comprises entre 0 et 1 où 1 représente une alarme sûre et 0 représente qu'il n'y a aucune alarme d'une manière sûre. L'analyse diagnostique proposée permet d'exploiter ces résultats en transposant la logique de l'analyse diagnostique en logique floue.

Le travail présenté dans ce mémoire a été développé dans le cadre du projet européen MAGIC (*Multi-Agents-Based Diagnostic Data Acquisition and Management in Complex systems*). Les résultats sont en cours de transfert vers différentes industries grâce aux partenaires industriels du projet : les entreprises SATE (*System Advanced Technologies Engineering*) et SMS-DEMAG. Un des principaux avantages de travailler sur ce type de projets est l'intéressante possibilité de la mise en oeuvre et l'application réelle des algorithmes conçus sur le plan théorique et, dans le cas de MAGIC, de pouvoir connaître d'autres outils proposés par les autres participants, compte tenu de la nécessité d'intégration au niveau de tous les « *Work Package* » (WP). La mise en oeuvre informatique demande l'acquisition et l'application de connaissances et habilités de divers domaines de l'ingénierie, notamment l'automatique et l'informatique industrielle. Un défi important dans ce projet a été d'essayer de généraliser l'applicabilité des outils vers différents types de processus. Beaucoup de solutions "*ad-hoc*" ont été abandonnées, ou certaines des fonctionnalités potentielles des algorithmes n'ont pas été explorées dans leur totalité, dû au fait qu'elles restaient hors des objectifs initiaux du projet.

▪ PERSPECTIVES

Le travail effectué dans cette thèse laisse un nombre de perspectives ouvertes :

- Les méthodes d'analyse diagnostique proposées présupposent que les variables physiques présentes dans les conditions de validité sont aussi présentes dans les relations comportementales. Si cette hypothèse est vérifiée la plupart du temps dans les systèmes continus, il en va tout autrement des systèmes à évènements discrets. Or, lever cette hypothèse a des répercussions importantes tant au niveau de la conception des SST qu'au niveau de l'analyse diagnostique. Des travaux importants restent à mener dans cette voie afin d'accroître plus encore la générécité de l'approche diagnostique proposée.

- Une réflexion approfondie s'impose quant à l'utilisation des architectures distribuées pour le diagnostic et la supervision. M. O. Cordier a proposé dans une conférence au GDR MACS

(groupe S3) une classification de ces architectures. La première distinction à faire concerne l'utilisation ou non d'un *superviseur*. Celui-ci est un agent d'un type particulier, qui coordonne les tâches de différents autres agents. Dans le cas du diagnostic, cela revient à faire un diagnostic global par un agent spécialisé, à partir de résultats partiels obtenus par divers autres agents. La décision finale est centralisée au niveau du superviseur, qui synchronise et mémorise les activités des autres agents. On remarquera que c'est exactement la politique choisie dans MAGIC, où l'Agent de Décision Diagnostic joue ce rôle. L'Agent d'Aide à la Décision adopte lui un point de vue centralisé sur le procédé. La deuxième distinction concerne la connaissance du modèle du procédé, qui peut être centralisée (tous les agents connaissent le même modèle global de l'installation, même si les calculs partiels sont répartis entre les agents) ou répartie (chaque agent ne connaît qu'un sous-modèle partiel de l'installation et ignore complètement la représentation des autres sous-systèmes ; il n'utilise qu'une partie des mesures disponibles). Dans MAGIC, la connaissance du modèle est répartie entre les divers Agents de Diagnostic, qui ignorent complètement ce que font les autres agents dédiés à d'autres sous-systèmes. Par contre, le superviseur dispose lui d'un modèle global du procédé, sans lequel il ne pourrait établir ses tables de signature. On peut donc dire au vu de cette classification que MAGIC constitue une architecture de diagnostic et de supervision distribuée. En effet, car nous avons distribué les différentes tâches d'une procédure de diagnostic par exemple les tests de détection. En revanche, cette architecture reste centralisée au niveau de l'analyse diagnostique. Les communications entre agents, en particulier, sont assez figées, compte tenu de l'organisation multi-niveaux. Il serait intéressant d'étudier d'autres architectures pour le diagnostic et la supervision, qui seraient moins centralisées. Dans ce type d'architecture, la principale difficulté réside dans la coordination entre agents.

Références Bibliographiques

[Adrot, 2000] Adrot O., « Diagnostic à base de modèles incertains utilisant l'analyse par intervalles : L'approche bornante ». Doctorat de l'Institut National Polytechnique de Lorraine, France, 2000.

[Afnor, 1994] AFNOR, Maintenance - Concepts et définitions des activités de maintenance, Norme NF X 60-010, Association Française de Normalisation, 28 pages, 1994.

[Austin, 1962] Austin, J.L. « How To Do Things With Words ». Oxford University Press, 1962.

[Albert et Langle et al, 2004] Albert, M., T. Langle and H. Worn (2004). « Real-time Requirements in Diagnostic Systems ». IFAC MMM 2004, Nancy, France.

[Alcorta et al, 1997] Alcorta Garcia E., Frank P., « Deterministic nonlinear observer-based approaches to fault diagnosis : A survey », Control Engineering Practice, vol.5, n°5, p.663-670, 1997.

[Åström et al., 2001] Åström , K., P. Albertos, M. Blanke, A. Isidori, W. Shaufelberger and R. Sanz. Control of Complex Systems. London, Springer-Verlag, (2001).

[Atlas, Lacatusu et al., 2003] Atlas, E., C. Lacatusu, M. Capobianco, S. Gentil, M. Albert, T. Marcu and B. Köppen-Seliger (2003). Process Dependent Choice of Diagnostic Methods. IFAC-SAFEPROCESS 2003, Washington, D.C.

[Baujard, 1992] Olivier Baujard. « Conception d'un environnement de développement pour la résolution de problèmes : Apport de l'intelligence Artificielle et Application à la Vision ». Thèse de l'Université de Joseph Fourier Grenoble I, 1992.

[Beard, 1971] Beard R. V., « Failure accommodation in linear systems through self reorganization". These de doctorat, Massachusetts Institute of Technology (MIT), 1971.

[Brunet et al., 1990] Brunet M., Jaume D., Labarrere M., Rault A., Verge M., « Détection et diagnostic de pannes-Approche par modélisation», Traité des nouvelles technologies, série Diagnostic et Maintenance, Hermèse, Paris 1990.

[Borras et al, 1999] Borras D., Castilla M., Moreno N., Montano J. C., « Wavelet and neural structure: a new tool for diagnostic of power system disturbances". IEEE SDEMPED'99, International Symposium On Diagnostics for Electrical Machines, Power Electronics and Drives, pp. 375-380, Gijon, Spain, Sept. 1-3, 1999.

[Boissier, 2001] Olivier Boissier, « Systèmes Multi-Agents », Cours SMA-DEA-CCSA, SMA/ENS Mines Saint-Etienne, 2001

[Boutobza, 2003] Boutobza, A. (2003). « Génération automatique des tests de détection pour le diagnostic de systèmes physiques complexes ». Rapport de DEA, Laboratoire d'Automatique de Grenoble, Grenoble.

[Bischop, 1994] Bischop C. M., « Neural networks and their applications »; Revue science instrument, vol. 65, n°6 p.1803-1832,1994.

[Bullemer et Nimmo 1996] Bullemer P.T., Nimmo, I., « A Training Perspective on Abnormal Situation Management: Establishing an Enhanced Learning Environment. AICHE conference on Process Plant Safety, Etats-Unis, 1996.

[Busson et al., 1998] Busson F., Aïtouche A., Ould Bouamama B., Staroswiecki M., « Sensors Failure Detection in Steam Condensers". 3rd IFAC Workshop on On-Line Fault Detection and Supervision in Chemical Process Industries, France, 1998.

[CHEM-Consortium, 2000] CHEM-Consortium (2000). Advanced decision support system for chemical/petrochemical manufacturing processes: http://www.chem-dss.org.

[Chen et Patton, 1999] Chen J., Patton R. J., « Robust model based fault diagnosis for dynamic systems Kluwer » Academic Publishers, Amesterdam, 1999.

[Cohen et Levesque, 1995] Cohen, P.R. et H.J. Levesque. « Communicative actions for artificial agents ». In Proceedings of the International Conference on multi-agents Systems, AAAI Press, Juin 1995

[Chow et Willsky, 1984] Chow E. Y., Willsky A. S., « Analytical redundancy and the design of robust failure detection systems ». IEEE transaction on Automatic Control, Vol. 29, pp. 603-614, 1984.

[Combastel, 2000] Combastel Christophe « Méthodes d'Aide à la décision pour la détection et la localisation de défauts dans les Entraînements Electriques ». Thèse de doctorat de l'Institut National Polytechnique de Grenoble, 2000.

[Cordier et al., 2000] Cordier, M.-O., P. Dague, M. Dumas, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-massuyès. « A comparative analysis of AI and control theory approaches to model-based diagnosis". 14th European Conference on Artificial Intelligence, Berlin, Germany, (2000).

[Cordier et al., 2004] Cordier, M.-O., P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-massuyès. "Conflicts Versus Analytical Redundancy Relations: A comparative Analysis of the Model Based Diagnosis Approache From the Artificial Intelligence and Automatic Control Perspectives". Special Issue of the IEEE Transactions on Systems, Man, and Cybernetics, pp. 2163-2177, October 2004, volum 34, Number 5.

[Dvorak et Kuipers, 1991] Dvorak, D. and B. Kuipers (1991). "Process monitoring and diagnosis: A model-based approach." IEEE Expert 6(3): 67-74.

[Dague, 1997] Dague, P., Travé-Massuyès L. Guerrin F. «Le raisonnement qualitatif ». Sous la direction de Louise Travé-Massuyès, Philippe Dague, François Guerrin., 1997.

[Dague, 2001] Dague, P., «Théorie logique du diagnostic à base de modèles ». in: Diagnostic, Intelligence artificielle et reconnaissance de formes, B. Dubuisson, ed., 17-104, Hermès, Paris.

[Daly et al, 1979] Daly K. C., Gai E., Harrison J, V., "Generalised likelihood test for FDI in redundant sensor configuration". Journal of Guidance and Control, Vol. 2, N° 1,p 9-17, 1979.

[Dalton et al, 1999] Dalton T., Kremmer N., Frank, P. M., "Application of fuzzy logic based methods of residual evaluation to the three tank benchmark". European Control Conference (ECC), 2, 6-28, 1999.

[Davis, 1984] Davis R., Diagnostic Reasoning based on structure and behaviour, Artificial Intelligence, vol. 24, pp. 347-410, 1984.

[Davis et al, 1992] Davis R., et Hamscher W, « model-based reasoning : troubleshooting », Morgan Kaufmann Publication, San Mateo, 1992.

[Debouk et al, 2000] Debouk R., Lafortune S., and Teneketzis D., Coordination decentralized protocols for failure diagnosis of discrete event systems, Discrete Event Dynamic System: Theory and Applications, 10(33), 2000.

[Declerck and Staroswiecki, 1991] Declerck, P., et M. Staroswiecki (1991). "Characterization of the canonical components of a structural graph for fault detection in large scale industrial plants". European Control Conference, Grenoble, France.

[Deckert et al, 1997] Deckert J.C., Desai M.N., Deyst J.J., Willsky A.S., "F-8 DEBW sensor failure identification using analytic redundancy", IEEE Transactions on Automatic Control Vol. 22, N°5, p 795-803,. 1977.

[De Kleer et Williams., 1987] De Kleer, J., and B. C. Williams. "Diagnosing multiple faults". Artificial Intelligence, 32, pp. 97-130,1987.

[De Kleer et Williams., 1989] De Kleer, J., and B. C. Williams. "Diagnosing with behavioral modes". In Proc. Of the 11th International Joint Conference on Artificial Intelligence, IJCAI-89, Detroit, MI, USA, p. 1324-1330, 1989.

[De Kleer et al, 1992] De Kleer J., Mackworth A., and Reiter R.. "Characterizing diagnoses and systems. Artificial Intelligence », 56(2–3) : 197–222, 1992.

[Delouis, 1991] Delouis I. « Une nouvelle génération d'architectures pour le développement de systèmes à base de connaissances ». L.R.I Rapport de recherche . 1991.

[Demazeau et Muller, 1990] Yves Demazeau et Jean-Pierre MULLER. « Decentralized Artificial Intelligence » Elsevier Science Publishers B.V (North Holland) 1990.

[Desai et al, 1976] Desai M.N., Deckert J. C., Deyst J. J., Willsky A. S., Chow E. Y, "Dual-redundant sensor FDI techniques applied to the NASA F8C-DFBW aircraf"t. AIAA Guidance and Control conference, San Diego (California, USA), Vol. 1, p 502-513,1976.

[Desai et al, 1976] Desai M.N., Deckert J. C., Deyst J. J., "Dual sensor failure identification using analytic redundancy". Journal of Guidance and Control, Vol.2, N°3, p213-220, 1979.

[Dressler, 1994] Dressler O., "Model-based diagnosis on board: Magellan-MT inside", dans working notes of the 5th International Workshop on Principales of Diagnosis DX'94, New Paltz, NY, Etat-Unis, 1994.

[Dressler et al., 1994] Dressler O., Freitag H., "Prediction sharing across time and contexts", dans Working notes of the 8th International Workshop on Qualitative Reasonning QR'94, Nara, Japon, 1994.

[Dressler, 1995] Dressler O., "On-line diagnosis and monitoring of dynamic systems based on qualitative models and dependency-recording diagnosis", dans Working notes of the 9th International Workshop on Qualitative Reasoning QR'95, Amsterdam, Pays-Bas, 1995.

[Dubois et al, 1994] Dubois, D., Prade H., Terrier F,. « Ensembles floues et raisonnement automatisé », Logique Floue, OFTA – Masson, 1994.

[Dubuisson, 2001] Dubuisson B., (dir), « Automatique et statistiques pour de diagnostic, Hèrmès, Traité IC2, Paris, 2001.

[Evsukoff et al, 1998] Evsukoff A., Montmain J., Gentil S., " Causal model based supervising and training", dans IFAC Workshop on On-line Fault Detection and supervision in the Chemical Process Industries, Solaize, Lyon, France, IFP-IFAC,1998.

[Evsukoff, Montmain et Gentil 1997] Evsukoff A., Montmain J., Gentil S., Dynamic model based supervising and causal knowledge-based fault detection and isolation. 3rd IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (Safeprocess), pp. 699-704, Angleterre, 1997.

[Evsukoff, 1998] Evsukoff A., « Le raisonnement approché pour la surveillance de procédés ». Thèse de doctorat de l'Institut National Polytechnique de Grenoble, 1998.

[FIPA, 1997] Foundation for Intelligent Physical Agents. FIPA 97 Specification. Part 2, Agent Communication Language. http://www.fipa.org, 1997.

[Frank, 1990] Frank, P. M., "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results". Automatica, 26(3), pp. 459-471, 1990.

[Frank et Kiupel, 1993] Frank, P. M., Kiupel "Fuzzy supervision and application to lean production".Int.J.Systems Sci.,vol.24,no.10, pp 1935-1944, 1993.

[Frank, 1996] Frank, P. M., "Analytical and qualitative model-based fault diagnosis – a survey and somme new results". European Journal of Control, 2, 6-28, 1996.

[Frank et al, 1996] Frank, P. M., Kiupel N., "Residual evaluation for fault diagnosis using adaptative fuzzy thresholds and fuzzy inference".IFAC, 13th Triennial World Congress, San Francisco, USA, 1996.

[Frank et al , 2000] Frank, P. M., Ding, S. X., Köppen-Seliger, B. (2000). Current Developments in the Theory of FDI, SAFEPROCESS 2000, Budapest (Hungary), pp. 16-27

[Ferber, 1995] Ferber, J. 1995. "Les systèmes multi-agents. Vers une intelligence collective". InterEditions, Paris.

[Ferber, 2000] Jacques Ferber "Multi-agent systems: an introduction to Distributed Artificial Intelligence", Published in 2000 by Addison Wesley.

[Friedrich et al., 1990] Friedrich G., Nejdl W., MOMO : "Model-based diagnosis for everybody", dans Proc. Of the 6th IEEE Conference on AI Applications, Santa Barbara, CA, Etats-Unis, p. 206-213, 1990.

[Frisk, 2000] Frisk, E. (2000). "Residual Generator Design for Non-linear, Polynomial Systems - A Gröbner Basis Approach". IFAC Fault Detection, Supervision and Safety for Technical Processes, Budapest, Hungary.

[Garcia-Beltran, Lescecq et al, 2003] Garcia-Beltran, C., S. Lesecq, S. Gentil and R. Stuecher (2003). « Signal-Based And Causal-Based Diagnoses of A Hydraulic Looper and a main Drive of a Hot Rolling Mill". Workshop on Advanced Control and Diagnosis, University Duisburg - Essen, Germany, IAR-ICD Working Group.

[Garcia-Beltran, 2004] Garcia-Beltran Carlos, «outils pour l'aide à la supervision de procédés dans une architecture multiagent », Doctorat de l'Institut National Polytechnique de Grenoble, France, 2004.

[Gasser, 1992] Gasser L., "An overview of DAI. In Distributed Artificial Intelligence : Theory and Praxis" (L. Gasser and N. M. Avouris, eds.), pp.9-30. Kluwer Academic Publishers, Boston, 1992.

[Gentil et al, 2004] S. Gentil, J. Montmain and C. Combastel. «Combining FDI and AI Approaches Within Causal-Model-Based Diagnosis». IEEE Transactions on Systems, Man, and Cybernetics, Volume 34, number 5, pp. 2207-2221, October 2004,

[Gerald, 1990] Masini Gerald « Les langages à objets langages de classes, langages de frame, langages d'acteurs ». Inter-edition 1990.

[Gertler, 1988] Gertler, J. Survey of Model-Based failure detection and isolation in complex plants. IEEE Control Systems Magazine, 11, pp. 3-11,1988.

[Gertler et al., 1990] Gertler J., Singer D., "A new structural framework for parity equation-based failure detection and islolation". Automatica, Vol. 26, No.2, pp.381-388, 1990.

[Gertler, 1991] Gertler J. (1991). Analytical redundancy methods in fault detection and isolation – survey and synthesis. Proceeding of the IFAC Symposium on Fault Detection Supervision and Safety for Technical Process, Baden Baden, Germany, pp. 9-22.

[Gertler et al., 1993] Gertler J., and Monajemy R. (1993)."Generating directional residuals with dynamic parity equations". IFAC Symposium on Fault Detection Supervision and Safety for Technical Process, Sydney, Australia, pp. 507-512.

[Gertler, 1997] Gertler J., "Fault Detection and Isolation using Parity Relations". Control Engineering Practice, Vol. 5, No. 5, pp. 653-661, 1997.

[Gertler, 1998] Gertler J., "Fault Detection and diagnosis in engineering systems", Marcel Dekker Inc., 1998.

[Greiner, 1989] Greiner, R., B. A. Smith, and R. W. Wilkerson. "A correction to the algorithm in Reiter's theory of diagnosis". Artificial Intelligence, 41(1), pp. 79-88, 1989.

[Guernez Jean, 1998] Guernez Jean Caroline, « Surveillance de systèmes a modèles polynomiaux : génération de résidus et étude de sensibilité », Doctorat de l'université des sciences et Technologies de LILLE, France, 1998.

[Gasser et al, 1987] L. Gasser, Braganza, C., and Herman, N. (1987). "MACE : A flexible testbed for distributed ai research". In Distributed Artificial Intelligence, London, 1987

[Haton, 1991] Jean Paul Haton « Le raisonnement en intelligence artificielle » Inter-editions 1991. Paris.

[Hamscher et al., 1992] Hamscher W., Consol L., De Kleer J., (dir.), "Readings in Model-Based Diagnosis", Morgan Kaufmann, San Mateo, CA, Etats-Unis, 1992.

[Harbouche, 1993] A. Harbouche, « Conception d'un système de résolution de problèmes en IAD ». Thèse de magister, USTHB 1993.

[Herceau et Ferbert, 1991] Jean Herceau et Jacques Ferber « L'intelligence artificielle distribuée ». Recherche 233, Juin 1991, volume 22.

[Iffencker, 1992] C. Iffencker. « Un système multi-agent pour le support des activités de conception de produit ». Thèse de l'Université Paris VI . Décembre 1992. Rapport n°93/01.

[IEEE, 1988] IEEE Standard dictionary of Electrical and Electronics terms, 4th edition, F. Jay (Ed.), The Institute of Electrical and Electronics Engineers, New York, IEEE 1988.

[Isermann et Balle, 1997] Isermann, R., and P. Balle. "Trends in the application of model-based fault detection and diagnosis of technical processes". Control Engineering Practice, 5(5), pp. 709-719, 1997.

[Isermann et al., 2000] Isermann, R., and P. Balle., "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, 2000.

[Iwasaki et al., 1994] Iwasaki, Y., and H. A. Simo. "Causality and model abstraction". Artificial Intelligence, 67, pp. 143-194, (1994)

[Izadi-Zanamabadi et Staroswiecki, 2000] Izadi-Zanamabadi, R., Staroswiecki M.; "A Structural Analysis Method Formulation for Fault-Tolerant Control System Design". In 39th IEEE Conference on Decision and control, pages 4901- 490,. 2000.

[Jones, 1973] Jones H. L., "Failure detection in linear systems", These de doctorat, Massachusetts Institute of Technology (MIT), 1973.

[Kapur et Lakshma, 1992] Kapur, D., and Y. N. Lakshman (1992). "Elimination Methods: an Introduction. Symbolic and Numerical Computation for Artificial Intelligence", Academic Press.

[Khoualdi, 1994] K.Khoualdi. « Filtrage d'alarmes pour un système automatisé par une approche multi-agents ». Thèse de l'Université Paris VI. LAFORIA, 8 Novembre 1994.

[Köppen-Seliger et al, 2002] Köppen-Seliger Birgit, Steven X. Ding and Paul M. Frank, " MAGIC-IFATIS", IFAC 2002.

[Köppen-Seliger, Marcu et al. 2003] Birgit Köppen-Seliger1, Teodor Marcu, Michele Capobianco, Sylviane Gentil, Martin Albert, Siegfried Latzel, « MAGIC: An integrated approach for diagnostic data management and operator support », IFAC SAFEPROCESS 2003., Washington, D.C.

[Krysanderet al.,2002] Krysander M., Nyberg M., "Structural Analysis utilizing MSS Sets with Application to a Paper Plant". Thirteenth International Workshop on Principles of Diagnosis, Australie, 2002.

[Labrou et Finin, 1998] Y. Labrou et T. Finin. « Semantics for an Agent Communication Language ». Agent Theories, Architectures and Languages IV. M. Wooldridge, J. P. Muller and M. Tambe, Springer-Verlag.

[Labrou et al, 1999] Y. Labrou, T. Finin et Y. Peng - Agent Communication Languages : the Current Landscape, IEEE Intelligent Systems, pp.45-52, March/April 1999.

[Labrou et Fenin, 1997] Labrou, Y. et T. Finin. A Proposal for a new KQML Specification. Technical Report CS-97-03, 1997

[Labidi et Lejouad, 1993] Sofiane Labidi et Wided Lejouad «From distributed intelligence to multi-agent systems ». INRIA Sophia Antipolis. 1993.

[Lacatusu, Stücher et al, 2002] Lacatusu, C., R. Stücher and M. Capobianco (2002). MAGD001S02: General Specifications for Process Specification and Data Acquisition Agents. Duisburg, UNIDUI.

[Lackinger et Nejdl, 1993] Lackinger, F. and W. Nejdl (1993). « Diamon: A model-based troubleshooter based on qualitative reasoning." IEEE Expert 8(1): 33-40.

[Leitao et al, 2000] Leitao, P., and Restivo F. «A framework for distributed manufacturing applications », in advanced Summer Institute International Conference (ASI 2000), pp.75-80, Septembre 2000.

[Leonhardt et Ayoubi 1997] Leonhardt S., Ayoubi M., « Methodes of fault diagnosis », Control Engineering Practice, vol.5, n° 5, pp 683-693, may 1997.

[Lesecq et al, 2001] Lesecq S., Petropol S., Barreau A. « Asyunchronous motor parametric faults diagnosis using wavelet analysis », Conférence IEEE Sdemped 2001, Goriza , Italie.

[Lesecq et al, 2003] Lesecq S., Gentil S., Exel M, Garcia-Beltran C., « Diagnostic Tools for a multi-agent monitoring system » IMACS Multiconference. Computational Engineering in Systems Applications, CESA'03, 2003, 9-11 juillet, Lille, 2003.

[Leyval et Gentil] Leyval L., S. Gentil, Feray-Beaumont. «Model based causal reasoning for process supervision", Automatica, 30 (8), pp. 1295-1306,1994.

[Luenberger, 1971] Luenberger D. G., « An introduction to observers », IEEE transactions on Automatic control, 16(6), pp. 596-602, 1971.

[Massoumnia, 1988] Massoumnia, M. M., and W. E. Van Der Velde (1988). Generating parity relations for detecting and identifying control system component failures. Journal of guidance, control and dynamics, 11(1), pp. 60-65.

[Mandiau, 2000(a)] Mandiau, R. (2000a). HDR : Modélisation et Evaluation d'Organisations Multi-Agents. LEMIH, L'universite de Valenciennes et du Hainaut-Cambresis.

[Mandiau, 2000(b)] Mandiau, R. (2000b). Modélisation et Evaluation d'Organisations Multi-Agents. Valenciennes, L'universite de Valenciennes et du Hainaut-Cambresis: 128.

[Malik et al., 1996] Malik A., Struss P., « Diagnossis of dynamic systems does not necessarily require simulation", dans working notes of the 7th International Workshop on Principles of Diagnosis DX'96, Val Morin, Canada, 1996.

[Martin et al, 2002] Martin Albert, T. langle, and H. Worn, "Development tool for distributed monitoring and diagnosis systems', in proceedings of the 13th International Workshop on principales of Diagnosis (DX-2002), simmering, Austria, (May 2002).

[Mc Carthy, 1986] Mc Carthy, J. « Applications of circumcription to formalizing common-sense knowledge ». Artificial Intelligence, 28, pp. 89-116, 1986.

[Milne, 1987] Milne R., Strategies for diagnosis. IEEE Transactions on Systems, Man and Cybernetics, Vol.17, N°3, p 333-339, 1987.

[Montmain et Gentil, 2000] Montmain J., S. Gentil., « Dynamic causal model diagnostic reasoning for on-line technical process supervision », Automatica 36, 1137-1152, 2000.

[Montmain, 1997] Montmain, J. (1997). « From Diapason Research Program to its Industrial application in nuclear fuel reprocessing ». IFAC-SAFEPROCESS '97, Hull, United Kingdom.

[Montmitonnet, 2002] Montmitonnet P., L. (2002). « Objectifs et modélisation ». Les Techniques de l'Ingénieur M 3065: 1-12.

[Narendra et al; 1990] Narendra K.S., Parthasarathy K., "identification and control of dynamical systems using neural networks », IEEE Transactions on Neural Networks, vol. 1, n°1, p. 4-27, 1990.

[Nyberg, 2001] Nyberg M., « A general framework for fault diagnosis based on statistical hypothesis testing », dans Working notes of the 12th International Workshop on Principles of Diagnosis DX'01, Sansicario, Italie, 2001.

[Ould Bouamama e tal., 2001] Ould Bouamama B., Staroswiecki M., Litwak R., « Surveillance d'un générateur de vapeur, in: Automatique et statistiques pour le diagnostic », B. Dubuisson, ed., 166-199, Hermès, Paris, décembre 2001.

[Oyeleye, Finch et al, 1990] Oyeleye, O. O., F. E. Finch and M. A. Kramer (1990). « Qualitative modeling and fault diagnosis of dynamic processes by MIDAS». Chemical Engineering Comminications 96: 205-228.

[Panati et al., 1998] Panati A., Theseider Dupré D. « State-based vs simulation-based diagnosis of dynamique systems », dans working notes of the 9th International Workshop on Principles of Diagnosis DX'98, Cape Code, MA, Etats-Unis, 1998.

[Patton et al., 1989] Patton, R.J, Frank P., and Clark R. (Eds). "Fault diagnosis in dynamic systems", Prentice Hall, 1989.

[Patton et al, 1991] Patton, R.J, and J. Chen, "A review of parity space approaches to fault diagnosis", in IFAC Safeprocess symposium, Baden-Baden, 1991.

[Patton, 1997] Patton, R.J, « Fault tolerant control: the 1997 situation", Proceedings of the IFAC Symposium on Fault Detection Supervision and Safety for technical process, Hull (United Kingdom), Vol. 2., p 1033-1055, August 26-27, 1997.

[Patton et Chen., 1997] Patton R., Chen J., « Observer based fault detection and isolation: robustness and applications". Control Engineering Practice, Vol. 5, No. 5, pp. 671-682, 1997.

[Petropol, 2001] Petropol S. « Ondelettes et diagnostic : application aux défauts dielectriques et electriques des machines tournante ». Doctorat de l'Institut National Polytechnique de Grenoble, France, 21 sept. 2001.

[Poncolé et al, 2002] Y. Poncolé, M.-O. Cordier, L. Rozé, « Une stratégie efficace pour une approche décentralisée du diagnostic de systèmes complexes », in : RFIA'02 (Congrès francophone Afrif-Afia de reconnaissance des formes et d'intelligence artificielle), p. 259–268, Angers, France, 2002.

[Poncolé, 2002] Poncolé Yannick, « Diagnostic décentralisé de systèmes à évènements discrets : application aux réseaux de télécommunications » Thèse de L'université de Rennes 1, 28 juin, 2002.

[Pénalva, Coudouneau et al, 1993] Pénalva, J. M., L. Coudouneau, L. Leyval and J. Montmain (1993). "DIAPASON: a supervision support system." IEEE Expert: Intelligent Systems and their Applications 8(5): 57-65.

[Ploix, 1998] Ploix, S., « Diagnostic des systèmes incertains: l'approche bornante" Thèse de l'institut National Polytechnique de Loraine, 23 décembre, 1998.

[Ploix et Follot, 2001] Ploix, S., et C. Follot (2001). « Fault diagnosis reasoning for set-membership approaches and application". CCA/ISIC'01, Mexico City, Mexico.

[Ploix et al, 2005] Ploix S., Désinde M., Touaf S. « Automatic Design of Detection Tests in Complex Dynamic Systems », 16th World Congress, IFAC, July 4-8- 2005, Praha, Czech Republic,

[Poole, 1989] Poole D., « Normality and faults in logic-based diagnosis", dans Proc. Of the 11th International Joint Conference on Artificial Intelligence IJCAI'89, Detroit, MI, Etats-Unis, p. 1304-1310, 1989.

[Potter et al, 1977] Potter J. E., Suman M. C., « Thresholdless redundancy management with arrays of skewed instruments". Agardograph 224, Integrity in Electronic flight control systems, p 15.1-15.24, 1977.

[Provan, 2002] Provan G. « A model-based diagnosis framework for distributed systems", in proceedings of the 13th international workshop on principales of diagnosis (DX-2002), Semmering, Austria, 2002.

[Reiter, 1987] Reiter R., « A theory of diagnosis from first principles". Artificial Intelligence, 32(1) : 57–96, 1987.

[Rosé, 1997] L. Rozé, « Supervision of telecommunication network : a diagnoser approach », in : Proceedings of the International Workshop on Principles of Diagnosis (DX'97), p. 103–111, Mont St Michel, France, 1997.

[Russell, 1997] Russell, S.J. « Rationality and intelligence". Artificial Intelligence, Vol. 94, 1997. p.57-77.

[Sampath et al, 1998] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, D. Teneketzis, « Active Diagnosis of Discrete-Event Systems », IEEE Transactions on Automatic Control 43, 7, 1998, p. 908–929.

[Staroswiecki, 2001] Staroswiecki M., « Redondance analytique, in : Automatique et statistiques pour le diagnostic », B. Dubuisson, ed., 43-68, Hermès, Paris. Décembre 2001.

[Steyer et al, 2001] Steyer J.P., Génovési A., Harmand J., « Outils d'aide au diagnostic et détection de pannes » in: Diagnostic, Intelligence artificielle et reconnaissance de formes, D. Dochain, ed., 215-244, Hermès, Paris, 2001

[Struss et Dressler, 1989] Struss P., Dressler O., « Physical negation : Integrating fault models into the general diagnostic engine » in Proc. Of the 11th International Joint Conference on Artificial Intelligence, IJCAI-89, Detroit, MI, USA, p. 1318-1323 (1989). (repris dans Readings in Model-Based Diagnosis, Hamscher W., Console L., de Kleer (dir.) Morgan kaufmann, p. 158, 1992).

[Struss, 1991] Struss P., « what's in SD ? » Towards a theory of modeling for diagnosis DX'91, Milan, Italie, p. 41-51, 1991.

[Struss, 1997] Struss P., "Fundamentals of Model-Based Diagnosis of Dynamic Systems », dans Proc. Of the 15th International Joint Conference on Artificial Intelligence IJCAI'97, Nagoya, Japon, p.480-485, 1997.

[Struss et al, 1997] Struss P., Sachenbacher M., Dummert F., « Diagnosing a dynamique system with (almost) no observations » dans working notes of the 11th International Workshop on qualitative reasoning QR'97, Cortona, Italie, 1997.

[Struss et al, 2002] Struss. P, Rehfus B., Brignolo R., Cascio F., Console L., Dague P., Dubois P., Dressler O., and Millet D, 2002. Model-based Tools for the Integration of Design and Diagnosis into a Common Process » –A Project Report.

[Stuecher et al, 2004] Stuecher, R. and A. Metzul (2004). « Simulations and Measurements of hot rolling mill components for use in diagnostic benchmark tests ». IFAC MMM 2004, Nancy, France.

[Su et al, 2000] Su R., and Wonham W., "Decentralized fault diagnosis or discrete-event systems", in Proc. 2000 CISS, Mars 2000.

[Shoham, 1993] Shoham,Y. "Agent-oriented programming". Artificial Intelligence, Vol. 60, 1993. p.51-92

[Thomson, 1999] Thomson W.T., "A review of on-line condition monitoring techniques for three-phase squirrel-cage induction motors-Past, present and future". Proceedings of IEEE SDEMPED'99, pp3-18, Gijon, Spain, 1999.

[Touaf et al, 2003 (a)] Touaf S., Ploix S., Flaus J-M., « Software architecture for diagnosing physical systems » , IEEE American Control Conference, ACC' 03, 2003, 4-6 juin, Denver, Colorado (USA).

[Touaf et al, 2003 (b)] Touaf S., Ploix S., Flaus J.M., «A Logical Diagnostic Method For Complex Dynamic Systems », 5ème Congrès International Pluridisciplinaire Qualité et Sûreté de Fonctionnement, Qualita, 2003, 18-20 mars, Nancy (France).,

[Touaf et al, 2003 (c)] Touaf S., Ploix S., Flaus J-M., « Diagnosis for large scale distributed industrial plants application to an hydraulic looper », 11th IFAC Symposium on Automation in Mining, Mineral and Metal processing, MMM04, 8-10 septembre 2004, Nancy, France.

[Touaf et al, 2004 (d)] Touaf S., Ploix S.,« Soundly managing uncertain decisions in diagnostic analysis », 15th International Workshop on Principles of Diagnosis (DX'04), Toulouse, 23-25 Juin, 2004.

[Travé-Massuyes et Milne, 1996] Trave-Massuyes, L. and R. Milne « Gas turbine condition monitoring using qualitative model-based diagnosis » IEEE Expert 12(3): 21-31, 1996.

[Travé-Massuyès et al., 1997] Travé-Massuyès L., Dague P., Guerrin F., « Le raisonnement qualitatif pour les sciences de l'ingénieur », Hermès, 1997

[Wakamiya et Nitta, 2000] Wakamiya, Y. and I. Nitta « A quality Control and Diagnostic System for Hot-Rolling Mills » Mitsubishi Electric ADVANCE 92: 30-33, December 2000

[Willsky, 1976] Willsky A S., « A survey of design methods for failure detection in dynamic systems ». Automatica, Vol. 12, pp. 601-611, 1976

[Wooldrige et Jennings, 1995] Wooldridge, M. et N. R. Jennings. « Agent theories, architectures, and languages ». Dans Wooldridge, Jennings (ed), Intelligent Agents, Springer Verlag, 1995. p.1-22.

[Zwingelstein, 1995] Zwingelstein G., « Diagnostic des défaillances: théorie et pratique pour les systèmes industriels », Traité des nouvelles technologies, série Diagnostic et maintenance, Hermès, Paris, 1995

[Zadeh, 1965] Zadeh, L., « fuzzy sets », Information and Control, vol.8, p. 338 –353,1965.