



HAL
open science

Outils pour l'aide à la supervision de procédés dans une architecture multiagent

Carlos Daniel Garcia-Beltran

► **To cite this version:**

Carlos Daniel Garcia-Beltran. Outils pour l'aide à la supervision de procédés dans une architecture multiagent. Automatique / Robotique. Institut National Polytechnique de Grenoble - INPG, 2004. Français. NNT: . tel-00169892

HAL Id: tel-00169892

<https://theses.hal.science/tel-00169892>

Submitted on 5 Sep 2007

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

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

À Verónica

«Ton devoir réel est de sauver ton rêve».

Amadeo Modigliani (1884-1920)

Remerciements

Cette petite section est la seule qui enchaîne aux critiques scientifiques, elle n'est pas même référencée dans la table de matières, je profite de cette liberté pour remercier à toutes les personnes qui ont contribué, de façon directe ou indirecte, dans la réalisation de cette thèse.

Le travail présenté dans ce mémoire a été réalisé au sein du Laboratoire d'Automatique de Grenoble.

J'adresse tout d'abord mes remerciements à tous les membres de jury pour l'honneur qu'ils m'ont fait de participer à l'examen de ma thèse. J'exprime toute ma gratitude à Monsieur Alain BARRAUD, Professeur à l'Institut National Polytechnique de Grenoble, et directeur du LAG, pour avoir accepté de présider mon jury. Je tiens à remercier également aux Professeurs Jean Philippe CASSAR, de la Université Polytechnique de Lille et Professeur Patrick MILLOT, du LAMIH - Université de Valenciennes, rapporteurs de ce mémoire pour leurs importantes remarques et suggestions. Je remercie aussi les examinateurs du jury, Professeur Dominique SAUTER, Professeur de l'Université de Nancy au CRAN, et Monsieur Jacky MONTMAIN, ingénieur expert au sein du Commissariat de l'Energie Atomique.

Je remercie très particulièrement Madame Sylviane GENTIL, Professeur à l'Institut National Polytechnique de Grenoble pour son encadrement scientifique, pour tous ses conseils, son soutien inestimable, son amitié et son partage de sa France, toute au long de mon stage.

Je remercie également à tous les «Magiciens» du projet, en commençant par ceux du laboratoire, Suzanne Lesecq, Stéphane Ploix, Ioanna Faragasan, Pascal Bellemain, Samir Touaf et Samir Taleb ; je remercie très spécialement Monsieur Matija EXEL, avec qui j'apprends beaucoup sur la « vraie » informatique, toute au long du développement du projet et à qui je considère un deuxième directeur. Je remercie aussi les Magiciens d'ailleurs : Professeur Birgitt Kopen-Silinger, Professeur Tehodor Marcu, Cyprian Lacatusu et Arkan Atlas de l'Université de Duisburg ; Monsieur Martin Albert de l'Université de Karlsruhe, Monsieur Reininhan Stuecher de SMS-DEMAG, Monsieur Birgenti et Luca Fogar de SATE. L'expérience d'avoir participé dans un projet comme MAGIC, avec un groupe de travail assez hétérogène culturellement, a enrichi énormément ma formation et ma vie.

Ma gratitude aussi pour l'ensemble du personnel administratif et technique du LAG (Marie-Rose, Marie-Thérèse, Patricia, Virginie, Philippe, Daniel, etc.) qui m'a permis d'accomplir mon travail dans un environnement très agréable.

Ce travail a bénéficié de l'appui financier du programme mexicaine ANUIES-SUPERA, dans le cadre d'un projet de collaboration franco-mexicaine ECOS NORD – ANUIES ; et aussi du PROMEP-COSNET, Je tiens à remercier ces organismes pour m'avoir donné l'opportunité de vivre cette expérience.

Je remercie également à tous les amis et familiers, pour les encouragements.

Finalement, je n'oublierais jamais de remercier à ma petite famille: ma chère Verónica et mes deux petites fleurs Daniela et Ximena, qui ont supporté mes périodes de découragement, mes nombreux retards et mes absences, physiques et d'esprit. Je vous dédie cette thèse.

TABLE DE MATIERES.

GLOSSAIRE	I
LISTE DE FIGURES:	II
CHAPITRE 1 INTRODUCTION GENERALE.....	1
1.1 CADRE DE DEVELOPPEMENT DE LA THESE.....	1
1.2 PROBLEME INITIAL	3
1.3 OBJECTIFS DE LA THESE	4
1.4 ORGANISATION DU DOCUMENT	5
CHAPITRE 2 LA SUPERVISION AVANCEE DES PROCEDES	7
2.1 INTRODUCTION.....	7
2.2 LA SURETE DE FONCTIONNEMENT DES SYSTEMES INDUSTRIELS.....	8
2.2.1 <i>Les attributs des systèmes sûrs de fonctionnement.</i>	8
2.2.2 <i>Les risques qui menacent la sûreté de fonctionnement.</i>	9
2.2.3 <i>Les moyens pour atteindre la sûreté de fonctionnement.</i>	10
2.3 LES SYSTEMES D'AIDE A LA DECISION DE L'OPERATEUR POUR LA SUPERVISION: AIDE A L'OPERATION ET LA MAINTENANCE.....	15
2.3.1 <i>Systèmes d'aide à la décision.</i>	16
2.3.2 <i>Diagnostic pour la surveillance : détection, localisation et identification de défauts ...</i>	19
2.3.3 <i>Aide à la décision pour la conduite du système face à des fonctionnements anormaux.</i>	25
2.3.4 <i>Aide à la décision pour la maintenance de procédés.</i>	27
2.3.5 <i>Le pronostic.</i>	29
2.3.6 <i>Les interfaces graphiques.</i>	30
2.4 L'IMPLEMENTATION DES SYSTEMES D'AIDE A LA DECISION.....	31
2.4.1 <i>La modularité vue comme la répartition d'expertise dans un système de supervision avancé.</i>	31
2.5 CONCLUSIONS.....	33
CHAPITRE 3 LE SYSTEME MAGIC.....	35
3.1 INTRODUCTION.....	35
3.2 LES FONCTIONNALITES GENERALES DE MAGIC	35
3.3 L'ARCHITECTURE GENERALE DE MAGIC	37
3.3.1 <i>L'approche multi-niveaux de MAGIC.</i>	37
3.3.2 <i>Les deux phases de fonctionnement de MAGIC.</i>	41
3.3.2.1 <i>Les modes de configuration et de personnalisation.</i>	41
3.3.2.2 <i>Le mode d'opération continue.</i>	43
3.4 ARCHITECTURE INTERNE DES AGENTS.....	45
3.5 LES MECANISMES DE COMMUNICATION DANS MAGIC.....	46
3.5.1 <i>Communication via le langage de communication d'agent FIPA.</i>	47
3.5.2 <i>Communication via notification d'événements et fournisseur de données.</i>	48
3.5.3 <i>Communication intra-agent.</i>	49
3.6 L'ONTOLOGIE DE MAGIC.....	50
3.6.1 <i>Mise en application de l'ontologie.</i>	52
3.7 CONCLUSION.....	52
CHAPITRE 4 LES AGENTS DE DIAGNOSTIC : LES CAS DU CBDA.....	55
4.1 INTRODUCTION.....	55
4.2 FONCTIONNALITES DES AGENTS DE DIAGNOSTIC.....	56
4.2.1 <i>L'interaction des agents de diagnostic avec les autres agents</i>	57
4.2.2 <i>Principes de communication des agents de diagnostic.</i>	57
4.2.3 <i>L'évaluation des tests de détection pour la génération de symptômes.</i>	60
4.3 GENERATION DES SYMPTOMES A PARTIR DU MODELE CAUSAL.....	61
4.3.1 <i>Algorithme général d'évaluation du symptôme (cf. définition § 4.2.3).</i>	63
4.3.2 <i>Caractérisation du résultat du test.</i>	64
4.3.3 <i>Simulation causale.</i>	64
4.3.4 <i>La validité du modèle.</i>	70

4.3.5	<i>Génération des résidus.</i>	73
4.3.6	<i>La validité de la décision.</i>	74
4.3.7	<i>Les propriétés de localisation de défauts des tests globaux et locaux.</i>	75
4.4	CONFIGURATION ET PERSONNALISATION DE L'AGENT DE DIAGNOSTIC.	75
4.4.1	<i>Construction des modèles causaux.</i>	76
4.4.2	<i>L'éditeur des graphes causaux.</i>	77
4.4.3	<i>Identification de fonctions de transfert en utilisant le GPMAS.</i>	80
4.4.4	<i>Obtention des tests de détection à partir du graphe causal.</i>	81
4.5	EXEMPLES ACADEMIQUES.	81
4.5.1	<i>Description du système de bacs.</i>	82
4.5.2	<i>Tests de détection implémentés dans le CBDA.</i>	83
4.6	CONCLUSION.	92
CHAPITRE 5 L'AGENT D'AIDE A LA DECISION		95
5.1	INTRODUCTION.	95
5.2	ARCHITECTURE INTERNE DE L'AGENT D'AIDE A LA DECISION.	96
5.3	FONCTIONNALITES DE L'AGENT D'AIDE A LA DECISION.	97
5.4	PRONOSTIC.	98
5.4.1	<i>Les réseaux neuronaux pour la prédiction de séries temporelles.</i>	98
5.4.2	<i>L'analyse de tendances pour la prédiction de séries de temporelles.</i>	102
5.4.3	<i>La prédiction de la propagation de défaut à partir des résultats de diagnostic.</i>	107
5.5	OUTILS POUR L'AIDE A LA CONDUITE FACE AUX DEFAUTS.	110
5.5.1	<i>Les moyens d'action.</i>	111
5.5.2	<i>Recherche en ligne de conseils d'action.</i>	113
5.6	OUTILS POUR L'AIDE A LA PRISE DE DECISION DANS LES POLITIQUES DE MAINTENANCE	114
5.7	CONFIGURATION DE L'AGENT D'AIDE A LA DECISION.	115
5.8	EXEMPLE ACADEMIQUE.	115
5.9	CONCLUSIONS.	118
CHAPITRE 6 TESTS SUR LE TENDEUR HYDRAULIQUE DE BANDE		121
6.1	INTRODUCTION.	121
6.2	LE PRINCIPE DU LAMINAGE A CHAUD.	122
6.3	DESCRIPTION DE LE TENDEUR HYDRAULIQUE DE BANDE ET SON ROLE DANS LE LAMINOIR. ...	122
6.4	LES DIFFERENTS TESTS D'ACQUISITION EFFECTUES SUR LE TENDEUR HYDRAULIQUE DE BANDE EN VUE DU DIAGNOSTIC.	123
6.5	DETECTION DE DEFAUTS DU TENDEUR HYDRAULIQUE DE BANDE.	124
6.5.1	<i>Détection à partir de données simulées.</i>	124
6.5.2	<i>Test de détection en utilisant des données réelles.</i>	128
6.6	CONCLUSION.	131
CHAPITRE 7 CONCLUSIONS ET PERSPECTIVES		133
7.1	DISSEMINATION	135
7.2	PERSPECTIVES	136
BIBLIOGRAPHIE		139

Glossaire:

MAGIC	Système Multiagents pour l'Acquisition de données, le diagnostic et la gestion de procédés complexes
PSA	Agent de spécification du procédé
GPMAS	Serveur d'algorithmes mathématiques génériques
DAA	Agent d'acquisition de données
DA	Agent de Diagnostic
DDA	Agent de décision de diagnostic
DSA	Agent d'aide à la supervision.
OIA	Agent d'interface pour l'opérateur
MSE	Ingénieur de support de MAGIC
EUSE	Ingénieur utilisateur final du système
ACL	Langage de communication de agents
FIPA	La Fondation pour les Agents Physiques
CORBA	Architecture commun d'un bus de objets répartis
SCADA	Systèmes Numériques de Contrôle Commande

Liste de Figures:

Figure 1-1. Incidence de notre travail dans l'architecture de MAGIC.....	5
Figure 2-1. L'arbre de sûreté de fonctionnement	8
Figure 2-2. Dégradation et défaillance	10
Figure 2-3. Procédure intégrée pour la conception de systèmes sûrs de fonctionnement.....	12
Figure 2-4. Modèle de prise de décisions de Rasmussen révisé par Hoc.....	17
Figure 2-5. Coopération verticale.....	18
Figure 2-6. Positionnement des différentes méthodes dans le contexte de la supervision.....	19
Figure 2-7. Classification des méthodes de détection de défauts.....	20
Figure 2-8. Illustration du comportement du système et du comportement de référence.....	23
Figure 2-9. Espace de résidus.....	24
Figure 2-10. La gamme d'applicabilité des algorithmes de prédiction.....	30
Figure 2-11. Système multi-agents vu selon différents niveaux de détail.	32
Figure 3-1. Cas d'utilisation de MAGIC.....	36
Figure 3-2. L'architecture générale de MAGIC.....	38
Figure 3-3. Interface OIA.....	40
Figure 3-4. Interface GUIA.....	41
Figure 3-5. Agents utilisés pendant l'étape de configuration et personnalisation.	43
Figure 3-6. Les agents qui participent au mode de fonctionnement continu.	45
Figure 3-7. Architecture générale d'un agent de diagnostic.	46
Figure 3-8. La position de l'intergiciel dans les applications réparties.....	46
Figure 3-9. Abonnement des agents aux canaux d'événement.	48
Figure 3-10. Communication via la notification d'événements et le fournisseur de données.	49
Figure 3-11. Une partie des éléments de l'ontologie du système MAGIC.	50
Figure 3-12. MAGIC comme système de support en la prise de décisions.	53
Figure 4-1. Les différents cas d'utilisation des DAS.....	56
Figure 4-2. Architecture générale d'un agent de diagnostic.	57
Figure 4-3. Cas d'utilisation de l'agent de diagnostic.....	58
Figure 4-4. Diagramme de séquences d'un processus d'évaluation des tests de détection.	59
Figure 4-5. Causalité: une relation d'ordre partiel.	62
Figure 4-6. Un sous-graphe élémentaire.	63
Figure 4-7. Validité de la décision.	74
Figure 4-8. Principe de localisation de défauts en utilisant la causalité.....	75
Figure 4-8. Vues globales de l'interface graphique.	79
Figure 4-9. Utilisation de graphes causaux dans le cadre de MAGIC.....	80
Figure 4-10. Vues globales de l'interface graphique du GPMAS.....	80
Figure 4-11. Le système complet des bacs.....	82
Figure 4-12. La configuration de deux Bacs en cascade.....	83
Figure 4-13. Graphe causal associé à la configuration de deux bacs en cascade.....	85
Figure 4-14. Fuite graduellement croissante du Bac 1.....	87
Figure 4-15. Scénario 1: Simulation d'une fuite progressive dans le Bac 1.	88
Figure 4-16. Fuite brusque du Bac 1.....	89
Figure 4-17. Scénario 2: Simulation d'une fuite brusque du Bac 1.	90
Figure 4-18. Fuite graduellement croissante du Bac 2.....	91
Figure 4-19: Scénario 3: Fuite progressive dans le Bac 2.....	92
Figure 5-1. Structure interne de l'agent d'aide à la décision	97
Figure 5-2. Les cas d'utilisation du DSA.....	98

Figure 5-3 L' architecture de l' ANFIS.....	99
Figure 5-4. Fonction d'appartenance du type cloche.	99
Figure 5-5. Prédiction multi pas.....	102
Figure 5-6. Segmentation de données.	104
Figure 5-7. Classification en formes temporelles.....	105
Figure 5-8. Utilisation de l'analyse de tendances pour la prédiction.	108
Figure 5-9. Possibilités de la propagation des effets du défaut	109
Figure 5-10. Exemple d'un graphe avec un cycle.	113
Figure 5-11. a) Contradiction irréductible, b) contradiction réductible.	114
Figure 5-12. Prédiction des variables après une fuite progressive du Bac 1.....	116
Figure 5-13. Prédiction des variables après une fuite brusque du Bac 1.	117
Figure 5-14. Prédiction des variables après une fuite progressive du Bac 2.....	118
Figure 6-1. Procédé de laminage au chaud.	122
Figure 6-2: Schéma mécanique et hydraulique du tendeur hydraulique de bande.....	123
Figure 6-3. Graphe causal pour le système avec capteur de force.	125
Figure 6-4. Test de réponse à l'échelon de la consigne de force	126
Figure 6-5. Test de position du tendeur hydraulique.	129
Figure 6-6. Test de position –comportement normal.	130
Figure 6-7. Test de position –comportement en défaut.....	131
Figure 6-8. Modèles pour la décision et l'évaluation de la validité du modèle.	131

Chapitre 1

Introduction générale

1.1 Cadre de développement de la thèse.

Dans les dernières décennies, la communauté internationale d'automatique a commencé à porter un intérêt de recherche nouveau vers les niveaux supérieurs de contrôle, que sont la supervision, la coordination de boucles et la planification (scheduling). 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-BRIDGE¹, IAR, ASM², COSY, OSA³. 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 and Kuipers 1991], MIDAS [Oyeleye, Finch et al. 1990], DIAPASON et SALOMON [Pénalva, Coudouneau et al. 1993; Montmain 1997], TIGER [Trave-Massuyes and Milne 1996], DIAMON [Lackinger and Nejdil 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 de recherche et développement dans le domaine de la supervision. Dans ce projet européen interviennent cinq participants : l'université de Duisburg, l'université de

¹ MONET: The European Network of Excellence in Model- based Systems and Qualitative Reasoning : <http://monet.aber.ac.uk:8080/monet/index.html#barcelona>

² Abnormal Situation Management Consortium: <http://www.asnconsortium.com>.

³ OSA-CBM : Open System Architecture for Condition Based Maintenance: <http://www.osacbm.org/>

Karlsruhe, le Laboratoire d'Automatique de Grenoble ; et deux entreprises privées : SATE 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 pour l'industrie métallurgique.

L'architecture de MAGIC est fondée sur le concept de système multi-agents pour le diagnostic distribué. Le développement du projet est divisé en sept Workpackages (WP) :

- **WP1 : Gestion et coordination de projet** : L'objectif de cette tâche est d'administrer le projet MAGIC et coordonner son avancement.
- **WP2 : Infrastructure de communication et agent GUI** : le premier objectif de WP2 est de concevoir et de développer l'infrastructure complète de communication qui peut accueillir les agents distribués et leur communication et de développer un langage de communication d'agent qui est conçu en fonction des besoins spécifiques. Le deuxième objectif de WP2 est de développer un agent appelé dans MAGIC « agent GUI » (Graphical User Interface) avec lequel l'architecture logicielle peut être administrée.
- **WP3 : Outils de développement des agents** : l'objectif de ce workpackage est de produire des enveloppes pour les différents agents du système MAGIC.
- **WP4 : Outils de diagnostic** : l'objectif de WP4 est de développer des algorithmes pour la détection des défauts (génération de symptômes) et leur intégration dans des Agents de Diagnostic. Ce WP inclut aussi le développement d'un serveur d'application générale de routines mathématiques (GPMAS) pour l'identification de modèles.
- **WP5 : Agents de pré-diagnostic** : le but de ce WP est de construire les agents de pré-diagnostic, c'est à dire l'Agent de Spécification du Procédé, responsable de la configuration et personnalisation des agents de diagnostic et d'aide à la décision, et des Agents d'Acquisition de Données (prétraitement des données utilisées pour le diagnostic).
- **WP6 : Agents de décision de diagnostic et d'aide à l'opérateur** : le workpackage WP6 a les objectifs principaux suivants:
 - proposer une décision de diagnostic finale (composants défectueux) à partir des symptômes élaborés par les agents de diagnostic impliqués dans la génération de symptômes, grâce à un Agent de Décision de Diagnostic ;
 - aider les opérateurs en les conseillant sur les décisions de conduite et de maintenance, grâce à un Agent d'Aide à la Décision ;
 - afficher l'information de diagnostic et les conseils aux opérateurs de façon claire, grâce à un Agent d'Interface Opérateur ;
 - vérifier l'intégration de tous les composants de MAGIC.
- **WP7 : Application à un laminoir** : Le but principal de WP7 est d'installer une plateforme, basée sur le système MAGIC, dans le cadre d'une démonstration dans l'industrie métallurgique.
- **WP8 : Diffusion** : l'objectif de WP8 est la définition, l'évaluation et l'exécution des stratégies de vente pour l'exploitation des résultats du projet. Le but est d'identifier des

secteurs de marché possibles pour les produits, les services et les processus qui peuvent dériver des résultats directs et indirects du projet MAGIC.

En particulier, le Laboratoire d'Automatique de Grenoble est responsable de deux agents de diagnostic, situés dans WP4, et de la totalité de WP6. Vu la nature du projet, les activités développées dans ces WPs ont été coordonnées avec les autres WPs.

Un des principaux avantages de travailler sur ce type de projet est la possibilité intéressante 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 WP. Il est clair qu'arriver jusqu'à une 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, à cause du fait qu'elles restaient hors des objectifs initiaux du projet.

1.2 Problème initial

La demande croissante en qualité, sécurité, disponibilité et optimisation des coûts dans les procédés industriels rend nécessaire l'utilisation de Systèmes de Supervision Avancées dans ces procédés. Ces systèmes constituent un niveau supérieur aux systèmes de contrôle et utilisent techniques et méthodes spécifiques. L'objectif final de la Supervision de procédés est de parvenir à systématiser la tolérance aux défauts ; pour cela une partie importante des tâches de Supervision se centre sur la détection et le diagnostic de différentes situations de défauts qui peuvent affecter le procédé. Une autre partie importante est l'aide à l'opérateur pour la prise de décisions pour diminuer les effets des défauts sur le fonctionnement du processus.

Toutes ces tâches doivent faire usage de la connaissance disponible sur le processus (la structure, des équations statiques et dynamiques et leurs paramètres, la connaissance heuristique, etc.) et des données qui proviennent sous forme de mesures. La différence de la provenance de l'information entraîne une série de problèmes qui compliquent son utilisation.

La supervision d'un procédé correspond à deux catégories de situations : le fonctionnement normal et le fonctionnement dégradé ; elle met en oeuvre des procédures très diverses, fondées sur de nombreuses mesures. Elle a pour but de démarrer ou d'arrêter un procédé en fonctionnement de routine, de maintenir celui-ci près de son point de fonctionnement nominal ou de l'y amener et de gérer les défauts de façon à garantir la sécurité d'abord et un fonctionnement acceptable ensuite. C'est dans cette dernière perspective que des travaux de recherche restent à accomplir.

Les Systèmes Numériques de Contrôle Commande (SCADA Supervisory Control and Data Acquisition) proposés de façon standard par diverses entreprises ont déjà une part importante de leurs tâches consacrées à la supervision traditionnelle.

Dans ces systèmes, une grande partie du développement a été mis sur la conception des interfaces. On peut avoir une vision globale de l'architecture matérielle du procédé grâce à un synoptique, qui reproduit les schémas de l'installation, souvent sous la forme de représentations de type PID (Process and Instrumentation Diagram). On peut aussi accéder à la vision des historiques des variables, ou de plusieurs variables à la fois s'il est intéressant de corréliser leur évolution.

Dans un système traditionnel, la génération d'alarmes, correspondant aux comportements anormaux du procédé, est utilisée pour faciliter la surveillance et la supervision. Il s'agit de surveiller simplement que les variables restent à l'intérieur d'un domaine de valeurs caractéristiques du fonctionnement normal. On fixe donc un seuil haut SH et un seuil bas SB pour chaque variable et l'on vérifie qu'à chaque acquisition, la variable appartient à l'intervalle [SB, SH]. On peut remarquer qu'implicitement, cela revient à utiliser un modèle fort simple pour chaque variable : elle doit rester dans l'intervalle défini par les seuils.

Comme les alarmes sont traitées de façon booléenne, elles sont souvent utilisées pour mettre en route certaines commandes de manière automatique. C'est bien sûr le cas des arrêts d'urgence. Mais les situations peuvent être moins dangereuses et toutefois nécessiter certaines actions de reprise. Celles-ci sont souvent simples : fermeture/ouverture de certaines vannes si un niveau est trop haut, arrêt du chauffage si une température est en dépassement, mise en route d'une pompe de secours lorsque la pompe usuelle chauffe... On est amené parfois à programmer toute une procédure séquentielle pour ramener le procédé dans un mode normal.

La supervision des boucles de commande locales est la tâche la plus simple. Les commandes ont été calculées grâce à un certain nombre d'hypothèses simplificatrices sur le fonctionnement du procédé, la plus fréquente étant la linéarité autour d'un point de fonctionnement.

Pour mener à bien les tâches de plus haut niveau, qui concernent la gestion des défauts, des opérateurs humains sont présents sur toutes les installations complexes. L'objectif de l'automatique n'est pas de les remplacer, mais plutôt de les aider dans leurs tâches de haut niveau. Dans ce cadre, l'opérateur doit accomplir différentes actions à l'aide de différents raisonnements basés sur l'analyse du procédé. Une des contraintes de base du projet MAGIC a été de concevoir des outils d'aide à l'opérateur, qui assistent celui-ci pour prendre des décisions, ou qui lui expliquent une situation, sans se substituer à lui dans l'action.

1.3 Objectifs de la thèse

Dans un contexte général, ce travail de thèse est à considérer comme un projet de recherche et développement avec l'objectif fondamental de développer des outils génériques pour la supervision de procédés industriels, intégrables dans l'architecture multi-agents de MAGIC. En particulier, les efforts ont été mis sur :

- le développement d'algorithmes de détection de défauts fondés sur une représentation causale du procédé ;
- le développement d'outils pour l'aide à la prise de décisions, notamment
 - des algorithmes de prédiction de l'évolution des variables en défaut,
 - des algorithmes d'aide à la conduite et la maintenance.

Le développement de ces outils entraîne non seulement la conception, le développement et la mise en oeuvre des algorithmes qui seront utilisés en ligne, mais encore l'adaptation des modes de fonctionnement et des structures d'information d'entrée et de sortie pour les rendre compatibles avec les méthodes de communication implantées dans MAGIC (communication inter agents). Le développement des outils nécessaires pour la configuration des algorithmes mentionnés a aussi fait partie des objectifs du projet et de la réflexion méthodologique. La Figure 1-1 montre l'incidence de notre travail dans l'architecture de MAGIC.

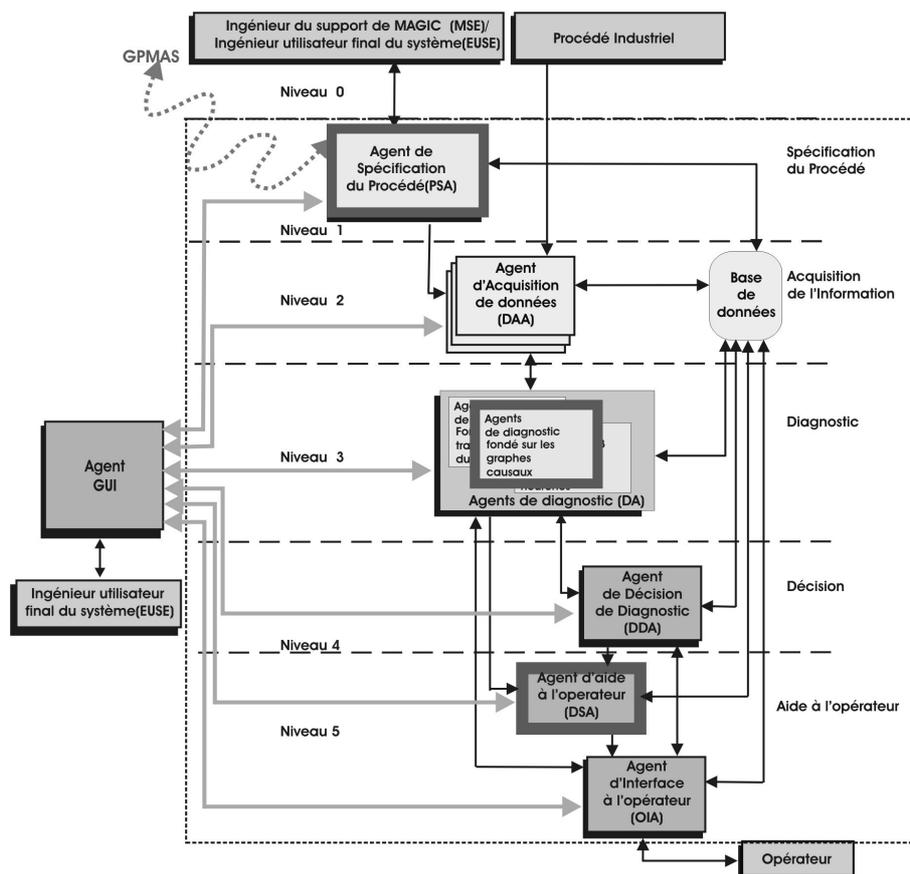


Figure 1-1. Incidence de notre travail dans l'architecture de MAGIC. *Entourés les agents, dont les algorithmes ont été développés au cours de ce travail de recherche. (développement d'algorithmes de détection fondée sur une représentation causale du procédé, développement d'outils pour l'aide à l'opérateur à la prise de décisions, outils de configuration et personnalisation pour ces algorithmes).*

1.4 Organisation du document

Nous présentons tout d'abord au **chapitre 2** le cadre scientifique du travail. On définira les caractéristiques de sûreté de fonctionnement exigées des systèmes industriels modernes, les risques qui menacent ces caractéristiques, et les moyens pour atteindre les objectifs de sûreté de fonctionnement souhaités. Ensuite, on établira les caractéristiques souhaitées pour les systèmes d'aide à la décision pour la supervision de procédés complexes. Finalement les concepts de base des agents logiciels et des systèmes multi-agents comme un

des moyens privilégiés pour la mise en oeuvre de systèmes informatiques complexes seront introduits.

Le **chapitre 3** est consacré à la description approfondie de l'architecture du système MAGIC. Cette description permettra de situer la contribution générale de MAGIC dans le contexte de la supervision avancée et de placer dans l'architecture générale les outils de diagnostic et d'aide à la supervision proposés dans cette thèse.

Le **chapitre 4** présente l'utilisation d'un système de détection de défauts fondé sur un modèle causal. Les fonctionnalités exigées des agents de diagnostic sont présentées, tant au niveau algorithmique de détection de défauts comme au niveau de communication inter agents. Dans la partie dédiée aux algorithmes, les concepts de simulation causale, et la génération de symptômes à partir de ce modèle sont expliqués. Ensuite, les interactions de communication entre cet agent et les autres agents de MAGIC sont définies. Ultérieurement, les méthodes utilisées pour la construction des graphes et l'identification des modèles dynamiques, utilisés dans le graphe causal, sont expliquées. Finalement on présente un exemple de détection de défauts, conçu autour d'un système simple qui permet d'illustrer les méthodes de façon pédagogique (système de bacs en cascade).

Au **chapitre 5** on présente les algorithmes proposés pour l'aide à la décision pour la conduite, la maintenance et la prédiction ainsi que les moyens de communication de l'Agent d'Aide à la Décision. On présente en particulier une alternative pour la prédiction des variables, la prédiction fondée sur les réseaux de neurones artificiels et la prédiction fondée sur l'analyse de tendances qualitatives. On suggère un algorithme qui détermine les moyens d'action possibles sur une variable touchée par un défaut, grâce à la représentation causale. On présente aussi une vision de la propagation des défauts de variable en variable, toujours grâce au graphe causal. L'exemple pédagogique est à nouveau utilisé dans ce chapitre pour montrer le potentiel des méthodes proposées.

Le **chapitre 6** présente les résultats que nous avons obtenus en appliquant les outils de détection de défauts sur un "Enrouleur Hydraulique" industriel. Ce chapitre propose une brève description d'un procédé de laminage et du rôle de l'Enrouleur Hydraulique dans ce procédé. L'obtention du modèle causal de l'Enrouleur Hydraulique grâce aux outils implantés dans MAGIC est commentée. Quelques résultats de détection de défauts obtenus par l'Agent de Diagnostic Causal sur des scénarios de données simulées et sur des données réelles de mauvais fonctionnement terminent cette présentation de l'exploitation industrielle de nos travaux.

Une conclusion présentant nos principaux apports et donnant quelques perspectives de développement futur achève ce mémoire.

Chapitre 2

La supervision avancée des procédés

2.1 Introduction.

Ce chapitre va introduire les différents concepts pour la compréhension des problèmes de supervision avancée. On a déjà décrit la supervision traditionnelle dans le chapitre antérieur. Dans ce chapitre, on fera entrevoir ce que peut être un système de supervision avancée, ou plus proprement dit, un système d'aide à la décision pour la supervision. On y décrira la problématique associée aux systèmes industriels modernes au niveau de la supervision, et les divers outils utilisés pour gérer ces problèmes.

Bien que les problèmes de sûreté de fonctionnement soient présents dans tous les domaines, dans ce chapitre, et en général dans cette thèse, on sera toujours orienté vers les problèmes de systèmes industriels.

Les systèmes de production modernes doivent être vu comme l'union d'un processus (matériel physique) et un système de commande (logiciel). Donc les défaillances peuvent arriver dans les deux systèmes.

2.2 La sûreté de fonctionnement des systèmes industriels.

La technologie des procédés a évolué vers des industries plus complexes avec un degré élevé d'automatisation. Un système complexe se réfère à un système composé «d'éléments», en général simples, fortement interconnectés. Le degré de complexité d'un système peut être mesuré en fonction de la quantité d'information nécessaire pour le décrire à un certain niveau de détail, c'est-à-dire du nombre d'éléments appartenant au système, de l'intensité, la manière, la diversité, et les modes d'opération des interactions entre ces éléments et avec leur environnement et la quantité d'activités et d'objectifs du système [Bar-Yam 1997]. À cet égard, on parlera ici de systèmes industriels complexes en relation aux systèmes de grande échelle, avec des dynamiques non linéaires. Des exemples de cette sorte de systèmes sont les réseaux de distribution d'énergie et de télécommunications, les procédés chimiques, et les avions modernes [Åström, Albertos et al. 2001]. Par exemple, une grande raffinerie peut avoir jusque 10 000 boucles de régulation automatique, et une machine à papier jusqu'à 5 000. La probabilité d'un fonctionnement défectueux augmente avec le degré de complexité ainsi que les conséquences des erreurs [Åström, Albertos et al. 2001]. En conséquence, on doit concevoir des systèmes avec un haut degré de sûreté de fonctionnement [Blanke 1996].

La sûreté de fonctionnement est synonyme d'un degré élevé de *disponibilité*, *fiabilité*, *sécurité*, *intégrité* et *maintenabilité* [Blanke 1996]. Selon la nature du système, on peut attacher plus ou moins d'importance à chacun de ces attributs ; ces attributs indiquent comment un défaut dans le système affecte la production, le procédé et l'environnement. Une exposition systématique des concepts liés à la sûreté de fonctionnement se compose de trois parties [Avizienis, Laprie et al. 2001]: les risques qui menacent la sûreté de fonctionnement, les attributs de la sûreté de fonctionnement, et les moyens par lesquels la sûreté de fonctionnement est atteinte (Figure 2-1). Ces concepts seront expliqués dans les prochaines trois sections.

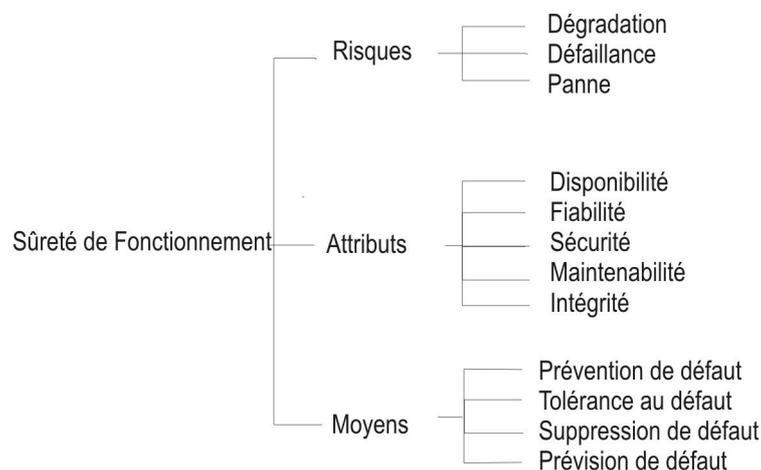


Figure 2-1. L'arbre de sûreté de fonctionnement, modifié d'après [Avizienis, Laprie et al. 2001]

2.2.1 Les attributs des systèmes sûrs de fonctionnement.

Le degré de sûreté de fonctionnement d'un système est donné en terme d'objectifs requis pour chacun des attributs mentionnés ci-dessus, de la durée maximale acceptable de

l'interruption, de l'ensemble de défauts spécifiques et de l'environnement dans lequel se trouve le système. Divers auteurs ont précisé différents attributs de base pour définir la sûreté de fonctionnement [Villemeur 1988; Zwingelstein 1995; Isermann and Ballé 1997; Jonsson 1998; Avizienis, Laprie et al. 2001], on présente ici ceux qui sont en général les plus acceptés:

Disponibilité : La mesure « probabiliste » $A(t)$ de l'aptitude d'une entité en état d'accomplir correctement une fonction requise dans des conditions données et à un instant k .

Fiabilité : La mesure « probabiliste » $R(t)$ de l'aptitude d'une entité à accomplir une fonction requise, pendant une durée donnée et dans des conditions données.

Sécurité : La mesure de l'aptitude d'une entité à éviter de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques.

Maintenabilité : La capacité $M(t)$ du système à subir toutes les activités destinées à conserver ou à rétablir un composant dans un état donné, pour accomplir une fonction requise. Ces activités sont une combinaison d'activités techniques, administratives et de gestion des réparations ou modifications.

Intégrité : La caractéristique protectrice du système pour la prévention d'altérations non autorisées du système.

2.2.2 Les risques qui menacent la sûreté de fonctionnement.

Dégradation : une dégradation est l'état d'une entité ou d'un sous-ensemble présentant une perte de performances d'une des fonctions assurées par l'entité (si les performances sont au-dessous du seuil d'arrêt défini dans les spécifications fonctionnelles, il n'y a plus dégradation mais défaillance) [Zwingelstein 1996]:

Défaillance : une défaillance est l'altération ou la cessation de l'aptitude d'un composant à accomplir sa ou (ses) fonction(s) requise(s) avec les performances définies dans les spécifications techniques. Cette altération ou cessation peut être causée par une défaillance interne du composant, ou encore par une défaillance externe (utilisation du composant hors des contraintes acceptables).

Panne : la panne est l'inaptitude d'un dispositif à accomplir une fonction requise.

Ces trois aspects sont liés à la fonction du système (ou du composant). On ajoute ici le terme défaut, qui n'est pas lié à la fonction du système mais à ses objectifs ou spécifications.

Défaut : un défaut est un écart entre un comportement attendu et un comportement observé. Cet écart est constaté grâce à un indicateur de défaut.

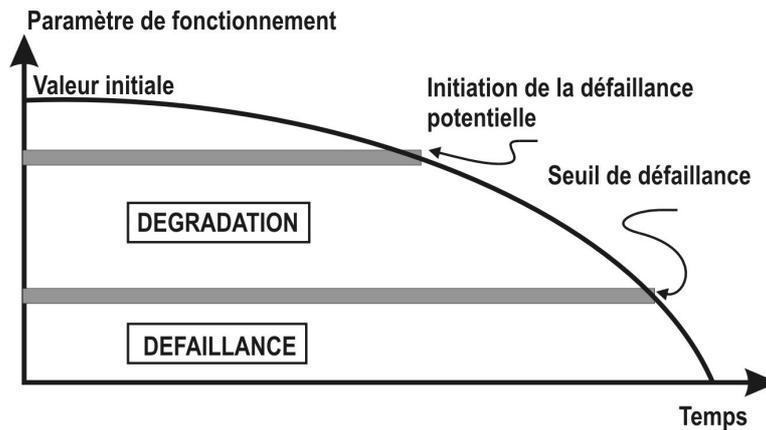


Figure 2-2. Dégradation et défaillance [Zwingelstein 1996]

2.2.3 Les moyens pour atteindre la sûreté de fonctionnement.

Le développement de systèmes sûrs de fonctionnement doit mettre en jeu diverses techniques afin de faire face aux risques :

La prévention de défauts : la prévention de défauts est atteinte en utilisant diverses techniques du domaine de la fiabilité et de la qualité pendant la conception et la construction du système (ou composant).

La suppression de défauts : la suppression de défauts est effectuée pendant la phase de développement, et pendant la vie opérationnelle d'un système. La suppression de défauts pendant la phase de développement est gérée par les procédures de vérification, de diagnostic et de correction. La suppression de défauts pendant l'exploitation du système est faite grâce à la maintenance corrective ou préventive.

La prévision de défauts : la prévision de défauts est menée en exécutant une évaluation du comportement du système par rapport à l'occurrence d'un défaut. L'évaluation a deux aspects : l'aspect qualitatif qui identifie et classe les modes de défaillance, ou les combinaisons d'événements qui mèneraient aux défaillances du système. L'aspect quantitatif qui fait son évaluation en termes de probabilités dans la période pendant laquelle des attributs de la sûreté de fonctionnement sont satisfaits. Des exemples de ces dernières sont la durée moyenne de fonctionnement d'une entité avant la première défaillance (MTTF) et la durée moyenne de fonctionnement après réparation (MUT).

La tolérance aux défauts : La tolérance aux défauts est prévue pour préserver les fonctionnalités du système en présence des défauts actifs. Elle est généralement conçue comme une démarche en deux phases : une phase de diagnostic et une phase subséquente de réglage du système pour réagir au défaut.

La plupart des développements appliquent ces techniques de manière isolée, peu nombreux sont les travaux les combinant. On peut trouver quelques exemples dans [Blanke 1996; Dash and Venkatasubramanian 2001; Isermann, Schwarz et al. 2002].

2.2.3.1 Les techniques employées pendant les phases de conception et de vérification du système.

La prévention, la prévision et une partie de la suppression des défauts doivent être accomplies dans les phases de conception et de vérification du système. Afin d'étudier les effets des défauts sur la *fiabilité* et la *sécurité* pendant ces phases, une gamme des méthodes d'analyse ont été développées [Villemeur 1988; Isermann, Schwarz et al. 2002]. Elles incluent :

- l'analyse préliminaire de dangers (APD)
- l'analyse des modes de défaillance et de leurs effets (AMDEC)
- la méthode de l'arbre des causes (MAC)
- la méthode de l'arbre de causes (MAC) et l'analyse de l'arbre de conséquences (MACQ)
- l'analyse du type HAZOP
- la méthode du diagramme causes-conséquences (MDCC).

Isermann (Isermann 2002) propose une démarche pour la combinaison de méthodes fiabilistes pour l'analyse de défaillances, pour la conception de moyens pour assurer la fiabilité, la sécurité, la maintenabilité et la tolérance aux défauts (Figure 2-3).

La procédure commence avec une analyse du type AMDEC, qui identifie tous les composants, les défaillances, les causes et les effets. Les défaillances sont analysées par la méthode de l'arbre de causes pour déterminer les causes et leurs dépendances logiques à un niveau composant. Les causes de défauts sont alors employées pour concevoir la fiabilité globale. Les défauts restants qui ne peuvent pas être évités sont alors classifiés et la procédure de maintenance est déterminée.

Basée sur l'AMDEC, l'analyse de risque extrait les défauts critiques pour la sécurité. Leur présentation dans un arbre (réduit) de défauts détermine les causes au moyen des interconnexions logiques. Basés sur les résultats de cette analyse, les systèmes de sécurité de bas niveaux peuvent être conçus. Les défaillances dangereuses restantes subissent alors une classification selon le risque et les méthodes de supervision pour la réduction des risques aux niveaux acceptables sont déterminées.

2.2.3.2 Les techniques de l'automatique employées pendant la phase d'exploitation du système

Vis-à-vis du système de contrôle/commande, la phase d'analyse de fiabilité et de risque permet la conception (et la spécification des cahiers des charges) des sous-systèmes de protection automatiques, des verrouillages, et des systèmes de contrôle/commande tolérants aux défauts (quand c'est possible), lesquels seront utilisés en ligne dans la phase d'exploitation du système.

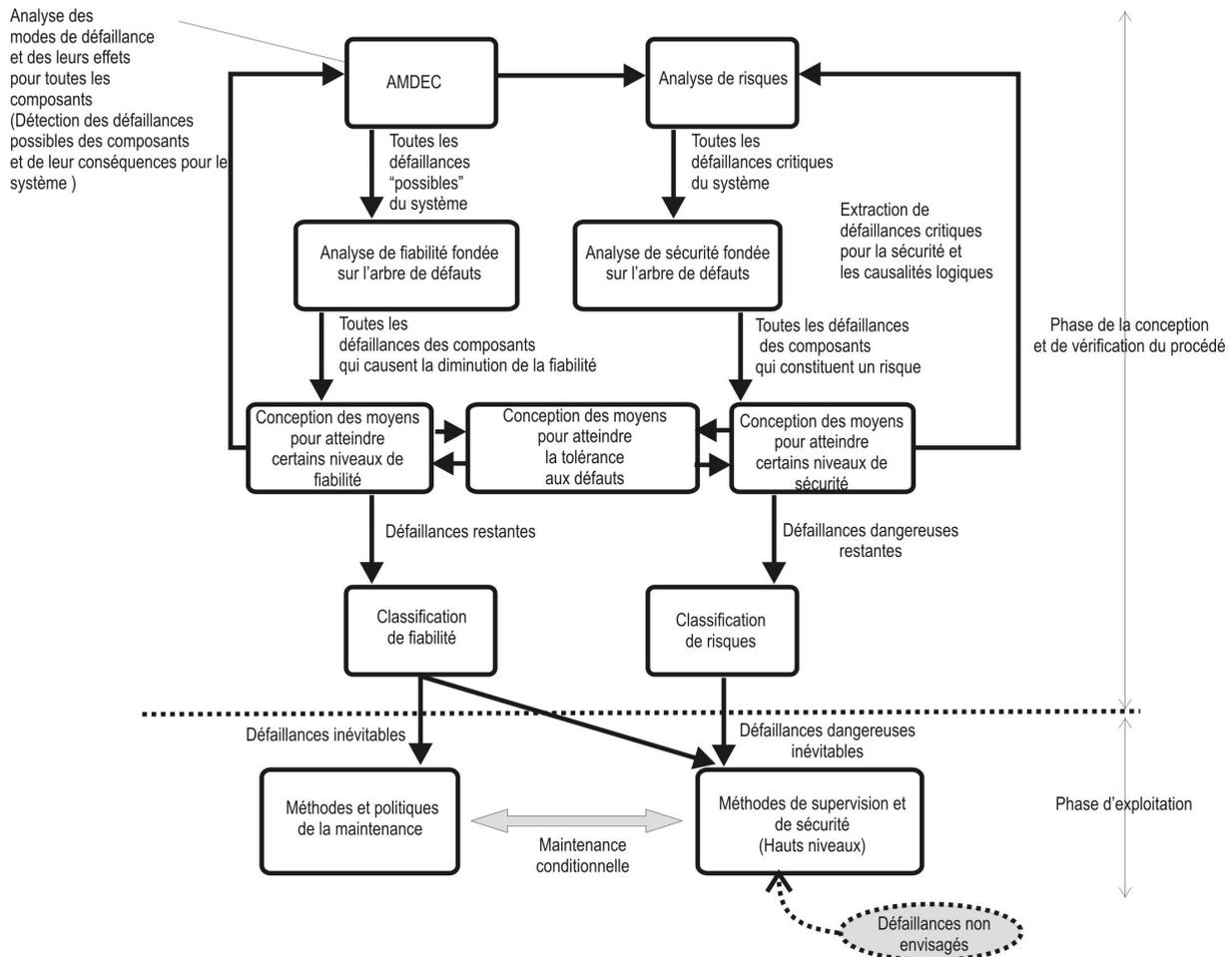


Figure 2-3. Procédure intégrée pour la conception de systèmes sûrs de fonctionnement, basée sur [Isermann, Schwarz et al. 2002]. Nous avons ajouté un lien entre les méthodes de supervision et la maintenance pour établir leur relation vis-à-vis de la maintenance conditionnelle. La petite bulle de défaillances non envisagées a été aussi ajoutée pour déclarer que la conception des systèmes de supervision doit aussi prévoir qu'il existe toujours des défaillances non prévues.

Les systèmes de contrôle/commande tolérants aux défauts :

Les systèmes de commande tolérants aux défauts peuvent être obtenus en donnant au système (ou sous systèmes) de commande la capacité de détecter, de localiser et d'identifier des défauts, et de réagir avec des actions pour s'adapter au défaut. Récemment, les concepts de commande tolérante aux défauts ont été définis. Pour bien établir le problème de la commande tolérante aux défauts, on commencera avec la définition du problème de la commande standard :

Définition 2-1 :

Le problème de *commande standard* est défini par un objectif de commande $\mathbf{o} \in \mathbf{O}$, une loi de commande $\mathbf{u} \in \mathbf{U}$ et des contraintes $\mathbf{c} \in \mathbf{C}$. Celles-ci définissent la représentation du procédé (structure $\mathbf{s} \in \mathbf{S}$ et paramètres $\boldsymbol{\theta} \in \boldsymbol{\Theta}$). Résoudre le problème de la commande consiste

à choisir la loi de commande $\mathbf{u} \in \mathbf{U}$ qui vérifie \mathbf{c} et des égalités ou inégalités à satisfaire, et permettant d'atteindre \mathbf{o} . Ceci peut être formulé de la façon suivant :

$$\text{Résoudre } \langle \mathbf{o}, C(\mathbf{s}, \boldsymbol{\theta}), \mathbf{U} \rangle$$

où $C_n(\mathbf{s}_n, \boldsymbol{\theta}_n)$ sont donnés par le procédé

où n indique le régime nominal.

En général, un indicateur de performance Π est associé à \mathbf{o} afin d'affiner la recherche de \mathbf{u} ⁴.

La commande tolérante aux défauts concerne la commande du système défectueux. Ceci peut être fait en changeant la loi de commande sans changer le système qui est actionné, ou en changeant la commande et le système.

Définition 2-2 :

Le problème de la *commande tolérante* peut être interprété comme une extension du problème de la commande standard dans lequel $\mathbf{s}_f \subset \mathbf{S}$ représente la structure possible du système défectueux dans l'ensemble des structures possibles et $\boldsymbol{\theta}_f \subset \Theta$ le sous-ensemble des paramètres qui lui sont associés dans l'ensemble de tous les paramètres possibles. La commande tolérante est alors obtenue par :

$$\text{Résoudre } \langle \mathbf{o}, C_f(\mathbf{s}_f, \boldsymbol{\theta}_f), \mathbf{U}_f \rangle$$

où f indique le mode de défaillance, et $C_f(\mathbf{s}_f, \boldsymbol{\theta}_f)$ est l'estimation des nouvelles restrictions du procédé, sont déterminés lors de la tâche de diagnostic.

Tolérance passive aux défauts. Dans la tolérance passive aux défauts, la loi de commande n'est pas changée quand le défaut arrive. Cela signifie que les objectifs du système peuvent être obtenus quand le système est en bonne santé, aussi bien que quand le système est défectueux. C'est évident que la gamme d'application de ce type de techniques est très limitée aux petits défauts.

Tolérance active aux défauts. Dans la tolérance active aux défauts, pour chacun des cas (système en bonne santé et le système défectueux) il existe une solution appropriée. Comme chacun des modes de fonctionnement (normal et anormaux) est associé à un problème de commande, le système doit connaître dans quel mode de fonctionnement il est ; ce sont les algorithmes de diagnostic qui fournissent cette connaissance. Il existe deux stratégies pour atteindre la tolérance active, l'accommodation aux défauts et la reconfiguration.

⁴ Par exemple, \mathbf{o} pourrait être « garantir la stabilité du système » et Π une marge de stabilité.

L'accommodation de la commande aux défauts consiste en un changement de paramètres du contrôleur ou de sa structure afin de commander le système en défaut et d'éviter les conséquences d'un défaut. Les connexions entre les entrées et les sorties du système de commande et le procédé sont inchangées. L'objectif initial de la commande est atteint bien que les indicateurs de performance peuvent être dégradés.

Définition 2-3

Le problème de *l'accommodation de la commande* consiste à résoudre $\langle \mathbf{o}, \hat{\mathbf{s}}, \hat{\boldsymbol{\theta}}, \mathbf{U} \rangle$ où $\langle \hat{\mathbf{s}}, \hat{\boldsymbol{\theta}} \rangle$ sont les estimations des nouvelles contraintes et des paramètres du système défaillante, déterminées lors de la tâche de diagnostic.

Le défaut peut être accommodé si le problème $\langle \mathbf{o}, \hat{\mathbf{s}}, \hat{\boldsymbol{\theta}}, \mathbf{U} \rangle$ présente une solution. Ce qui signifie que la loi de commande $\mathbf{u} \in \mathbf{U}$ permet au système d'atteindre ses objectifs initiaux.

La reconfiguration du système de commande consiste en un changement des entrées-sorties entre le contrôleur et le procédé par le changement de a) la structure et des paramètres de contrôleur et/ou b) le changement de la structure du procédé. Le cas b) suppose la possibilité d'arriver à une structure du procédé qui contient seulement des composants en bon état. L'objectif original de la commande est atteint quoique les indices de performance puissent se dégrader.

Définition 2-4

Le problème de la *reconfiguration de la commande* consiste à déterminer un ensemble de contraintes $(\mathbf{s}_d, \boldsymbol{\theta}_d) \in (\mathbf{S}, \boldsymbol{\Theta})$ et $\langle \mathbf{o}, \mathbf{s}_d, \boldsymbol{\theta}_d, \mathbf{U} \rangle$ a une solution.

La loi de commande $\mathbf{u}_d \in \mathbf{U}$ est appliquée au système dégradé. Néanmoins, les objectifs \mathbf{o} , sont supposés atteignables malgré les changements de contraintes et de paramètres opérés lors de la phase de diagnostic.

Les défaillances inévitables doivent être couvertes par la maintenance et les méthodes de supervision et de sûreté du fonctionnement [Isermann, Schwarz et al. 2002]. Evidemment les défauts « imprévus » doivent être aussi couverts par la supervision. On définit l'accommodation des objectifs comme la capacité de surveiller si les objectifs de la commande sont remplis et, dans le cas échéant, calculer des nouveaux objectifs de commande et une nouvelle structure et paramètres de commande qui font que le système défectueux atteint les nouveaux objectifs. L'accommodation des objectifs est menée au niveau de la supervision et devrait entrer en vigueur si les défauts se produisent et s'il n'est pas possible de répondre à l'objectif d'origine de commande dans le schéma tolérant au défaut. En continuant avec la nomenclature utilisée dans la section antérieure, on peut définir le problème de l'accommodation des objectifs comme :

Définition 2-5

Le problème de l'*accommodation des objectifs* consiste à surveiller le triplet $(\mathbf{O}, \mathbf{S}, \boldsymbol{\theta})$ et déterminer si l'objectif de commande est atteint. S'il ne l'est pas, un objectif dégradé $\mathbf{o}_d \in \mathbf{O}$ et une structure dégradée $(\mathbf{s}_d, \boldsymbol{\theta}_d) \in (\hat{\mathbf{S}}, \hat{\boldsymbol{\theta}})$ doit être déterminée, tel que le problème relâché $\langle \mathbf{o}_d, \mathbf{s}_d, \boldsymbol{\theta}_d, \mathbf{U} \rangle$ ait une solution.

On trouve dans la littérature, par exemple dans [Blanke, Kinnaert et al. 2003] le nom « problème de supervision » pour définir ce problème, mais, comme on le verra dans la prochaine section, cette étiquette est un peu réductrice par rapport à l'ensemble de problèmes que doivent être gérés au niveau de la supervision.

2.3 Les systèmes d'aide à la décision de l'opérateur pour la supervision: aide à l'opération et la maintenance.

Dans la section précédente, on a expliqué la démarche "idéale" pour la construction de systèmes sûrs de fonctionnement. Certains défauts sont supprimés pendant les étapes de développement et de vérification du procédé ; la redondance physique est ajoutée pour quelques composantes critiques; les systèmes de protection automatique et les verrouillages d'urgence sont conçus et mis en oeuvre pour assurer la sécurité du système, et certains systèmes de commande tolérants aux défauts sont aussi conçus et mis en oeuvre pour tolérer certains défauts. Toutefois tous ces processus sont faits, la plupart du temps, par étapes distantes dans le temps, sauf pour des processus réellement critiques (les centrales nucléaires ou les avions par exemple) ou les systèmes embarqués de petite échelle (par exemple les actionneurs intelligents [Stolz 1998]). Beaucoup d'industries suivent des améliorations progressives dans leurs systèmes de protection ; les analyses de fiabilité et de risques sont menées après que le processus a été construit, ce qui complique leur modification ; et, la plupart des procédés industriels sont commandés par des systèmes de régulation simples du type PID, et non par des algorithmes avancés de commande tolérant aux défauts. Tout cela entraîne en fait que dans la réalité les niveaux de « supervision » doivent supporter beaucoup plus de défauts que ceux envisagés dans cette démarche idéale, soit via la conduite soit via la maintenance du système. C'est dans cette perspective que se situent nos travaux.

Quand des décisions stratégiques concernant les buts du système, comme la solution aux problèmes de supervision, doivent être prises, les opérateurs humains sont généralement impliqués, avec des systèmes d'aide à la décision et de diagnostic et en prenant en compte les buts globaux du procédé [Blanke, Frei et al. 2000; Staroswiecki and Gehin 2000]. Les limites actuelles de l'automatique, principalement pour la résolution des problèmes complexes que pose la supervision, notamment de prise de décision, imposent la présence d'êtres humains dans la boucle [Popieul 2003]. Sa capacité à anticiper, à gérer des situations nouvelles, à créer, à s'adapter, à décider, rend nécessaire la présence de l'homme au sein des systèmes complexes [Hoc 1993]. Cependant, la surcharge cognitive de l'opérateur générée par la grande quantité de variables changeant constamment et de composants à surveiller et les interactions entre eux dans une usine rendent nécessaire des outils d'aide à la décision [Bainbridge 1993]. Un grand nombre de variables peuvent avoir une valeur anormale simplement parce qu'elles sont les conséquences d'autres variables ayant des valeurs

anormales. Les alarmes se propagent généralement avec le temps tout le long du procédé, donnant ce qui s'appelle parfois un "arbre de Noël". Ainsi il y a un besoin de traiter globalement les divers symptômes d'un comportement anormal afin d'aider les opérateurs à assimiler rapidement les données, pour relier toutes les observations à un composant défectueux (souvent loin des alarmes observées) et pour diagnostiquer un état potentiellement dangereux. L'objectif d'un système d'aide à la décision de l'opérateur dans la supervision est donc de réduire au minimum les chances d'erreur de l'opérateur face à une situation anormale et de permettre une réduction significative de sa charge de travail sans toutefois altérer sa vigilance ou ses connaissances.

Notre travail considérera donc toujours cette complémentarité nécessaire entre automatismes et opérateurs. Cela guidera nos choix, aussi bien des méthodes de diagnostic que des méthodes d'aide à la décision.

2.3.1 Systèmes d'aide à la décision.

Les systèmes d'aide à la décision expliqués dans la section précédente sont conformes aux propriétés de la prise de décision complexe [Alty 1997] :

- La prise de décision exige la coopération entre les opérateurs humains et le système.
- Le temps est habituellement critique à certaine(s) étape(s) dans le processus de décision. Les événements qui affectent le processus décisionnel sont souvent en dehors de la commande des opérateurs et procèdent en temps réel.
- Une grande quantité d'information est habituellement impliquée dans le processus de décision.
- La collecte de données, et leur traitement sont exécutés par systèmes informatiques.

Les systèmes d'aide à la décision concernent une large variété de systèmes, outils et technologies qui peuvent être employés pour aider dans la résolution des problèmes complexes de prise de décision [Borne, Fayech et al. 2003].

Divers paramètres de qualité d'un système d'aide à la décision ont été proposés pour évaluer ses performances [Meystel 2001; Boné and Crucianu 2002; Fayech, Hammadi et al. 2002]:

- Le degré de réduction de la complexité du problème.
- Le degré de coopération du système avec d'autres systèmes ou avec l'opérateur humain.
- L'augmentation de fonctionnalités de l'opérateur humain.
- Le degré « d'intelligence » du système d'aide à la décision dans les solutions proposées.
- Le degré d'autonomie du système d'aide à la décision pour proposer des solutions.
- Le degré d'augmentation de la probabilité de succès dans la résolution des problèmes.

Dans le contexte d'aide à la décision dans la supervision, diverses tâches pour la gestion de la conduite et de la maintenance des procédés industriels nécessaires pour atteindre les degrés de sûreté de fonctionnement souhaités sont situées au niveau de la supervision. Subséquemment, les systèmes avancés d'aide à la décision pour la supervision sont conçus

Dans le but d'aider l'opérateur dans sa tâche, le système d'aide à la décision doit faire face à la complexité du système. Si cette complexité entraîne une diminution de sa sûreté de fonctionnement, les possibilités de traitement numérique vont par contre contribuer à l'amélioration de certains de ces attributs [Staroswiecki 1994]. Ces outils doivent prendre en considération que le raisonnement de l'opérateur est qualitatif plutôt que quantitatif. L'opérateur traite de données symboliques. Les systèmes de supervision devraient combiner la technologie algorithmique avec des techniques d'intelligence artificielle, afin de fournir une meilleure coopération entre les humains et le système.

Le système d'aide à l'opérateur pour la supervision doit donc fournir des outils « utiles » d'aide à la décision ; c'est-à-dire, des outils qui servent d'appui dans la démarche naturelle de résolution des problèmes chez l'opérateur. À cet égard, l'opérateur humain traite l'information au moyen d'une séquence d'activités cognitives. Le modèle présenté sur la Figure 2-4 décrit ce processus. La branche gauche (ascendante) est une démarche d'analyse de la situation conduisant à un diagnostic, sa branche droite (descendante) une démarche de planification de l'action [Rasmussen 1993]. Ces deux démarches s'appuient aussi sur l'aspect de prédiction de l'évolution du système [Hoc 1993].

On utilise le terme de *coopération verticale* (**Figure 2-5**) pour nommer le type d'assistance à l'opérateur au moyen d'un système automatisé où l'opérateur reste le seul maître des actions sur le procédé en toutes circonstances. D'ailleurs, les études ergonomiques rejettent l'utilité de l'automatisation complète pour les procédés complexes. La supervision doit encore être centrée sur l'opérateur humain, et conçue pour une meilleure coopération homme-machine [Milot 1996]. Deux types de coopération verticale sont envisageables, la *coopération verticale active* et la *coopération verticale passive* [Gandibleux 1993] :

- Dans la coopération verticale active, l'opérateur consulte, informe ou interroge le système d'aide à la décision. La prise de décision est commune à l'opérateur et au système d'aide et se fait par étape. Compte tenu des contraintes d'interaction lourdes qui sont induites, cette solution n'est pas applicable à la gestion des situations d'urgence.
- Dans la coopération verticale passive, le système est capable d'élaborer lui-même la solution, et de la communiquer à l'opérateur humain qui peut choisir de la mettre en oeuvre. Cette solution est bien adaptée à la gestion des situations d'urgence dans la mesure où le système est capable de trouver et de justifier une solution dans toutes les situations.

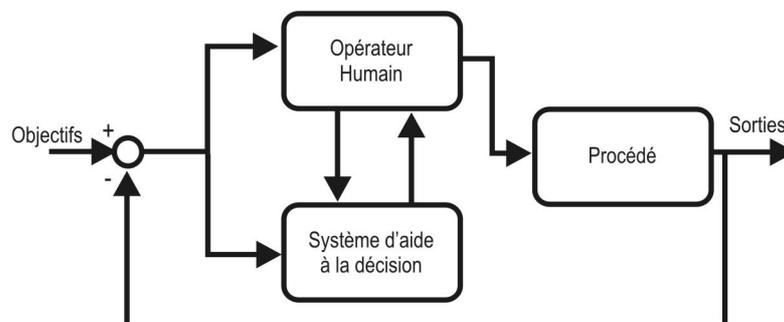


Figure 2-5. Coopération verticale.

Le domaine de recherche des systèmes modernes d'aide à la décision est concentré sur la théorie et l'application des systèmes intelligents comme les systèmes experts, les systèmes multiagent, le calcul qualitatif, les algorithmes évolutionnistes et la logique floue [Carlsson and Turban 2002]. Ceci inclut les processus de planification, et de prise de décision. Le contexte pour cette recherche s'étend à plusieurs niveaux, de la gestion stratégique de production jusqu'à la production elle-même.

Dans la suite de cette section, on parlera de quelques outils d'aide à la décision dans les démarches d'analyse, de planification et de prédiction de l'évolution du système, notamment :

- le diagnostic,
- le pronostic,
- les conseils d'action pour la conduite,
- l'aide dans le cadre de la maintenance conditionnelle,
- les interfaces pour l'opérateur.

En reprenant la Figure 2-4, on peut bien identifier l'utilité de ces outils. D'abord, l'opérateur doit identifier si le système fonctionne correctement ou pas, il est donc utile d'avoir un système de détection de défauts pour l'aide dans cette tâche. Après, l'opérateur cherche des explications, ce qui est fait grâce aux techniques de diagnostic. Le pronostic aide l'opérateur dans les démarches d'anticipation de l'évolution du système et d'interprétation des conséquences. Finalement, les conseils d'action et de maintenance sont utilisés pour la définition de tâches.

2.3.2 Diagnostic pour la surveillance : détection, localisation et identification de défauts

La détection et le diagnostic de défaillances sont deux fonctions de base d'un système de supervision industriel avancé. Leur objectif est l'activation d'actions appropriées sur un système en fonction de son état ou son mode de fonctionnement [Isermann 1997]. La Figure 2-6 montre la position de ces deux fonctionnalités.

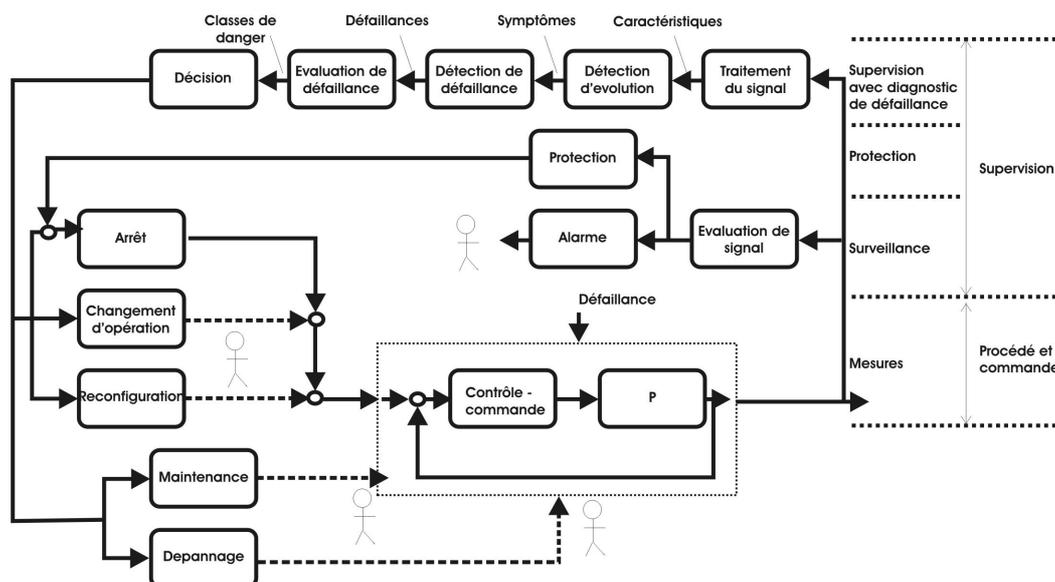


Figure 2-6. Positionnement des différentes méthodes dans le contexte de la supervision [Isermann 1997]

Les problèmes de gestion de défauts au niveau global d'un procédé sont d'une grande difficulté. Ils concernent en effet une grande quantité d'information. Remarquons aussi que pour la plupart des systèmes complexes, il y a continuité entre le fonctionnement «normal» et «dégradé», plutôt qu'un saut brutal de l'un à l'autre. Une qualité attendue du système de gestion de défauts est alors la précocité, car comme on a précisé quelques paragraphes plus haut, l'opérateur humain préfère anticiper qu'agir précipitamment. On retrouve clairement dans les étapes du diagnostic la démarche générale de prise de décision chez un opérateur humain, telle que Rasmussen la décrit. Nous allons donc dans cette section nous intéresser aux approches actuelles du diagnostic proposées par la communauté des automaticiens.

La *dégradation* d'un composant est la perte de performances d'une des fonctions assurées par l'entité ; quand cette perte dépasse une certaine limite, on parle d'une *défaillance*, dans laquelle le composant est incapable d'exécuter sa fonction. Aussi bien les *dégradations* que les *défaillances* peuvent causer un ensemble de *défauts* dans le système, ce qui signifie qu'une certaine variable caractéristique ou propriété s'écarte d'une gamme acceptable. Pour les opérateurs humains ou pour les systèmes de surveillance automatique, la manière d'observer les défauts est au moyen de symptômes. Les *symptômes* sont des changements des entités observables ou calculées par rapport à leur comportement normal [Isermann and Ballé 1997]. Le diagnostic est formellement défini comme la détermination du composant défectueux (*localisation*) et du genre et de la taille de cette perte de performances : de *dégradation (acceptable)* à *défaillance «totale» (panne) (identification de la défaillance)*. Cette tâche suit la tâche de *détection de défauts*, où la présence des défauts est déterminée. De cette façon, le diagnostic est une évaluation de l'état actuel (et passé) d'un système basée sur des symptômes observés.

Dans plusieurs cas, les trois étapes sont fortement liées et il est difficile dans certains cas de pouvoir différencier la limite entre la détection et la localisation ou entre la localisation et l'identification de défaut. Dans les sections suivantes on essaiera de faire une classification de méthodes de détection et de localisation/identification.

2.3.2.1 Les méthodes de détection de défauts.

La sélection d'une méthode de détection de défauts dépend clairement du procédé et de l'information disponible. De nombreuses méthodes ont été proposées dans les dernières décennies, les principales contributions peuvent être trouvées dans les congrès IFAC SAFEPROCESS. Diverses classifications de ces méthodes ont été aussi rapportés [Venkatasubramanian, Rengaswamy et al. 2002].

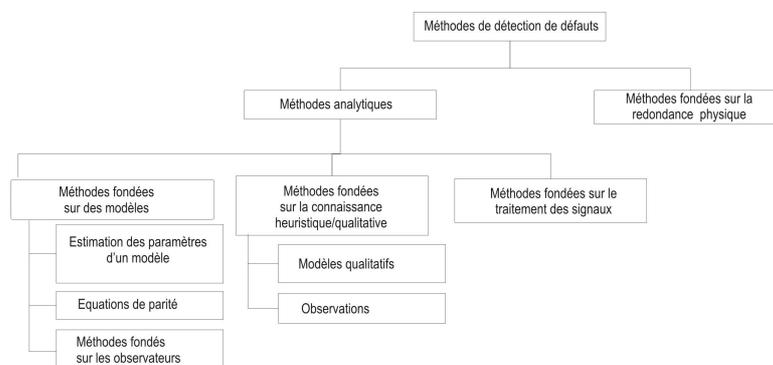


Figure 2-7. Classification des méthodes de détection de défauts.

- *Les méthodes fondées sur les modèles*

Les algorithmes de détection de défauts utilisent la redondance analytique et mettent en oeuvre la démarche suivante : 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. Si celles-ci ne vérifient pas les équations du modèle, on en déduit que le fonctionnement réel n'est pas le fonctionnement normal. La cohérence entre le système et le modèle peut être vérifiée au temps t grâce aux *résidus*, qui représentent la différence entre la valeur mesurée et la valeur de référence :

$$r(t) = y(t) - \hat{y}(t)$$

Dans le cas sans défaut, la valeur du résidu est proche de zéro. Une valeur différente de zéro indique l'existence de défaut.

Le modèle est habituellement développé sur la compréhension de la physique du processus. Dans les modèles quantitatifs, cette compréhension est exprimée en termes des relations mathématiques entre les entrées et les sorties du système.

- **Observateurs :** Les observateurs d'état sont des algorithmes, fondés sur un modèle du procédé, chargés de poursuivre l'état de celui-ci. On essaye, en général, de rendre l'observateur indépendant des perturbations non mesurées (entrées inconnues) et dépendant de certains défauts [Patton and Chen 1997].
 - **Equations de parité :** un modèle du procédé est alimenté en parallèle et reconstruit les sorties qui sont comparées aux mesures. Il est possible d'utiliser plusieurs modèles. Pour décrire des procédés non linéaires, certains auteurs proposent d'utiliser des réseaux de neurones comme modèles du type « boîte noire ».
 - **Les paramètres nominaux du modèle** sont déterminés par des méthodes d'identification de paramètres pendant la période d'apprentissage hors-ligne. Les paramètres estimés en ligne sont ensuite comparés aux paramètres nominaux pour la détection de défaut. Cette méthode est surtout utilisée lorsque la structure du modèle est bien connue et ses paramètres ont un sens physique. Cette méthode a besoin de signaux suffisamment excités pour que l'algorithme d'identification converge.
- *Les méthodes fondées sur la connaissance.*

La connaissance sur les systèmes physiques est souvent partielle et imprécise et il n'est pas toujours possible d'en élaborer un modèle par simulation « classique ». Dans ce cadre, deux approches sont distinguées, les approches purement qualitatives et les approches semi-qualitatives.

Le raisonnement qualitatif permet de formaliser la connaissance dans des cas pratiques et de développer des modèles du mode de pensée humain à propos des systèmes physiques [Forbus 1984]. Cette connaissance peut se représenter par une tendance

(généralement caractérisée par le signe des dérivées des mesures ou des résidus) ou un ordre de grandeur (faible, moyen, grand). Les modèles qualitatifs peuvent être utilisés lorsque aucun modèle mathématique suffisamment précis n'est disponible. Le traitement de l'information qualitative pour la détection de défauts peut être accompli par une base de règles [Pessi and Luparia 1992].

- *Les méthodes fondées sur le traitement de signal.*

Dans un certain nombre de cas, il est impossible d'obtenir le modèle mathématique du procédé, ou d'une partie de celui-ci, reliant les signaux d'entrée à ceux de sortie. On peut toutefois analyser les signaux issus de l'instrumentation et proposer pour chacun de ces signaux un modèle. En général, ce type d'analyse dépend de la région d'opération du système. Ce modèle est en fait une référence du comportement de ce signal, et qui peut être représentatif de l'état normal de l'installation ou d'un défaut particulier.

- Analyse statistique : dans cette approche, on analyse en général les signaux issus du procédé pendant son fonctionnement normal et on déduit une loi de densité de probabilité pour chaque signal analysé. La détection consiste à déterminer ensuite en temps réel si le signal obéit bien à cette loi, ou s'écarte significativement. De nombreux tests ont été proposés, en particulier le test de saut de moyenne et le test de variance [Basseville and Nikiforov 1993; Maquin and Ragot 2000]
- Analyse fréquentielle : dans cette approche, l'analyse de la densité spectrale du signal est effectuée. Cette densité peut changer à l'occurrence d'un défaut. Pour déterminer le spectre du signal on se sert en général de la Transformée de Fourier Rapide [Orfanidis 1996] plus de méthodes de comparaison avec de gabarits de fréquence de bon (ou mauvais) fonctionnement pour la décision de la détection [Garcia-Beltran, Leseq et al. 2003].

La méthode qui sera utilisée au chapitre 4 est une méthode particulière qui repose à la fois sur la connaissance qualitative (relations causales entre variables) et de la connaissance quantitative (équations numériques entre variables). Elle est fondée sur une représentation du procédé par un graphe causal [Montmain and Gentil 2000] dont les propriétés seront résumées dans ce chapitre.

Le problème de détection de défauts.

On peut poser le problème de la détection de défauts :

Définition 2-6

Soit un système observé représenté par le triplet $\langle DS, COMPS, OBS \rangle$ où **DS** est la description du comportement du système (sans défaut), **COMPS**, les composants du système et **OBS** un ensemble d'observations. Soit aussi **B** l'ensemble de toutes les paires entrée-sortie (u,y) des trajectoires possibles pour le système sans défaut et **C** le complément de **B**, et **B_{ref}**

l'ensemble de paires entrée-sortie (u,y) des trajectoires décrites par le modèle du système sans défaut

Le problème de détection est le choix d'une des interprétations suivantes:

-H0 : la différence entre les observations **OBS** et le comportement « calculé » du système \mathbf{B}_{ref} est non nulle à cause des bruits de mesure ou des erreurs de modélisation, et le fonctionnement est normal (Figure 2-8 point b) ;

-H1 : la différence est non nulle parce que le comportement d'un (ou plusieurs) composant $c \in \mathbf{COMPS}$ du système n'est plus dans \mathbf{B} (Figure 2-8 point c) ;

-H2 : la différence est nulle parce que le comportement de tous les composants $c \in \mathbf{COMPS}$ du système sont dans \mathbf{B} (Figure 2-8 point a) ;

-H3 : la différence est nulle à cause des erreurs de modélisation ($\mathbf{B}_{ref} \not\subset \mathbf{B}$), mais le comportement n'est pas normal (Figure 2-8 point d).

On peut expliquer ce problème avec la Figure 2-8. Le point « a » appartient à la zone \mathbf{B} de bon comportement, il a été ainsi modélisé, le point « c » est un comportement en défaillance. Le point « b » n'a pas été inclus dans la modélisation du bon comportement du système, et peut produire de faux indicateurs de défaut; finalement, le point « d » a été inclus dans le modèle de bon fonctionnement, en pouvant créer un manque à la détection.

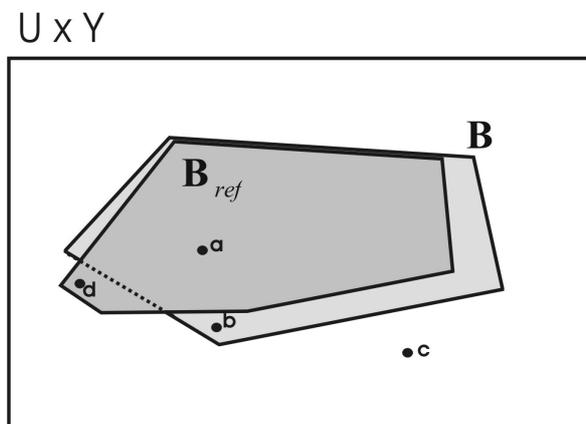


Figure 2-8. Illustration du comportement du système et du comportement de référence.

Quand on s'intéresse aux systèmes continus, la taille de l'espace de trajectoires est très grande. Il est donc préférable de réaliser un changement d'espace de représentation vers des espaces invariants. La représentation la plus courante est l'utilisation de l'espace de résidus. Dans le cas sans défaut, la valeur des résidus est faible (proche de zéro) indépendamment de la trajectoire. Dans le cas contraire, un résidu non négligeable indique une différence entre les mesures et la valeur calculée, qui indique l'existence d'un défaut.

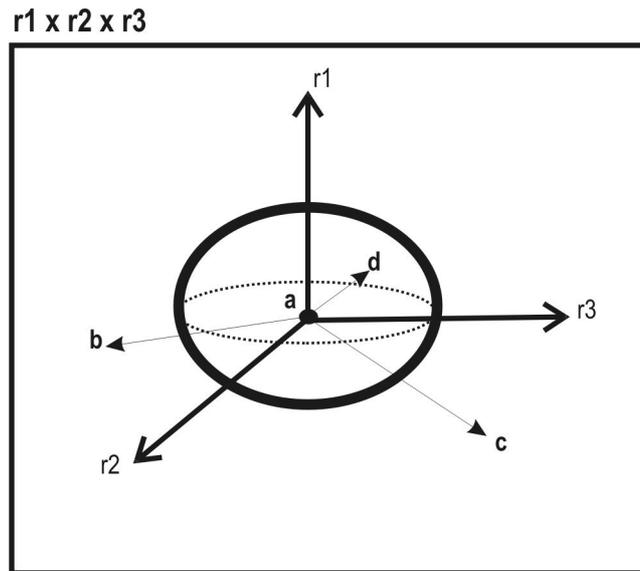


Figure 2-9. Espace de résidus.

2.3.2.2 Les méthodes de décision pour la localisation des composants défectueux.

Dans la suite, ce paragraphe présente les principales approches employées pour la localisation du composant en défaillance (et parfois l'identification). Dans le meilleur des cas, on dispose d'un modèle de défauts qui relie les causes et leurs conséquences. Dans la pratique, la disponibilité d'un tel modèle est difficile et longue à obtenir pour plusieurs raisons. Tout d'abord, il est difficile d'envisager toutes les causes possibles de défaillances d'un système, et quand bien même cela serait faisable, les causes étant connues, il est également difficile d'établir toutes les conséquences qui leur sont associées. Ensuite, même si l'on fait l'hypothèse simplificatrice de ne considérer qu'une partie des défaillances possibles du système, il est en général difficile, à partir des seules mesures ou résidus individuels, de remonter des conséquences aux causes : plusieurs causes différentes entraînent parfois des conséquences similaires, tout au moins sur les variables observées pour établir le diagnostic [Popieul 2003]. On établit ici le problème de localisation de défaillances :

Définition 2-7

Soit **COMP** l'ensemble de composants du système, le problème de localisation est celui de trouver le sous-ensemble minimal de composants anormaux **NOGOODS** tel que :

$\{AB(c) : c \in \text{NOGOODS}\} \cup \{\neg AB(c) : c \in \text{COMP} \setminus \text{NOGOODS}\} \cup \text{DS} \cup \text{OBS}$
est consistant.

Où $AB(c)$ représente l'assertion que le comportement du composant est anormal.

De la même façon que pour la détection de défauts, il existe diverses méthodes pour la localisation (identification) des défaillances :

- *Méthodes fondées sur les systèmes experts :*

Les systèmes experts constituent une approche possible pour le diagnostic, en particulier lorsque les observations sont de type symbolique. Le principe de fonctionnement d'un système expert s'articule autour de deux éléments principaux. La représentation de connaissances de surface qui ont trait au mauvais fonctionnement du système sous forme de règles permet d'effectuer automatiquement des opérations de diagnostic. Le cas le plus utilisé est l'utilisation de l'arbre de défaillances qui répertorie l'ensemble des défaillances, du général au particulier. Un cheminement dans l'arbre permet, à partir des observations, d'incriminer un composant (ou des composants) situés au niveau des feuilles de l'arbre. La traduction de cette connaissance sous forme d'une base de règles permet une exploitation dans un système à base de connaissances par des mécanismes de chaînage avant/arrière/mixte [Frydman, LeGoc et al. 2001].

- *Méthodes fondées sur la modélisation structurelle :*

Dans cette approche, la représentation du système est réalisée par un modèle de sa structure (composé de composants connectés). Le comportement du système est déduit par composition des différents comportements des composants. Ces méthodes peuvent être vues comme une extension des méthodes de détection de défauts basées sur des équations de parité ou sur des observateurs mais utilisées pour la localisation de défaillances. Dans les deux cas, les équations de parité, ou les observateurs, sont conçus de telle sorte que les indicateurs de défauts, dans ces deux cas, les résidus, soient structurés, c'est-à-dire qu'en la présence d'une certaine défaillance, seulement un sous-ensemble d'entre eux ne sont pas satisfaits, ce qui permet en même temps de reconnaître le composant défectueux [Cassar, Cocquempot et al. 1994; Chen and Patton 1995].

- *Méthodes de classification - reconnaissance des formes.*

On utilise la classification de données pour reconnaître un objet, une « forme », en la comparant à divers prototypes représentatifs de classes d'objets : on conclut que la forme appartient à la classe définie par le prototype le plus ressemblant. On peut donc utiliser ces méthodes pour classer des formes caractéristiques de diverses situations de défaillance [Jämsä-Jounela, Vermasvuori et al. 2003]. Cette méthode est particulièrement intéressante si on ne connaît pas de modèle de l'installation, mais si on a un ensemble de données très complet de son comportement défaillant. Résoudre un problème de diagnostic par classification se décompose en plusieurs étapes : 1) définir l'espace de représentation, c'est à dire choisir des attributs qui donnent un espace de dimension minimale et des classes facilement localisables 2) définir l'espace de décision (apprentissage des classes) et 3) en temps réel, affecter une forme à une classe.

2.3.3 Aide à la décision pour la conduite du système face à des fonctionnements anormaux.

Les opérateurs qualifiés connaissent généralement la décision appropriée à prendre après qu'un défaut a été clairement identifié. L'objectif d'un système intelligent de supervision est d'aider les opérateurs, dans les situations critiques, pour l'analyse d'un procédé que

l'automatisation a rendu de plus en plus complexe, mais l'objectif n'est pas de le décharger des tâches de prise de décision.

Le problème principal de l'opérateur vient de la difficulté de lier les comportements corrélés des variables (par exemple dans les boucles de régulation ou quand les retards entre les variables sont très significatifs). Dans une situation stressante ou devant des alarmes en cascade (phénomène de sapin de Noël), ou simplement quand l'opérateur est fatigué, même un opérateur qualifié peut faire une erreur, en utilisant une action inappropriée parce qu'il n'a pas bien interprété la source précise de tous les problèmes qu'il observe, ou en utilisant l'action dans le mauvais sens (par exemple, en ouvrant une vanne qui devrait être fermée).

L'aide à l'opérateur devrait principalement consister à détecter et à évaluer l'état du procédé, et à construire une liste de moyens d'action potentiels concernant la situation. La situation est différente si :

- la tâche de détection de défaut montre la présence d'une alarme prioritaire inattendue. Ce genre d'alarme doit être pris en considération automatiquement par les procédures de secours. Un module préprogrammé groupe toutes les actions d'urgence à prendre. L'intervention humaine ne joue aucun rôle « intelligent » de surveillance, mais d'action. Comme on a vu dans la section 2.2.3, la séquence de tâches est conçue dans la phase de conception/validation comme un moyen d'atteindre la sécurité du système
- une ou plusieurs variables sont hors de leur zone nominale. Ceci peut arriver par exemple à cause d'un défaut de dérive ou d'une perturbation lente. Afin d'empêcher la dégradation de l'évolution du procédé et/ou d'éviter des procédures lourdes de reprise de secours, c'est approprié de réduire au minimum ou même supprimer ces déviations. Après que l'opérateur humain a été informé de l'état du procédé, on lui fournit des conseils d'action afin de faire face à la situation. Si un modèle précis du procédé est disponible, il peut être intéressant de faire une prédiction du comportement du procédé afin de montrer à l'opérateur comment le problème se développera à l'avenir si aucune action n'est faite ou après certaine décision d'action. C'est particulièrement vrai si la détection de défaut et la localisation ont détecté une dérive lente.

Faisons la distinction subtile entre le conseil d'action et la commande. Dans le cas d'un système d'aide à la supervision, l'opérateur est toujours maître de la situation. Le système d'aide à la décision informe et conseille, mais jamais n'agit. De ce fait le système ne calcule pas une valeur précise, mais indique plutôt quel moyen d'action doit être employé. Par exemple, il pourrait conseiller de changer un point de référence d'une boucle de commande parce qu'un capteur a une dérive ou de changer la commande manuelle parce que le régulateur travaille hors de son domaine de stabilité. Mais il n'est pas censé trouver le point de réglage ou la valeur précise de l'action. Ceci a besoin d'une étude plus soignée de validation, qui est toujours un problème de recherche.

Le chapitre 5 présentera des outils, complémentaires des outils de diagnostic et utilisant la connaissance présente dans le graphe causal, pour aider l'opérateur à bien choisir les actions pertinents permettant d'arrêter l'effet des défauts.

2.3.4 Aide à la décision pour la maintenance de procédés.

En considérant la Figure 2-3, on peut constater que un grand nombre des défaillances sont gérées par les stratégies de maintenance. Malheureusement, cette discipline n'est pas toujours considérée comme un domaine de recherche, comme l'établit Zwingelstein :

« Considérée presque toujours comme un poste inévitable de dépenses dans les entreprises, la maintenance a aujourd'hui une image de marque défavorable. Assimilé à des actions de dépannage ou d'entretien, le rôle de la maintenance est rarement considéré comme une activité stratégique au sein des entreprises. De plus, comme la plupart des audits de maintenance le démontrent, les services de maintenance font figure de parents pauvres par rapport aux services de production...Peu ou pas reconnue comme une discipline scientifique à part entière, elle n'a incité que très peu de chercheurs et d'ingénieurs à approfondir ces domaines » [Zwingelstein 1996].

2.3.4.1 Définitions.

La maintenance est définie comme l'ensemble des activités, destinées à maintenir ou à rétablir un composant dans un état donné de sûreté de fonctionnement, pour accomplir une fonction requise. Ces activités sont une combinaison d'activités techniques, administratives et de gestion. De façon générale, on peut parler de quatre types différents de stratégies de maintenance :

- La maintenance corrective (MC): Ensemble des activités réalisées après la défaillance du bien, ou la dégradation de sa fonction pour lui permettre d'accomplir une fonction requise, au moins provisoirement : ces activités comportent notamment la localisation de la défaillance et son diagnostic ; la remise en état avec ou sans modification, le contrôle du bon fonctionnement.
- La maintenance préventive (MP): La maintenance ayant pour objet de réduire la probabilité de défaillance ou de dégradation d'un bien ou d'un service rendu. Les activités correspondantes sont déclenchées selon un échéancier établi à partir d'un nombre prédéterminé d'usage et/ou des critères prédéterminés significatifs de l'état de dégradation du bien ou du service.
- La maintenance prédictive conditionnelle (MC): Maintenance préventive subordonnée à l'analyse de l'évolution surveillée de paramètres significatifs de la dégradation du bien, permettant de retarder et de planifier les interventions

2.3.4.2 La maintenance conditionnelle.

Les systèmes d'aide à la maintenance les plus récents sont fondés sur les concepts de la maintenance conditionnelle, dans lequel le remplacement ou la réparation des composants devrait être basés sur le vrai état du composant et leur prédiction, et pas sur une politique systématique définie a priori. Aujourd'hui, la MC est utilisée conjointement avec les

approches traditionnelles de la maintenance corrective et de la maintenance préventive. Les facteurs qui ont conduit à une augmentation de l'utilisation de la MCP incluent le besoin de :

- Réduction de coûts d'entretien et logistiques,
- Augmentation de la disponibilité de l'équipement,
- Protection contre les défauts de composants critiques.

Une architecture complète pour des systèmes de MC devrait couvrir la gamme des fonctions de l'acquisition de données jusqu'aux recommandations d'actions spécifiques de maintenance. Les fonctions clef qui facilitent la MC incluent [Lebold and Thurston 2001] :

- a. Instrumentation et acquisition de données.
- b. Traitement des signaux et extraction de caractéristiques.
- c. Production d'alarmes ou d'alertes.
- d. Système de Diagnostic.
- e. Prédiction : projection des profils de santé à la future santé ou évaluation de la vie utile restante.
- f. Aide à la décision : recommandations de tâches de maintenance, ou évaluation de la promptitude de remplacement des composants pour un scénario opérationnel particulier.
- g. Gestion des flux de données.
- h. Gestion des séquences des tests fonctionnels du système.
- i. Gestion de stockage de données historiques et d'accès à ces données.
- j. Gestion de la configuration (opération) du système.
- k. Interface avec l'opérateur humain.

Comme les points *a*, *b*, *c*, et *d* sont aussi utilisés pour le système d'aide à la conduite et à la supervision, on se focalisera sur les aspects liés à l'aide à la décision dans la maintenance. L'importance des interfaces correctes pour l'opérateur humain sera discutée dans la section 2.5, et la gestion de données dans la section 2.4.

Dans le cadre de la *maintenance conditionnelle préventive*, on peut envisager un groupe de tâches destinées à maintenir l'opération des composants [Zwingelstein 1996]:

- Petit entretien.
- Contrôle.
- Essais ou test opérationnels.
- Essais ou test fonctionnels.
- Test pour la recherche de défaillances cachées.
- Surveillance en service.
- Inspection.
- Essais en exploitation.
- Visite.
- Révision.
- Modification.
- Réparation (en fait cette tâche correspond à la maintenance corrective).

La gestion des flux de données et des séquences des tests fonctionnels du système ont pour objectifs de vérifier que le matériel concerné remplis sa fonction avec ses caractéristiques associées dans des conditions données d'exploitation. En présence des

systèmes redondants, et en particulier pour les matériels redondants qui ne sont pas actifs, il est impératif de procéder à des essais fonctionnels particuliers. Ces essais permettent de s'assurer qu'en cas de défaillance du matériel en marche, les matériels redondants seront aptes à remplacer la fonction perdue par le matériel défaillant.

Le chapitre 5 proposera l'implémentation de conseils de maintenance à l'opérateur. Les conseils pourront être reliés soit aux symptômes actuellement observés sur le procédé, soit au pronostic de l'évolution de celui-ci sous l'effet de défauts en train de s'établir.

2.3.5 Le pronostic.

Le but principal du pronostic est d'anticiper et d'empêcher le développement de défauts critiques. Les résultats du pronostic sont employés pour la prise de décisions proactives sur l'opération ou la maintenance avec le but économique de maximiser la durée de vie des composants remplaçables, tout en réduisant au minimum des risques opérationnels [Mathur, Cavanaugh et al. 2001]. Le pronostiqueur est vu comme un prédicteur qui reçoit des données, et aide l'opérateur à déterminer la fenêtre permise de temps pendant laquelle l'entretien de machine doit être exécuté ou l'opération actuelle du système peut être maintenue de sorte que l'intégrité du processus puisse être garantie autant que possible. Deux genres de pronostics peuvent être envisagés comme outils pour l'opérateur.

Les politiques de maintenance et d'opération pourraient être aidées par le pronostic de l'évolution du défaut dans le temps, pour éviter un changement prématuré des composants encore fonctionnels, de ce fait permettant à l'opérateur de projeter la politique d'entretien (exécuter l'entretien seulement si nécessaire). Ce raisonnement est basé sur une prédiction des séries temporelles de l'évolution des symptômes et variables critiques qui sont fournis par les agents de diagnostic. La prévision de la manière dont un défaut pourrait se développer est reliée avec la prévision de la valeur des variables critiques ou des symptômes. Avec une connaissance plus profonde (physique, experte ou statistique) sur les composants, le temps restant avant le défaut ou la vie utile restante d'un composant pourrait être prévu.

Il existe de nombreuses approches de pronostic, la gamme de ces approches commence de simples modèles de taux de défaillance aux modèles physiques de haute précision [Propes, Lee et al. 2002], (**Figure 2-10**):

- a. Algorithmes généraux de traitement de signal : algorithmes de faible complexité, grande applicabilité, basés sur l'expérience.
- b. Identification de modèles, logique floue, réseaux de neurones, modèles d'estimation.
- c. Modèles physiques : Algorithmes de précision élevée, mais avec une complexité et un coût élevés.

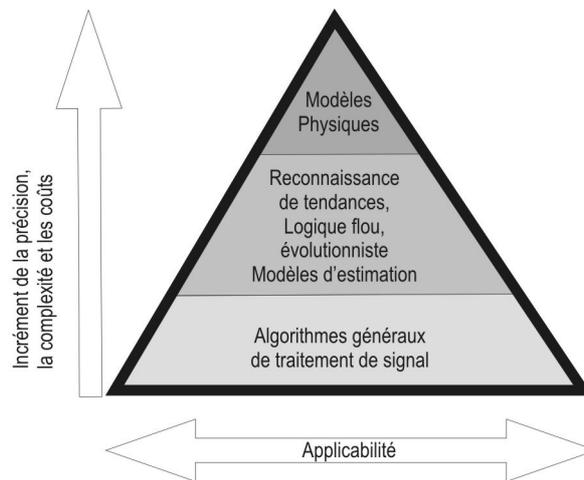


Figure 2-10. La gamme d'applicabilité des algorithmes de prédiction.

Algorithmiquement le pronostiqueur est un genre de prédicteur dynamique de séries temporelles. Ainsi, le but du pronostiqueur est de prévoir les valeurs futures d'une variable en utilisant ses valeurs passées. Le terme "dynamique" est lié à la condition fonctionnelle que la variable cible est dynamiquement mise à jour quand plus d'information est fournie [Vachtsevanos and Wang 2000]. Dans un problème de prévision de séries temporelles, le but principal est l'estimation de la valeur future $y(t+th)$ d'un variable y , où $th > 0$ se rapporte à l'horizon de temps. Pour les valeurs relativement grandes de th nous parlons de la prévision à long horizon. La prédiction est calculée à partir des valeurs courantes et passées $y(t)$, $y(t-1)$, $y(t-2)$... jusqu'à la taille de la fenêtre disponible au temps t , donnant une estimation de la variable $y(t+th)$.

Plusieurs méthodes ont été développées pour fournir cette sorte de prédiction. Les méthodes classiques sont basées sur les modèles ARIMA [Box, Jenkins et al. 1994]. La prévision peut également être basée sur un réseau de neurones. L'analyse de tendance est une autre manière de faire (Chapitre 5).

En cas de défaut, un autre outil pour la décision est le pronostic sur la propagation possible des effets du défaut dans le procédé. Néanmoins, cette connaissance peut être tout à fait incertaine parce que le défaut a modifié le comportement normal du système. Le chapitre 5 introduit une méthode fondée sur un graphe causal pour la prédiction des effets de défaut.

2.3.6 Les interfaces graphiques.

Les interfaces homme-machine jouent un rôle très important dans les systèmes d'aide à la décision car elles représentent le seul point d'interaction entre l'opérateur et les algorithmes d'aide à la décision. Dans les situations anormales, le rôle de l'opérateur est de détecter avec exactitude, de diagnostiquer et de prendre l'action de reprise et/ou de maintenance appropriée en utilisant son expérience et ses habiletés et les outils à sa disposition.

Il est important de souligner que les opérateurs humains raisonnent en termes qualitatifs, c'est pourquoi il est important que même si les algorithmes implantés utilisent des concepts quantitatifs, l'interface graphique puisse effectuer, dans certains cas, une "traduction"

des résultats vers des termes qualitatifs. L'utilisation de codes de couleur est un exemple de ce type de traduction [Evsukoff 1998].

2.4 L'implémentation des systèmes d'aide à la décision.

Les approches traditionnelles de développement des systèmes de supervision et de commande sont concentrées sur les aspects fonctionnels et de temps réel. Quand ces systèmes deviennent plus complexes, des approches systématiques de développement sont nécessaires pour gérer la complexité. Comme Sanz le précise [Sanz, Pfister et al. 2001], le développement de logiciels pour les systèmes de commande (et de supervision) doit se concentrer également sur la qualité du logiciel, avec des concepts tels que la cohésion, la qualité, la réutilisation et la modularité.

Vu la diversité de tâches en rapport avec la supervision et la maintenance, un des aspects principaux dans son développement est la modularité du système, qui permet d'obtenir aussi de hauts degrés de sûreté de fonctionnement de ce système.

2.4.1 La modularité vue comme la répartition d'expertise dans un système de supervision avancé.

Une évolution importante dans l'informatique est la conception et le développement modulaire exemplifiés par l'utilisation des objets ou des composants logiciels. Les approches modernes dans la construction de système basent leur processus de conception sur l'identification de fonctionnalités et/ou responsabilités et sur l'encapsulation des services sous la forme d'objets ou de composants pour les implanter. Toutefois les systèmes de commande, de surveillance et de supervision sont des entités qui échappent au modèle de base de l'objet. Quand les objets ont des responsabilités autonomes, ils deviennent des objets actifs ou agents [Brungali 1998].

Les modèles d'agent fournissent le meilleur cadre pour la conception des systèmes modulaires et actifs de la commande et la supervision. Dans le cadre de multiples agents, les agents agissent les uns sur les autres pour atteindre un certain objectif fournissant une base pour la résolution distribuée des problèmes. Les agents, comme les objets et composants, peuvent être faits avec des capacités de réutilisation, simplifiant le processus de la construction de systèmes [Sanz, Segarra et al. 1999]. De cette façon, les agents autonomes et les systèmes multi-agents représentent une nouvelle manière de concevoir, d'analyser et d'implanter les logiciels complexes [Demazeau and Briot 2001].

2.4.1.1 Définition d'agents et systèmes multi-agents.

Un *agent* est une unité abstraite, située dans un certain environnement, capable d'exécuter une action de façon autonome afin de répondre à ses objectifs de conception. L'agent est informé sur son environnement qui peut agir sur lui. L'agent commande ses actions et son état interne d'une manière flexible ; il peut aussi agir sur autres d'agents si c'est approprié. Un agent logiciel se distingue d'un objet informatique ou d'un programme

classique par sa capacité à agir sans qu'il ait été explicitement sollicité. C'est pour ça qu'on dit que les agents sont « autonomes » [Mandiau 2000a]. Pour atteindre sa flexibilité l'agent doit être capable de répondre à temps, d'exhiber un comportement proactif et opportuniste et d'être capable d'interagir avec les autres agents logiciels et avec les humains [Chaib-draa, Jarras et al. 2001].

Les capacités de l'agent sont de trois ordres qui répondent aux trois phases générales de réalisation d'une tâche [Mandiau 2000b] :

- une phase de perception : l'agent perçoit son environnement et met à jour ses représentations internes de l'environnement et des autres agents.
- une phase de cognition : l'agent détermine ce qui est à faire (sa tâche) et décide quand et comment le faire.
- une phase d'action : la réalisation effective des actions qui ont été décidées.

Les agents sont divisés en deux catégories [Demazeau and Briot 2001]:

- Les **agents réactifs** sont de nombreux agents à grain fin, simples, non intelligents et sans représentation de leur environnement. Combinés dans un système multi-agent, les agents réactifs sont intimement couplés, ils agissent l'un sur l'autre en utilisant un type de comportement stimulus-réponse. Une solution du problème pourrait être trouvée par des effets secondaires, et le comportement intelligent « global » émerge à partir de l'interaction des agents.
- Les **agents cognitifs**, au contraire, sont les agents moyens ou à grain brut, ils sont intelligents avec une représentation explicite de leur environnement. Ils sont dotés de connaissances, d'habiletés, d'intentions et de plans, ce qui leur permet de coordonner leurs actions pour la résolution des problèmes.

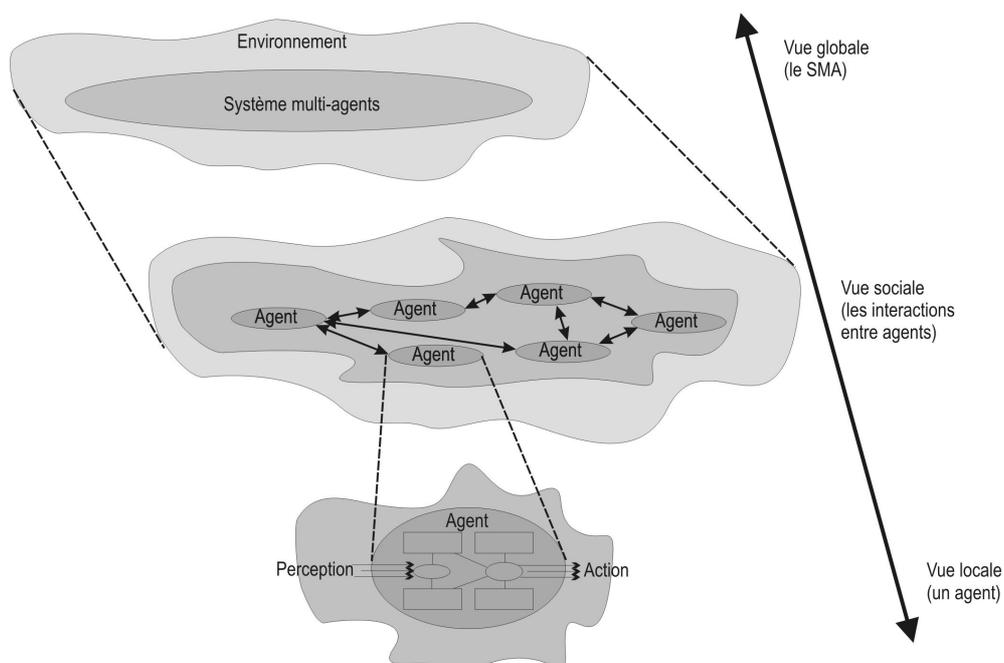


Figure 2-11. Système multi-agents vu selon différents niveaux de détail.

Un système multiagents est défini comme une collection d'agents autonomes résolvant les problèmes localement et fonctionnant ensemble vers la résolution d'un problème global (Figure 2-11). Un agent est défini comme entité autonome physique ou informatique, capable de représenter sa propre connaissance, de prendre des décisions et d'avoir une relation réciproque avec son environnement [Khoualdi and Dumas 1994].

2.5 Conclusions.

Le problème de la sûreté de fonctionnement des systèmes est vaste, c'est pourquoi les techniques reposent sur des connaissances diverses et varient considérablement entre elles. Les systèmes de supervision avancés évoluent de telle façon qu'ils combinent la technologie algorithmique avec des techniques d'intelligence artificielle afin de fournir une meilleure coopération entre les humains et le système. Les systèmes de surveillance avancés doivent analyser et interpréter ainsi de grandes quantités de données, en appliquant une grande diversité de méthodes, chacune adaptée à la situation des sous-systèmes.

Afin de coordonner l'application de cette diversité de techniques, la construction de systèmes de commande et de supervision avancés doit considérer l'utilisation de techniques de développement de logiciels intrinsèquement "sûrs", afin d'éviter une augmentation de la complexité du système.

Chapitre 3

Le système MAGIC

3.1 Introduction

Dans le chapitre antérieur, toute la problématique théorique associée aux systèmes avancés de supervision a été présentée. À cet égard, le but principal du système MAGIC⁵ est de fournir à l'opérateur une information claire, compréhensible et détaillée selon les besoins, sur la cause d'une situation anormale et de lui suggérer des actions de remédiation ou la reconfiguration appropriée du système [Köppen-Seliger, Marcu et al. 2003].

Étant donnée la complexité du système MAGIC, seules ses principales caractéristiques seront décrites ici. Les fonctionnalités et interactions des agents de diagnostic et plus précisément celui qui est fondé sur le graphe causal et de l'agent d'aide à l'opérateur seront expliquées de manière exhaustive dans les chapitres 4 et 5.

3.2 Les fonctionnalités générales de MAGIC

Pour accomplir sa mission d'aide à l'opérateur, différentes fonctionnalités ont été implémentées dans MAGIC (**Figure 3-1**). Une de ses principales fonctionnalités, sinon le

⁵ Multi-Agents-Based Diagnostic Data Acquisition and Management in Complex Systems

cœur du système, est la détection en ligne et le diagnostic précoce de défauts abrupts ou de défauts qui se développent lentement dans les systèmes complexes. En détectant les défauts avant qu'ils ne se développent entièrement, l'opérateur peut planifier la conduite de l'installation ou encore le système de contrôle automatique ou le procédé peuvent être reconfigurés pour compenser le défaut. De ce fait, non seulement la continuité de la production peut être assurée mais également la qualité du produit. L'identification précoce des conditions potentiellement défectueuses fournit les informations clef pour l'application des politiques de maintenance prédictive. Ainsi l'inspection, la réparation ou le remplacement des composants peuvent être programmés dans les périodes de maintenance préfixées. En conséquence, le temps d'arrêt de l'installation est réduit au minimum, la production est maintenue, et l'utilisation du personnel d'entretien optimisée, maximisant de ce fait la disponibilité du procédé et son efficacité.

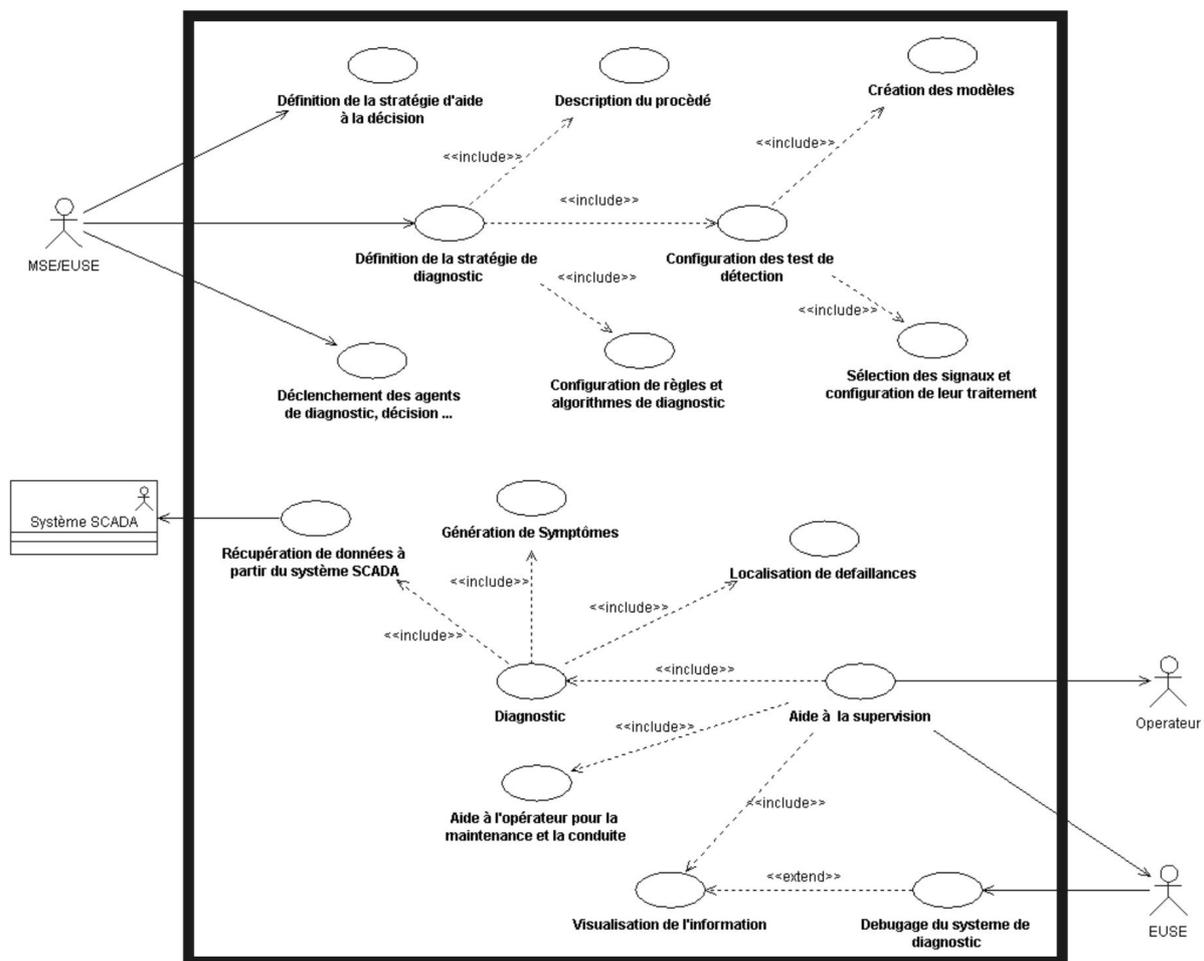


Figure 3-1. Cas d'utilisation de MAGIC

Comme déjà mentionné dans le chapitre 2, la diversité et l'hétérogénéité des tâches associées à un système de supervision avancée exige que le logiciel qui supporte ces tâches soit du type distribué. Le système MAGIC est subséquemment fondé sur une architecture multi-agents basée sur le concept de multi-niveaux – multi-agents (MAML, par *Multi Agent Multi Level*), dans lequel les tâches complexes de diagnostic et d'aide à l'opérateur sont distribuées sur un nombre d'agents « intelligents ». Chacun de ces agents développe sa tâche individuelle d'une manière presque autonome et l'ensemble des agents communiquent entre eux par l'intermédiaire d'une infrastructure de communication inter-agents. Dans ce contexte,

les "agents" sont des composants logiciels fonctionnant, soit sur le même ordinateur, soit séparément dans plusieurs ordinateurs, coopérant les uns avec les autres, ayant chacun une tâche spécifique. Ils forment une architecture d'éléments distribués fonctionnellement, sémantiquement et dans l'espace, ayant pour résultat un système complet mais modulaire .

En général l'outil MAGIC peut être appliqué dans n'importe quel secteur industriel où la surveillance, l'acquisition de données et le diagnostic sont nécessaires pour assurer un comportement fiable et sûr du procédé. MAGIC est un système configurable de sorte que l'utilisateur final puisse choisir la quantité et la nature des éléments (agents) nécessaires pour les tâches de supervision ; à cet égard, et faisant partie des fonctionnalités de MAGIC, le système propose certains outils pour la sélection des méthodes les plus appropriées pour le diagnostic en fonction du type de procédé. L'utilisateur final pourra aussi, dans la mesure du possible, utiliser une configuration d'agents déjà existante, pour son application à un autre procédé industriel semblable (même type unités et composants physiques); évidemment, ceci pourra nécessiter des modifications inhérentes au comportement particulier de chaque composant.

3.3 L'architecture générale de MAGIC

L'architecture distribuée de MAGIC est fondée sur le concept multi-niveau par rapport à ses fonctionnalités, (Figure 3-2). Ces fonctionnalités sont distribuées sur 5 niveaux:

- Niveau 1 : spécification du procédé
- Niveau 2 : acquisition de l'information
- Niveau 3 : génération de symptômes
- Niveau 4 : décision de diagnostic
- Niveau 5 : aide à l'opérateur
- Niveau de gestion du système.

3.3.1 L'approche multi-niveaux de MAGIC

En référence au concept Multi-Agents Multi-Niveaux (MAMN), illustré sur la Figure 3-2, les différents agents de MAGIC ont les tâches décrites ci-dessous :

- **Niveau 1 : spécification du procédé :**

PSA : l'agent de Spécification de Procédé (PSA) a les tâches de caractérisation de la topologie du procédé à surveiller et de la prescription et configuration des divers agents de MAGIC.

GPMAS : Rigoureusement le GPMAS (Serveur d'algorithmes mathématiques génériques) 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 algorithmes adéquats pour la modélisation des comportements des composants et des unités physiques, ainsi que pour la caractérisation des signaux utiles pour le diagnostic.

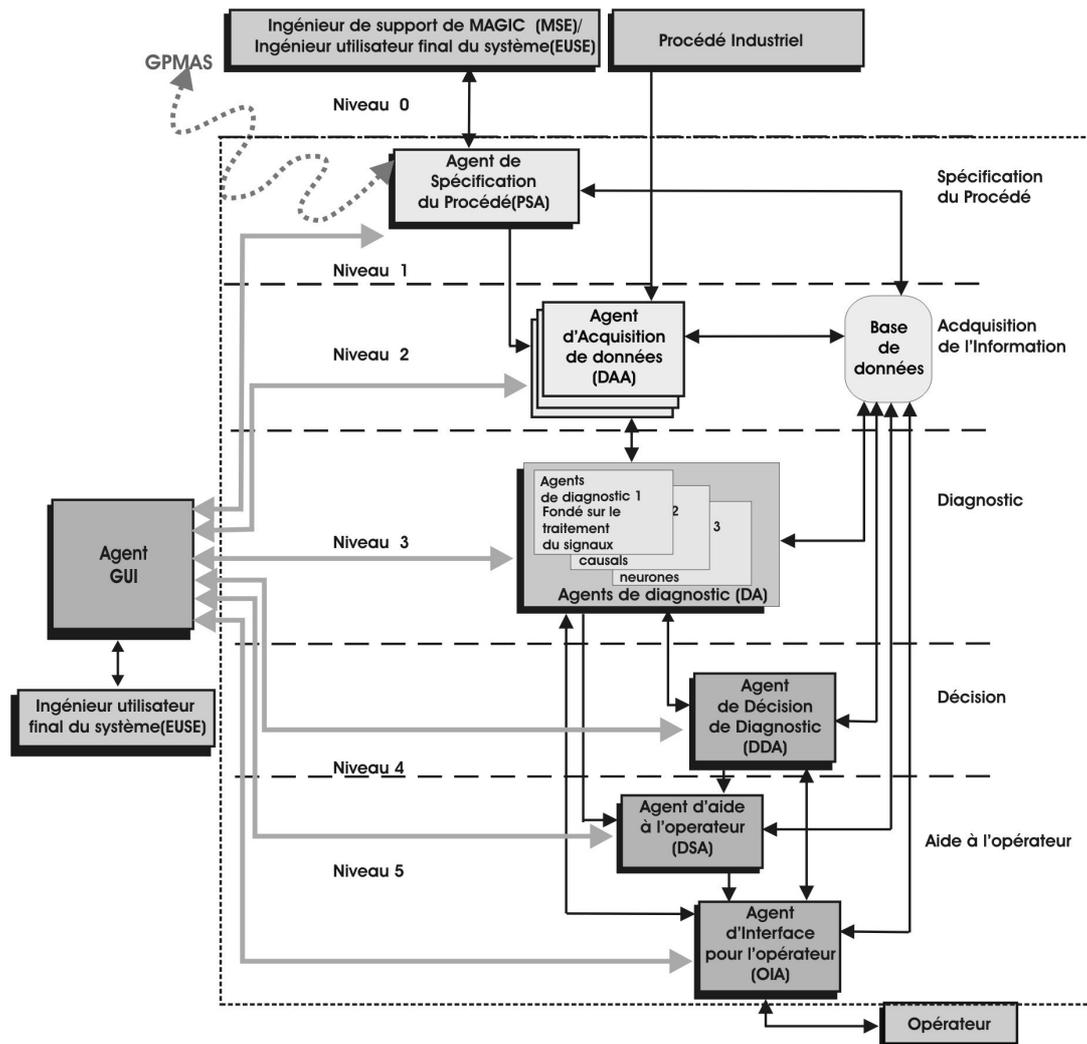


Figure 3-2. L'architecture générale de MAGIC

- **Niveau 2 : acquisition de l'information :**

DAA : Dans le cadre de MAGIC, chaque agent d'acquisition de données (DAA) est la passerelle entre le système MAGIC et l'application industrielle. Il est responsable de la gestion des données du procédé récupérées à partir des capteurs et des actionneurs (par l'intermédiaire du système SCADA). Cet agent est le seul qui joue un rôle à la fois dans la phase de customisation/configuration et dans la phase de fonctionnement continu, puisque ces deux phases dépendent des données acquises sur le procédé.

Base de Données : La base de données de MAGIC est employée pour stocker et rechercher toute information nécessaire pour le fonctionnement de MAGIC (information obtenue du procédé et aussi toute information générée par le système). La base de données stocke les signaux venant du procédé, toute l'information de configuration pour chacun des agents de MAGIC, les messages de communication inter-agents, la description structurée du procédé et les messages générés pour ou par l'opérateur.

Ce niveau transmet les données présentes et passés si nécessaire aux autres niveaux.

- **Niveau 3 : génération de symptômes :**

DA : Les agents de diagnostic (DA) participent au processus de diagnostic global en effectuant les "tests de détection de défauts" et en produisant des symptômes qui seront examinés plus tard par le DDA. Plusieurs agents de diagnostic peuvent fonctionner en parallèle ou être déclenchés dans un ordre préétabli. Ils peuvent aussi fonctionner de façon complètement indépendante sur des composantes différentes.

Divers algorithmes de détection ont été choisis pour être intégrés dans le système MAGIC. Cette sélection a été basée sur une étude comparative des différentes méthodes existantes [Atlas, Lacatusu et al. 2003]. Néanmoins, d'autres algorithmes peuvent être intégrés si les structures d'information de MAGIC sont respectées. Actuellement les algorithmes mis en oeuvre sont les suivants:

- Méthodes fondées sur la théorie du traitement du signal
 - Méthodes statistiques
 - Méthodes dans le domaine du temps
 - Méthodes dans le domaine de la fréquence
- Méthodes fondées sur les modèles analytiques
 - Banques d'observateurs
 - Modèles causaux
- Méthodes fondées sur des modèles générés à partir de données
 - Réseaux de Neurones basés sur des liens fonctionnels dynamiques
 - Réseaux de Neurones Multicouches
 - Méthodes d'identification de représentation d'état

Ce niveau élabore des symptômes qui sont traités dans le niveau suivant pour localiser les défauts.

- **Niveau 4 : décision de diagnostic :**

DDA : Dans ce niveau, un agent de décision de diagnostic (DDA) propose une décision de diagnostic finale à partir des résultats des DAs. Ce diagnostic correspond à la localisation du composant défectueux.

Deux méthodes ont été implémentés à ce niveau. Une table de signature, outil classique des automaticiens, est construite pour analyser en ligne l'ensemble des symptômes. Une approche classique en intelligence artificielle et fondée sur un raisonnement logique a aussi été implémentée.

- **Niveau 5 : Aide à l'opérateur :**

DSA : L'objectif de l'agent d'aide à la décision (DSA) est de fournir des conseils à l'opérateur afin de l'aider dans les tâches de surveillance et d'action. Comme cela sera expliqué dans le chapitre 5, ces conseils ne comprennent pas la mise en marche ni l'arrêt d'urgence normalement effectuées par le système de contrôle/commande classique, ni l'accommodation des régulateurs. Il s'agit plutôt, de conseils de maintenance pour la réparation des composants endommagés ou de conseils d'action pour l'aide à la conduite.

OIA : Le système MAGIC est équipé d'une interface implantée dans l'agent d'interface pour l'opérateur (**Figure 3-3. Interface OIA.**).

Cet niveau est chargé de toute la communication homme/machine.

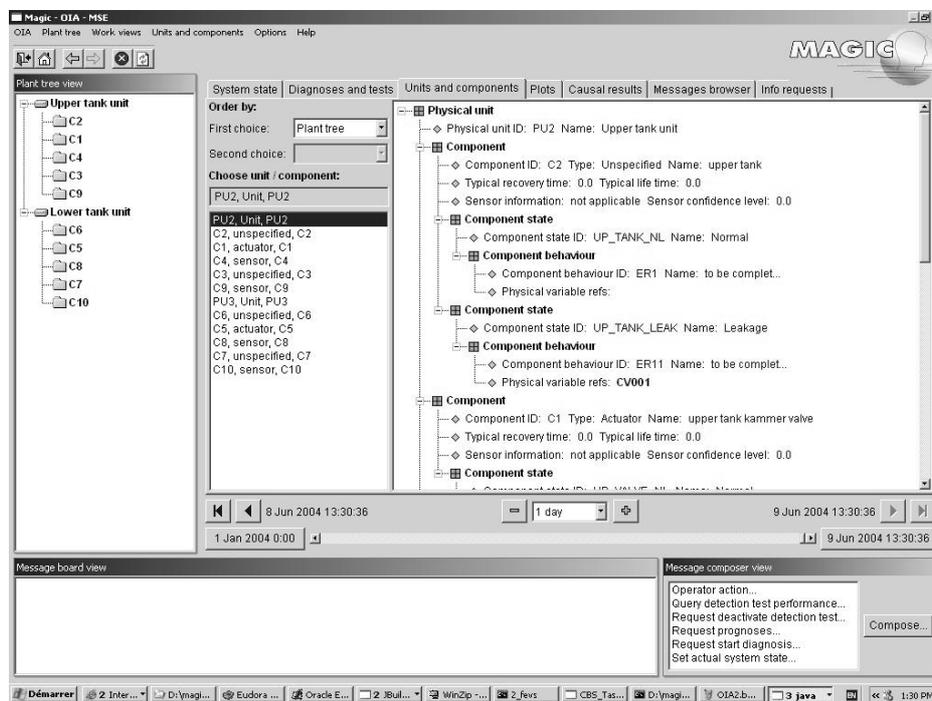


Figure 3-3. Interface OIA.

- **Niveau de gestion du système :**

GUIA : Hors de la hiérarchie fonctionnelle de MAGIC, l'agent GUI (*Graphical user Interface*) est en charge de la gestion du système au niveau informatique. L'agent GUI a la capacité de déclenchement des agents, l'interrogation de leur état, et leur arrêt. Grâce à l'agent GUI le réseau d'agents peut être surveillé, commandé et configuré, même à distance (Figure 3-4). Le GUIA entre en opération principalement pendant la procédure de démarrage du système MAGIC, pendant son arrêt et toutes les fois que l'exécution appropriée des divers agents doit être supervisée.

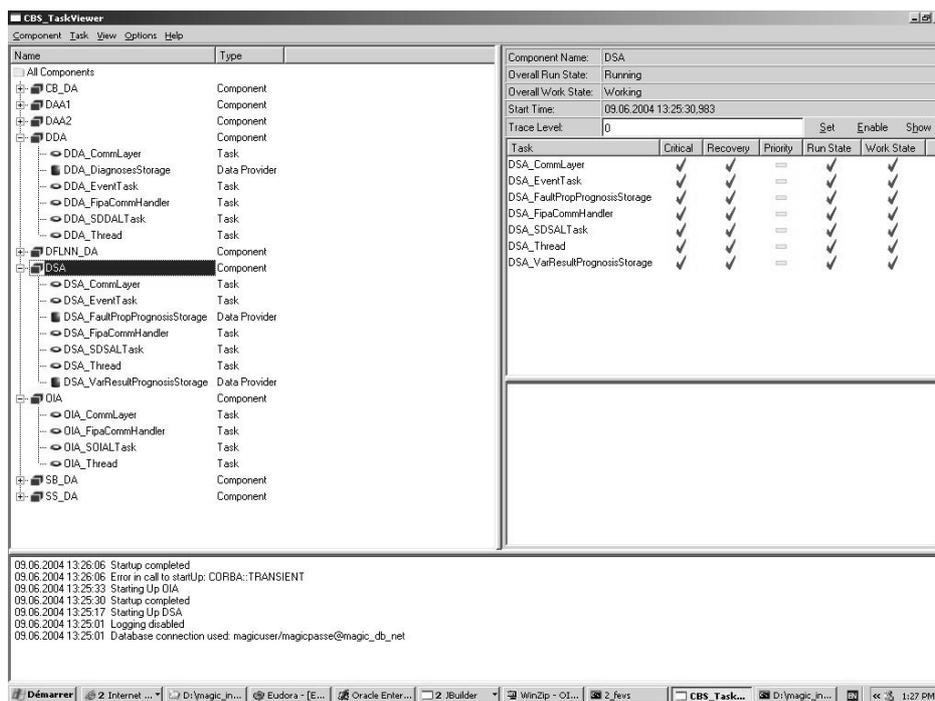


Figure 3-4. Interface GUIA.

3.3.2 Les deux phases de fonctionnement de MAGIC

Pendant les étapes de *configuration* et *personnalisation*, seules les fonctionnalités implémentées dans les niveaux 1, 2 et de gestion 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.

3.3.2.1 Les modes de configuration et de personnalisation.

Les phases de customisation et de configuration permettent de sélectionner les fonctionnalités en accord avec l'application industrielle [MAGK001R01 2002].

La personnalisation concerne la description du procédé (composants, topologie) selon la sémantique adoptée dans l'ontologie et l'adoption de la stratégie de diagnostic et d'aide à l'opérateur. Ceci couvre en particulier la définition des différents modèles utilisés pour la détection.

La configuration couvre tous les ajustements de paramètres en particulier ceux des modèles et de valeurs opérationnelles.

Grâce à la modularité de MAGIC, l'utilisateur final peut choisir entre les divers types d'agents proposés, ceux qui s'adaptent le mieux aux besoins du procédé [Atlas, Lacatusu et al. 2003; MAGD001S04]. Les tâches de customisation et configuration initiale peuvent être classées grossièrement en trois rubriques :

- i. Caractérisation du procédé :
 - La structuration de l'application industrielle : cette tâche a pour objectif d'implémenter la description du procédé industriel sous une forme hiérarchique. De cette façon, un procédé industriel est composé d'unités physiques et chaque unité physique est constituée de composants physiques. Un composant physique est alors une partie indivisible de l'application industrielle.
 - La caractérisation des variables physiques impliquées dans l'application industrielle: l'implémentation d'un système de diagnostic automatique a besoin des mesures des diverses variables du procédé pour la génération de symptômes. En conséquence, la caractérisation des variables physiques permet d'établir quelles variables physiques sont disponibles, sur quels composants, avec quelles caractéristiques (bruit, précision de capteurs, temps d'échantillonnage etc.).
 - La description des états physiques: cette tâche représente le trait d'union entre la structure physique de l'application industrielle et le système de diagnostic. Ici, les différents états fonctionnels des composants ou des unités sont décrits. Le cas le plus fréquent est lié à l'état de comportement normal du système, mais il est aussi possible de décrire le comportement d'états anormaux associés à des défauts précis. Un état est donc lié à un comportement, lequel peut être représenté avec un modèle mathématique généré à partir des lois physiques et qui relie différentes variables. À ce stade, il est plus important de connaître quelles variables sont impliquées que de connaître les valeurs des paramètres du modèle.
- ii. Sélection de la stratégie globale de diagnostic et sélection des méthodes de diagnostic au niveau des unités physiques grâce à un système de sélection assisté par ordinateur implémenté dans le PSA.
- iii. Mise en oeuvre des tests de détection associés aux agents de diagnostic
 - La génération de modèles de référence: cette tâche a comme objectif la génération de modèles de référence et/ou l'établissement des gabarits et des seuils utilisés pour la décision de la détection qui sont utilisés en ligne par les agents de diagnostic. Cette fonction est accomplie grâce à la collaboration du PSA, du GPMAS et des divers DAAs (Figure 3-5). Ici le PSA se sert comme « client » de ces deux agents. D'un côté, le PSA demande aux DDAs les signaux nécessaires pour la construction de modèles et d'un autre côté, le PSA demande au GPMAS la construction de modèles. Les fonctionnalités des DDAs dans cette étape sont :
 - Récupérer les données à partir de la Base de Données pour leur utilisation dans les tâches d'identification des modèles utilisés par les Agents de Diagnostic. Plusieurs jeux de données peuvent être envisagés, pour tester et valider les modèles.

- Réaliser le prétraitement des données (filtrage, mise en forme, validation, etc.)

Le GPMAS est responsable des activités d'identification et de validation des modèles. Finalement, les modèles sont récupérés par le PSA qui les stocke dans la base de données, où les DAs, comme l'agent causal décrit au chapitre suivant, pourront les lire.

- La saisie de l'information pour la configuration des DDA, DSA et OIA: Cette tâche a pour objectif la saisie de diverses informations nécessaires pour le fonctionnement continu de MAGIC. Des exemples de cette information sont les règles d'activation des tests de détection, les conseils de maintenance associés aux composants, etc.

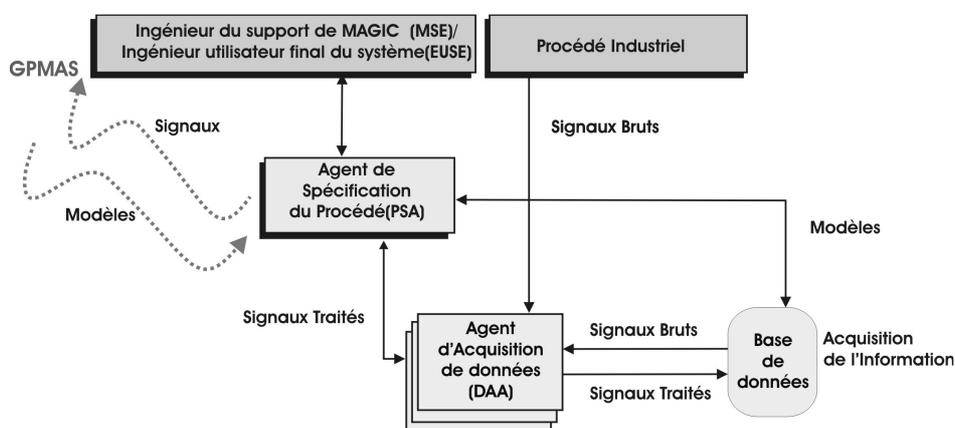


Figure 3-5. Agents utilisés pendant l'étape de configuration et personnalisation.

3.3.2.2 Le mode d'opération continue.

Le mode d'opération continue de MAGIC est le mode d'opération normal, ici les agents d'acquisition de données (DAAs), de diagnostic (DAs), de décision (DDA) et d'aide à l'opérateur (DSA et OIA) entre en jeu pour aider l'opérateur dans ses tâches de supervision en ligne. On décrit ici les démarches associées à ce mode d'opération :

- Acquisition de données :

Le processus de diagnostic et d'aide à l'opérateur commence avec l'acquisition de données du procédé. Différents DAAs récupèrent les données brutes à partir du système SCADA et réalisent le prétraitement des données si c'est prévu (le filtrage, la mise au format requis, la validation, etc.) Ils les stockent aussi dans la base de données. Ensuite, ces agents informent les autres agents de la disponibilité de nouvelles données et fournissent, sur demande, ces données aux agents qui peuvent en avoir besoin.

ii. Diagnostic :

- Génération de symptômes : après l'acquisition des données à partir du procédé, les symptômes doivent être calculés. Ici, la consistance entre le comportement réel du système physique et les modèles de référence doit être vérifiée par les différents tests de détection [Garcia-Beltran, Lesecq et al. 2003].
- Raisonnement de diagnostic : le DDA analyse les symptômes fournis par les DAs courants pour déterminer l'état du procédé entier. Après, il associe les comportements anormaux aux composants physiques. Si les symptômes concernent le même sous-système, cet agent doit résoudre les conflits possibles. De plus, il agrège les symptômes dans une décision globale pour décrire l'état du sous-système. Si les symptômes concernent différents sous-systèmes reliés, cet agent doit employer des informations sur la structure du procédé et sur les influences qu'un sous-système a sur les autres, afin de filtrer les symptômes partiels de sorte qu'il puisse attribuer l'origine du problème observé à une source commune. Deux méthodes différentes de raisonnement pour la localisation de défauts sont utilisées pour localiser les défauts à partir des symptômes détectés; la première méthode est basée sur les tables de signature et la deuxième sur le raisonnement logique[Ploix, Gentil et al. 2003].

L'étape de diagnostic peut opérer de deux façons différentes :

Opération périodique : l'acquisition et la transmission de mesures sont réalisés périodiquement et indéfiniment ; de cette façon des symptômes sont générés continuellement avec une périodicité liée à la périodicité de l'acquisition.

Opération à la demande : la génération d'un ensemble fini de symptômes est réalisée en utilisant seulement un ensemble de données acquises auparavant (et stockées dans la base de données).

iii. Aide à l'opérateur :

- Génération de conseils d'aide à la conduite et la maintenance.
À partir des résultats de diagnostic générés par le DDA, le DSA essaye de faire la prédiction du comportement du système, et de générer les conseils d'action et de maintenance associés aux composants suspectés.
- Interaction avec l'opérateur: au moyen de l'interface graphique de l'OIA, l'opérateur est capable d'accéder à l'information de diagnostic et d'aide à l'opérateur générée par les différents agents de MAGIC. L'opérateur reçoit l'information relative aux anomalies dans le fonctionnement du système (symptômes et variables), la liste des composants suspects (diagnostic), le comportement possible du procédé dans le futur (pronostic) et les conseils pour faire face à la situation anormale.

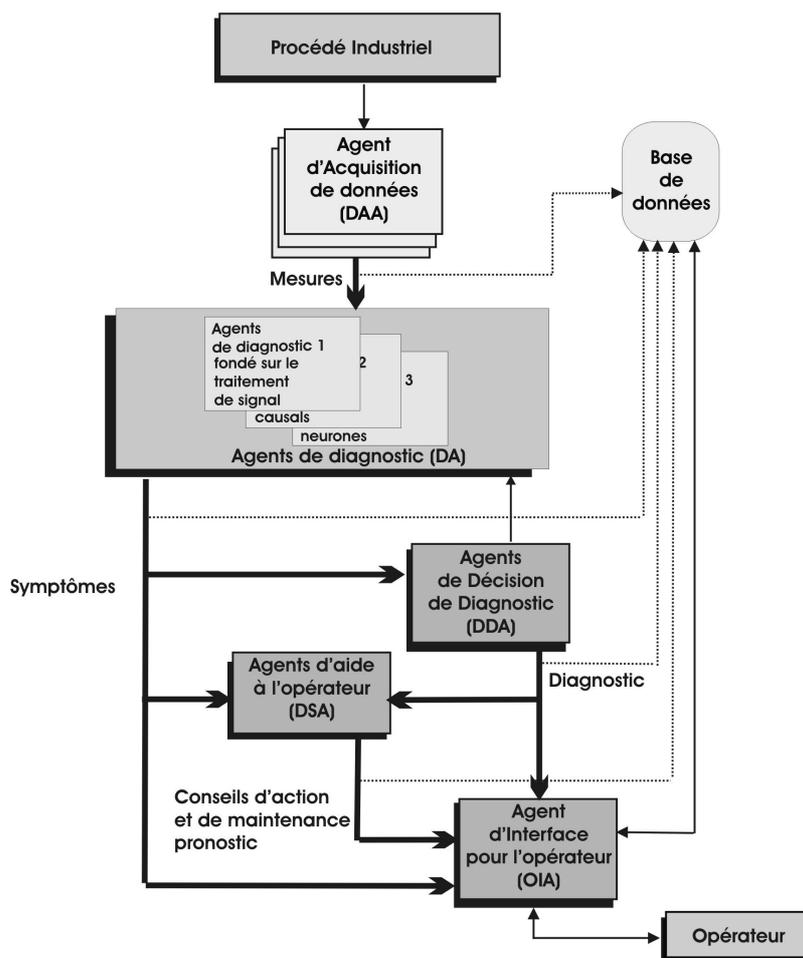


Figure 3-6. Les agents qui participent au mode de fonctionnement continu.

3.4 Architecture interne des agents.

Tous les agents mis en œuvre dans MAGIC ont une structure semblable. Par conséquent, les composants de logiciel déjà développés peuvent être facilement réutilisés. Chaque agent consiste en trois couches différentes (Figure 3-7):

- *Une couche générale de communication* : Cette couche est identique pour tous les agents, elle inclut les fonctionnalités pour la communication avec les autres agents et les fonctionnalités pour l'administration de l'agent au niveau informatique.
- *Une couche spécifique de communication* : elle fournit les services de communication spécifiques et nécessaires pour la couche "supérieure" de l'agent, et pas directement fournis par la couche générale de communication. Par exemple, elle est responsable de stocker les données reçues et de réaliser le prétraitement de l'information.
- *Une couche spécifique* où les algorithmes consacrés aux tâches spécifiques sont localisés.

Cette décomposition permet de dissocier les tâches de communication des tâches de diagnostic ou d'aide à la décision. De plus, cette structure permet de lancer les algorithmes

même s'ils ont été créés dans divers langages de programmation (par exemple JAVA, C ++ ou Matlab). L'architecture d'un agent de diagnostic sera détaillée au chapitre 4 et celle de l'agent d'aide à la décision détaillée au chapitre 5.

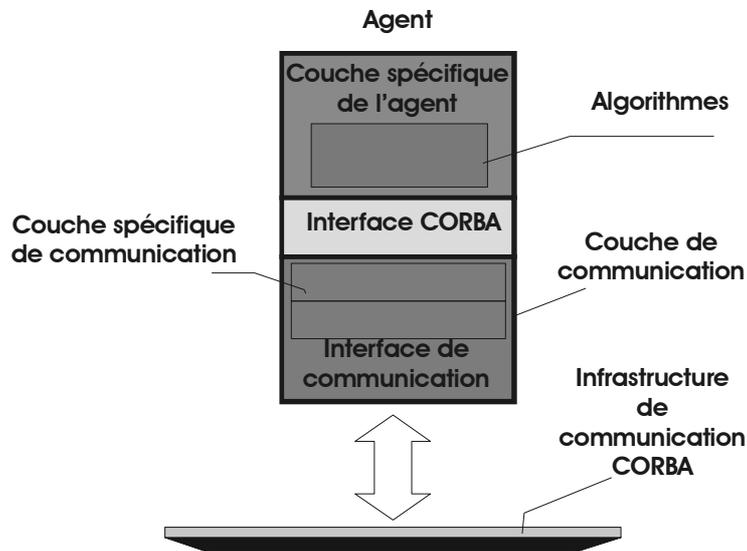
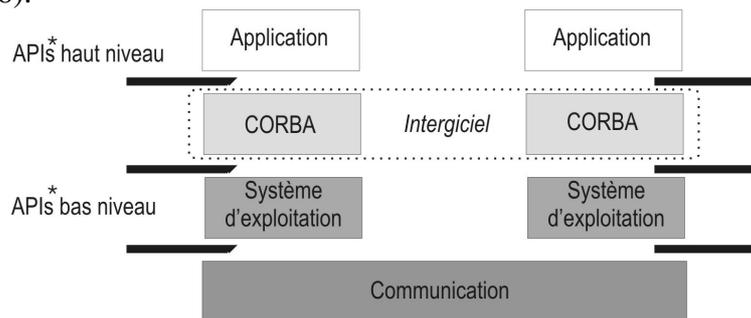


Figure 3-7. Architecture générale d'un agent de diagnostic.

3.5 Les mécanismes de communication dans MAGIC.

Les différents agents de MAGIC accomplissent des tâches très spécifiques qui sont seulement une petite partie du processus global. Afin d'atteindre le but supérieur, les agents doivent échanger l'information. En raison de sa standardisation, CORBA (Common Object Request Broker) a été sélectionné comme *middleware*⁶ de base pour toutes les tâches de communication (Figure 3-8).



*API = Application Programming Interface

Figure 3-8. La position de l'intergiciel dans les applications réparties.

⁶ « 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.

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.

La transparence de la communication fournie par CORBA permet de parler d'un niveau de communication plus élevé, placé dans une couche logiquement supérieure à celle des protocoles de transfert de données (intergiciel + système d'exploitation + TCP/IP, HTTP, IIOP), et destiné au niveau intentionnel et social des agents.

Un agent MAGIC est capable de communiquer avec les autres agents de diverses manières [Albert, Langle et al. 2004] :

- a) en utilisant le langage de communication d'agent FIPA,
- b) en employant une notification d'événements,
- c) en employant la notification d'événements en combinaison avec une technique de transfert d'information via un fournisseur de données.

De plus, l'agent peut employer une interface du type CORBA pour la communication intra-agent et il peut même utiliser le langage SQL pour accéder à une base de données centrale.

3.5.1 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 ; l'un d'entre eux, l'ACL (Agent Communication Language) de la FIPA (the Foundation for Intelligent Physical Agents, [FIPA 2003]) a été pris pratiquement comme standard dans les développements des systèmes multi agent. Le but de l'ACL FIPA est d'interagir entre les agents quelque 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 messages minimums ne suffisent pas toujours pour communiquer : on peut avoir besoin, pour la compréhension du message et pour la rapidité de celle-ci ainsi que la rapidité de traitement du message, d'indiquer d'autres informations telles que le langage utilisé dans le contenu du message, le protocole, l'ontologie à laquelle le message se rattache, la référence d'un message antérieur auquel le message actuel se rattache. Ce langage apporte une grande flexibilité à la communication grâce :

- a) au fait que le contenu du message peut être complètement différent,
- b) aux différents protocoles de transmission,
- c) à la possibilité d'avoir différentes réponses potentielles de l'agent de réception.

Du point de vue de la flexibilité, l'utilisation d'un langage de communication d'agent permet d'avoir un grand contrôle sur l'état de la communication qui rend le système très fiable. Malgré ses avantages, cette communication est associée à un coût en termes de vitesse d'exécution, qui est souvent très importante dans les systèmes critiques multi-agents. Ce type de communication est employé dans MAGIC dans tous les cas où un agent veut communiquer avec un autre et les besoins de vitesse de traitement sont moins importants que la fiabilité et la flexibilité. Cette communication sera employée pour toutes les interactions qui ne sont pas partie du processus régulier de diagnostic :

L'OIA fait une demande au DDA de désactivation d'un test de détection.

L'OIA fait une demande au DSA d'un pronostic d'une variable.

3.5.2 Communication via notification d'événements et fournisseur de données.

Dans un système de diagnostic, la flexibilité apportée par les langages de communication d'agents n'est pas toujours exigée et n'est pas favorisée quand l'échange de l'information entre les agents a été bien défini et reste statique à tout moment où les agents accomplissent leurs tâches.

Dans les systèmes répartis à grande échelle, le meilleur modèle d'une communication robuste et fiable est par l'intermédiaire de la notification d'événements. Contrairement aux autres modèles de communication, ce modèle permet à chaque agent émetteur de continuer sa tâche après les envois de la notification d'événement, l'agent émetteur n'a pas besoin de savoir quels agents sont intéressés par cette notification. Au contraire, sont les agents récepteurs qui doivent s'abonner au service de notification d'événements. Néanmoins, puisque la notification est transmise par l'agent seulement quand il y a un événement 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 a le risque de perdre l'information qui est envoyée à un canal d'événement, mais qui n'est utilisée par aucun agent. Afin d'éviter cette perte d'information, tous les événements qui sont 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.

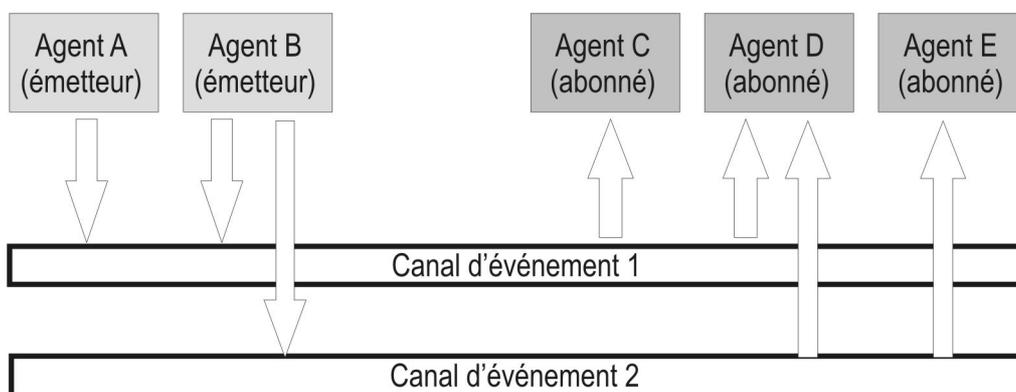


Figure 3-9. Abonnement des agents aux canaux d'événement.

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 .

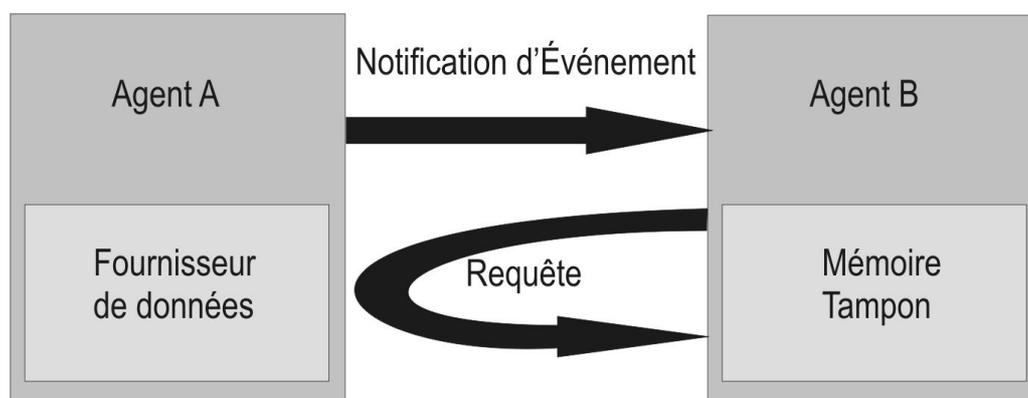


Figure 3-10. Communication via la notification d'événements et le fournisseur de données.

Ce type d'échange de données est employé toutes les fois qu'un flot prédéfini d'information dans le processus de diagnostic est utilisé :

- Transfert des mesures à partir du DAA vers les DAs.
- Transfert des symptômes créés par les DAs vers le DDA et le OIA.
- Transfert de diagnostic à partir du DDA vers le DSA et le OIA.
- Transfert de conseils d'action et de maintenance et des prédictions à partir du DSA vers l'OIA.

3.5.3 Communication intra-agent.

En plus de la communication entre différents agents, les différentes couches d'un agent, chacune étant responsable d'une tâche spécifique de l'agent, doivent également échanger l'information. 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.

3.6 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 de MAGIC, l'organisation de la connaissance associée à la supervision de procédés permet d'avoir une signification unique des concepts d'automatique, de traitement de signal, d'informatique, d'ingénierie de procédé qui sont utilisés.

L'ontologie de MAGIC est la base de la conception du système de communication de haut niveau puisqu'elle décrit la sémantique des messages circulant. 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 3-11**):

- 1) L'ontologie du procédé.
- 2) L'ontologie du diagnostic.
- 3) L'ontologie de l'aide à la décision.

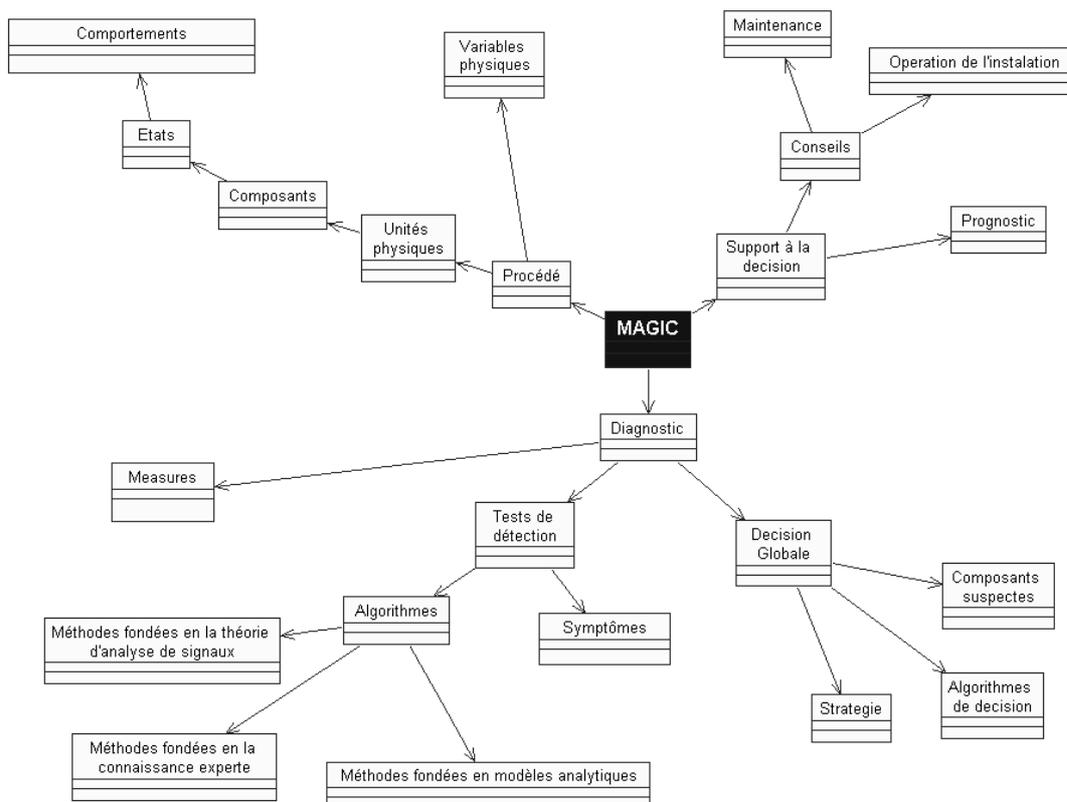


Figure 3-11. Une partie des éléments de l'ontologie du système MAGIC.

Tableau 3-1. Ontologie du procédé.

Unité physique	Est un ensemble de composants ou d'autres unités physiques.
Composant	Est une entité physique considérée comme indivisible ⁷ .
État du composant	Un des états de chaque composant est "normal". Cet élément permet de discriminer entre l'état normal d'un composant et les autres qui peuvent être associés à des modes de défaillance particuliers.
Comportement du composant	Cet élément permet d'établir la manière dont le composant doit se comporter. 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 ou par un gabarit précis de signal. Chaque mode est décrit par un comportement différent.
Variable physique	Une variable physique représente une certaine quantité physique, mesurée ou pas.

Tableau 3-2. Ontologie de diagnostic.

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 qu'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 le résultat des algorithmes utilisés pour la localisation du composant défectueux.

Tableau 3-3. Ontologie du diagnostic.

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.

⁷ La granularité de la décomposition dépend des objectifs. Dans le cas de la maintenance, la meilleure solution est d'arrêter la décomposition au niveau auquel le composant peut être remplacé.

Les symptômes relatifs à un agent de diagnostic seront décrits précisément dans le chapitre 4 et les conseils et le pronostic dans le chapitre 5.

3.6.1 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 dû au 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 *Protégé-2000* [Musen, Ferguson 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 expliqués.

3.7 Conclusion

MAGIC est un système d'aide à la décision pour la supervision de procédés complexes basé sur une architecture multi-agents. Les fonctionnalités requises pour l'aide à la décision ont été distribuées tout à la fois à différents niveaux et dans différents agents.

Sur le plan informatique, MAGIC repose sur une architecture multi-agents hiérarchisée d'agent hétérogènes, qui combine des techniques de communication multi-agents standard avec des techniques adaptées à leur interaction en temps réel avec les systèmes d'automatisation industriels. On peut remarquer que chaque agent comporte dans sa couche spécifique des algorithmes sophistiqués qui contribueront pour beaucoup à l'intelligence du système.

Du point de vue de l'interaction homme-machine, MAGIC situe l'opérateur humain comme le centre de réflexion pour la résolution de problèmes. Toute l'information est utile seulement si l'opérateur humain en fait usage, étant donné qu'il n'existe aucune interaction directe de MAGIC avec le système de commande automatique ou la génération de consignes. On peut ainsi faire le lien avec le schéma de résolution de problèmes proposé par Rasmussen, et déjà expliqué dans le chapitre 2. Le système MAGIC fournit un aide dans la phase de *détection de conditions anormales* au moyen d'un raisonnement automatisé de détection de défauts, dans *l'identification de l'état du système et son anticipation*, grâce aux fonctionnalités de localisation de défaut et de pronostic respectivement, et dans la *définition de tâches* à partir de conseils d'action et de maintenance. Il constitue aussi un aide dans les tâches de *recherche explicite d'information* au moyen des méthodes de diagnostic lancées à la demande (Figure 3-12).

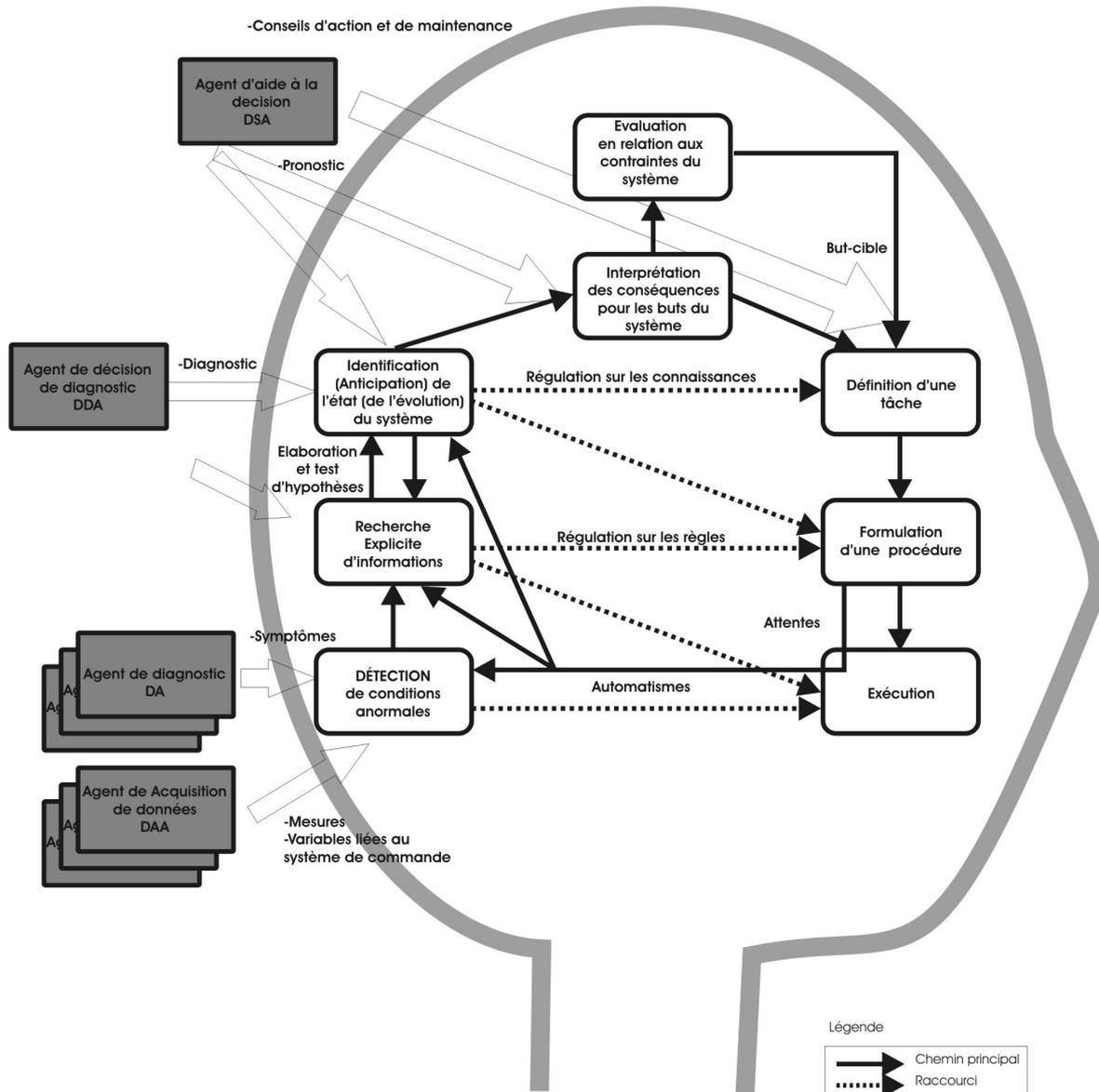


Figure 3-12. MAGIC comme système d'aide à la décision.

Chapitre 4

Les agents de diagnostic : le cas de la détection de défauts fondée sur un modèle causal.

4.1 Introduction

Le chapitre 2 a fait un survol rapide des différents outils de supervision, dans le cadre de l'aide à l'opérateur en particulier. Nous avons montré que le diagnostic de défaillances était au cœur d'un tel système.

Dans le chapitre 3, l'architecture générale du système MAGIC a été présentée; un des objectifs principaux de ce système est celui d'expliquer à l'opérateur l'état du système. Pour cette tâche, MAGIC propose une fonctionnalité de diagnostic qui inclut la détection et la localisation de défauts. L'architecture de MAGIC la décompose en trois parties principales: 1) l'acquisition d'observations 2) la détection de défauts, et 3) leur localisation, ce qui correspond à l'approche usuelle de automaticiens pour le diagnostic.

Comme cela a été déjà mentionné, les différents DAAs 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 DAs après traitement (filtrage, linéarisation de caractéristiques, élimination de points aberrants, etc. [Lacatusu, Stücher et al.

2002]). Le but principal d'un DA est de fournir des *symptômes*, qui représentent l'analyse de l'état d'un sous-système et d'envoyer cette information au DDA, qui exécute le diagnostic final (la localisation d'un composante défaillante, et si il est possible, l'identification du défaut) et au DSA, qui aide l'opérateur pour les décisions d'action ou de maintenance.

Ce chapitre est consacré toute d'abord à la description des fonctionnalités communes aux agents de diagnostic (détection de défauts). Cela amène un certain nombre de contraintes ou conséquences sur les méthodes de détection implémentées. La suite du chapitre décrit de façon précise la manière dont l'agent de détection fondé sur un graphe causal (CBDA : Causal Based Diagnostic Agent) met en œuvre ces fonctions.

4.2 Fonctionnalités des agents de diagnostic.

Une fois que un DA est déclenché, il interagit avec d'autres agents pour pouvoir remplir les fonctions suivantes (voir Figure 4-1) :

1. *Initialisation de l'agent*: l'agent est créé et déclenché. Les divers mécanismes de communication (CORBA, la connexion avec la base de données) sont activés ainsi que les algorithmes de diagnostic.
2. *Génération de symptômes*: À partir de mesures reçues l'agent DA génère les symptômes, et ils sont envoyés vers le DDA, l'OIA et le DSA.
3. *Arrêt de l'agent* : arrêt des activités de l'agent.
4. *Désactivation d'un test de détection* : le DDA peut demander l'arrêt de l'exécution d'un test de détection particulier.
5. *Inspection de l'état de l'agent* : l'agent GUI peut demander si le DA est encore actif.

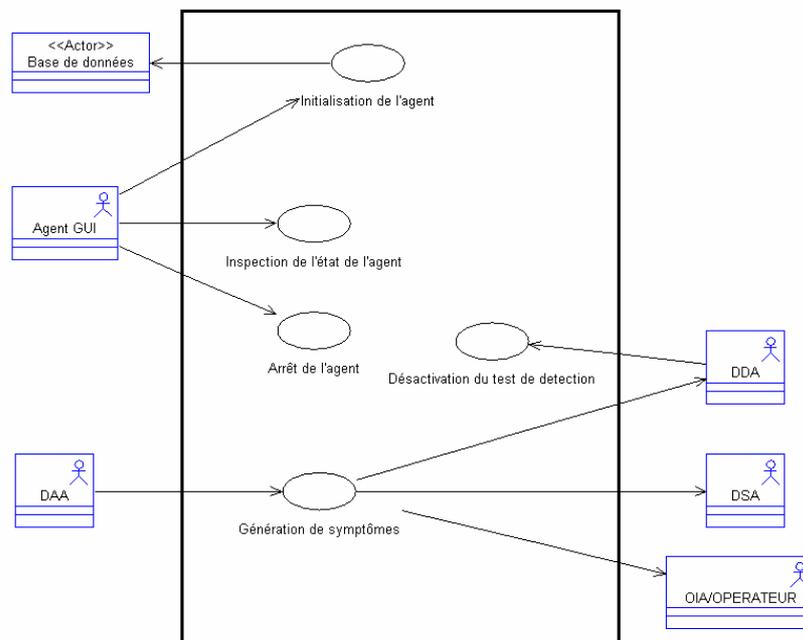


Figure 4-1. Les différents cas d'utilisation des DAs.

4.2.1 L'interaction des agents de diagnostic avec les autres agents

La principale interaction d'un agent de diagnostic pendant le fonctionnement en mode continu est avec les DAAs qui lui envoient les mesures et avec les agents DDA, DSA et OIA qui reçoivent les symptômes générés par cet agent (

Figure 4-2).

Le DDA localise les défauts à l'aide des symptômes fournis par plusieurs agents de diagnostic, le DSA conseille l'action et la maintenance en fonction de l'évolution des variables et des symptômes et l'OIA affiche les symptômes pour l'opérateur.

Tous les agents mis en application dans MAGIC ont une structure semblable : une interface de communication, une couche spécifique de communication et un cerveau d'agent, appelée dans MAGIC la Couche Spécifique. Tandis que les deux premiers points sont exactement les mêmes pour tous les DAs, les Couches Spécifiques sont différentes. En particulier, les algorithmes de détection sont naturellement spécifiques pour chacun des types d'agents de diagnostic.

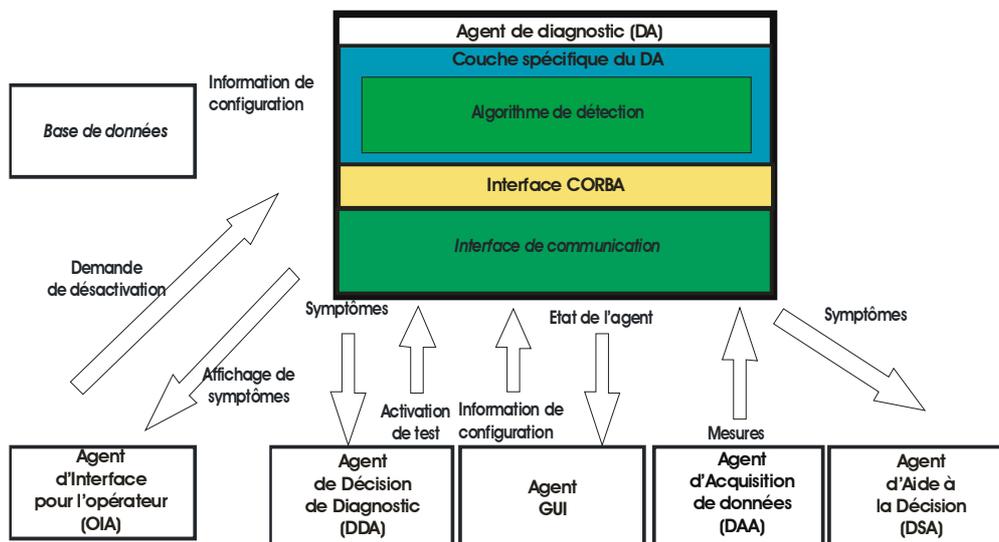


Figure 4-2. Architecture générale d'un agent de diagnostic.

Subséquentement, les fonctionnalités d'un DA sont réparties entre sa couche de communication et sa couche spécifique

Figure 4-3. On parlera ici des fonctionnalités associées à la couche spécifique, et plus précisément de la fonctionnalité de génération de symptômes.

4.2.2 Principes de communication des agents de diagnostic.

Comme cela a été déjà expliqué dans le chapitre 3, la communication de chaque agent de diagnostic est gérée par la couche de communication de chaque agent. Dans le cas des DAs, cette couche a comme fonctionnalités principales :

- L'interaction avec la couche spécifique.
- L'interaction avec l'agent GUI.
- La manipulation des mesures à l'entrée.
- La manipulation des symptômes à la sortie.
- L'interaction avec les autres agents.

La couche de communication est censée être indépendante de l'exécution de la couche spécifique.

Le scénario de communication le plus important de l'agent diagnostic, est pendant l'évaluation des tests de détection. Le diagramme de séquences présenté sur la Figure 4-4 fournit une vue d'ensemble de l'interaction des agents de diagnostic avec les autres au moment du déclenchement d'un test de détection.

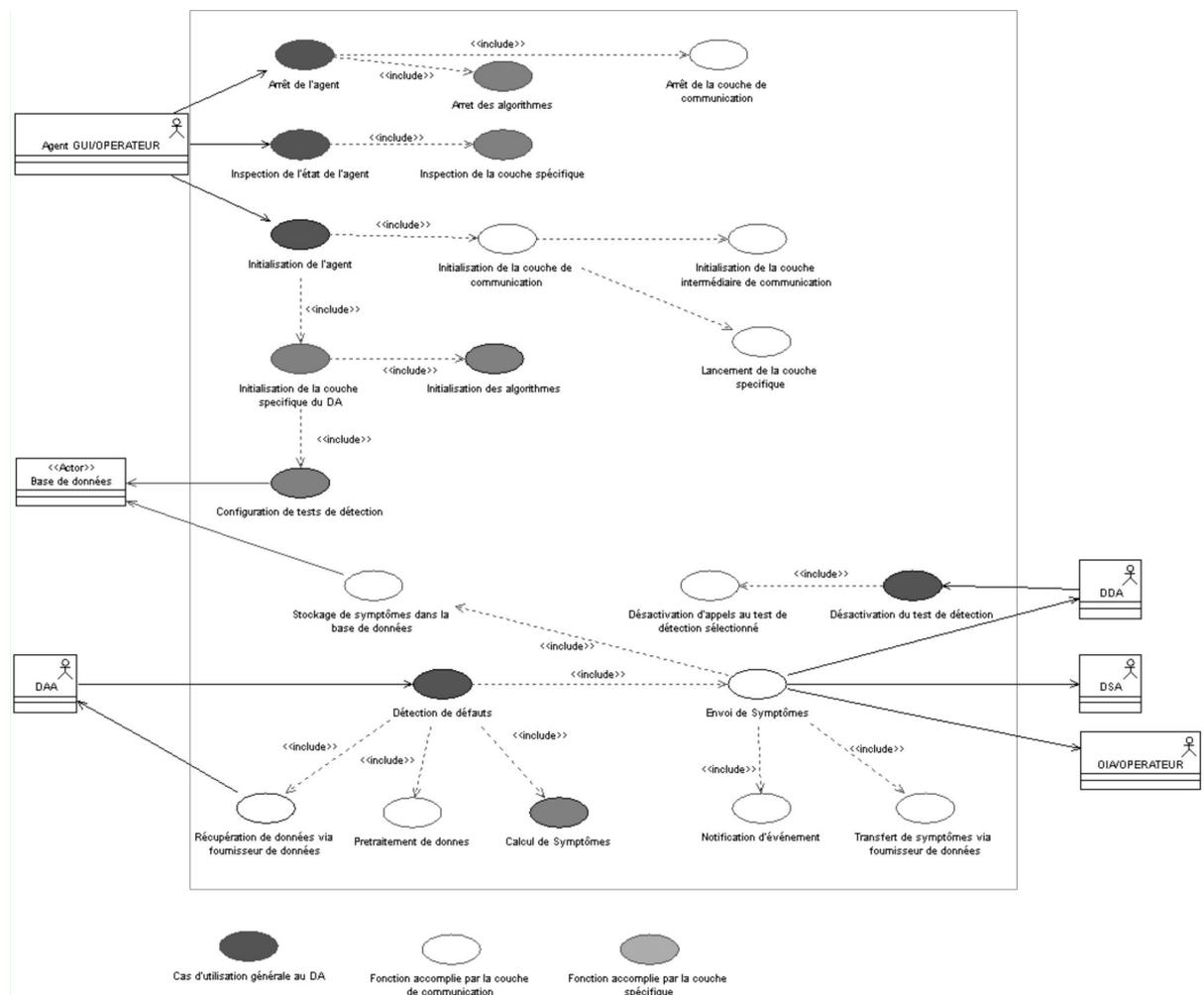


Figure 4-3. Cas d'utilisation de l'agent de diagnostic.

L'agent d'acquisition de données accède à de nouvelles valeurs et envoie un événement de « *mesures disponibles* » au système MAGIC. Cet événement informe sur le nom de la variable physique qui a de nouvelles valeurs. L'agent de diagnostic (DA) reçoit cet événement, et il regarde dans ces paramètres s'il est intéressé par de nouvelles valeurs de cette

variable; si c'est le cas, il réalise une demande des valeurs des variables physiques à l'agent d'acquisition de données. Le fournisseur de données du DAA envoie les données vers le DA. Les mesures reçues des variables physiques sont stockées dans la mémoire tampon de l'agent de diagnostic.

L'agent de diagnostic (DA) vérifie s'il y a suffisamment de données pour une nouvelle évaluation d'un test de détection (disponible dans cet agent), certains algorithmes nécessitent une fenêtre de données relativement longue pour pouvoir être lancés (test statistiques par exemple), et si les conditions pour l'activation du test de détection sont remplies. Si toutes les conditions sont satisfaites, l'agent de diagnostic déclenche un nouveau test de détection. Par conséquent, il doit fournir toutes les valeurs qui sont nécessaires. Après avoir évalué les algorithmes de détection, le DA stocke les symptômes dans la mémoire tampon du fournisseur de données (dans ce cas, fournisseur de symptômes) et il envoie un événement du type « symptômes disponibles » au système MAGIC.

L'agent de décision de diagnostic est un agent qui est intéressé par les nouveaux symptômes. Il reçoit l'événement et réalise une demande au fournisseur de symptômes de l'agent de diagnostic.

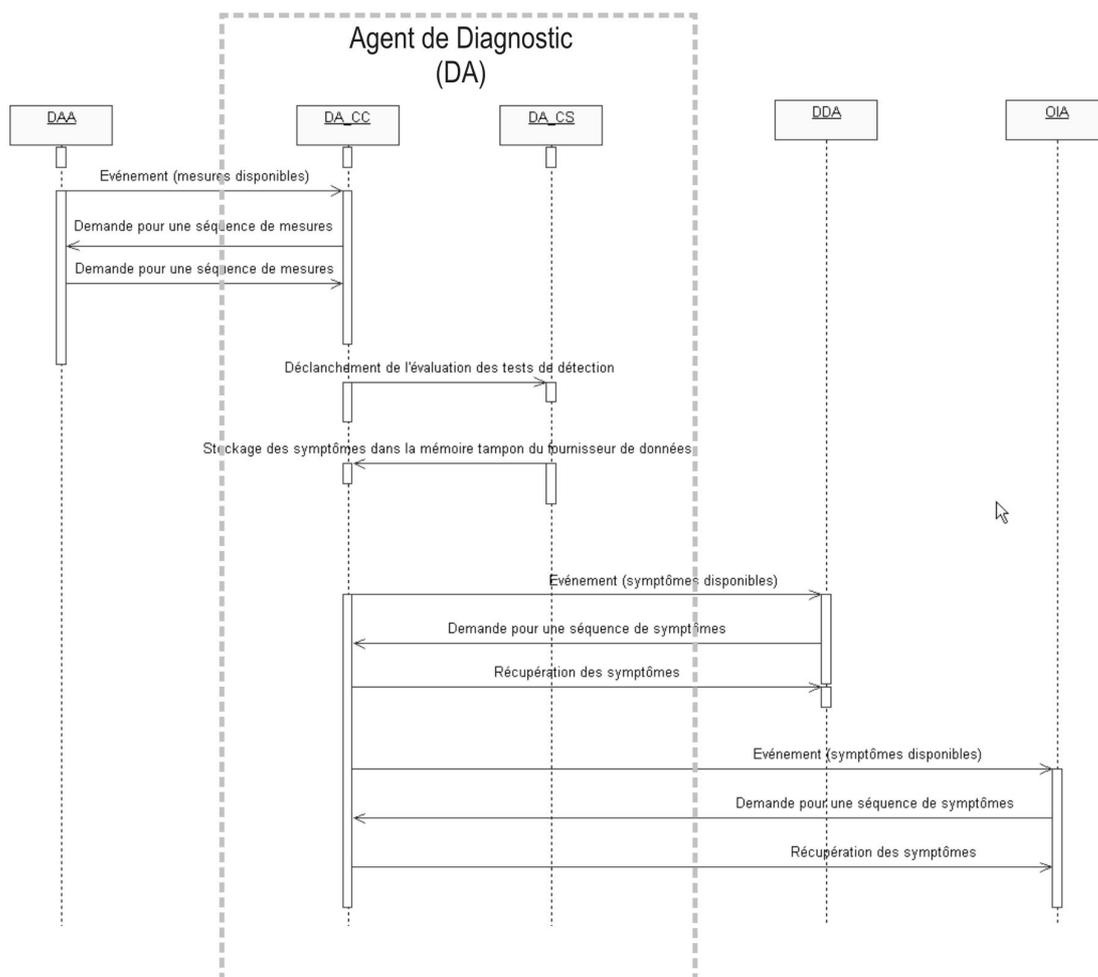


Figure 4-4. Diagramme de séquences d'un processus d'évaluation des tests de détection. (DA_CC : couche de communication, DA_CS : couche spécifique).

4.2.3 L'évaluation des tests de détection pour la génération de symptômes.

Au niveau de la couche spécifique du DA, c'est la définition du *Test de Détection* qui contient l'information nécessaire pour l'évaluation des algorithmes de détection. En suivant l'ontologie de MAGIC, un test de détection est composé de l'information suivante :

Elément de l'ontologie	Attributs
Test de détection	Identificateur unique pour ce test de détection Référence aux algorithmes utilisés Références aux composants testés Les types de déclenchement permis Les références aux signaux utilisés par les algorithmes Un indicateur associé au processus informatique

Un DA peut exécuter un ou plusieurs Tests de Détection d'une catégorie donnée.

Afin de collaborer avec le DDA dans le processus de diagnostic, la sortie d'un Test de Détection, appelé *Symptôme* contient information sur l'état du système mais aussi l'information précise sur l'agent qui a généré le symptôme :

Elément de l'ontologie	Attributs
Symptôme	La caractérisation du résultat du test de détection. La validité de la décision, un nombre dans [0,1]. La validité du modèle, un nombre dans [0,1]. L'instant de génération du symptôme. La référence au test de détection utilisé pour la génération de ce symptôme.

Dans les prochaines sections, on montrera comment ces attributs sont calculés.

Typiquement, les algorithmes classiques de détection sont construits pour produire des décisions booléennes (symptôme au sens classique) suite à l'évaluation d'un indicateur de défaut et ils n'évaluent pas la pertinence des modèles utilisés. Il est plus complexe de générer les validités graduelles de décision et de modèle, et il est bien connu que la pertinence du modèle ne puisse pas toujours être estimée. Néanmoins, toute cette information est nécessaire pour le DDA afin de gérer tous les diagnostics partiels et pour arriver à une décision globale.

Chaque algorithme de détection a ses propres méthodes pour l'évaluation des différents attributs du symptôme. Le système MAGIC implémente une grande variété de méthodes, fondées sur des tests statistiques, des analyses fréquentielles, des modèles analytiques. Dans ce dernier cas, les modèles sont de type entrés/sorties, ou d'état et les méthodes utilisent des modèles en parallèle, des observateurs ou l'estimation des paramètres. On voit qu'à peu près toute la panoplie de méthodes décrites au chapitre 2 sont représentées. Dans ce cadre générique, nous avons été chargés d'une méthode fondée sur un modèle causal de l'installation. Nous allons rapidement citer les avantages et les cas où son utilisation est intéressante. Ensuite les méthodes associées aux modèles fondés sur les modèles causaux sont expliquées en détail.

Deux types de modèles sont généralement disponibles pour les procédés industriels : a) les bilans énergétiques ou de matière établis à partir de schémas fonctionnels et des schémas TI (tuyaux/instrumentation) du procédé qui regroupent la connaissance de l'opérateur aux règles de production, et b) les modèles construits à partir des équations analytiques complexes, non linéaires et basées souvent sur les dérivées partielles qui sont écrit par des physiciens ou des chimistes quand le centre d'intérêt est le procédé et les phénomènes physiques impliqués. De toute façon, ils ont été conçus pour des buts différents de la surveillance. Ainsi, la première étape d'une méthode de diagnostic fondée sur modèle est d'obtenir, bien évidemment, un modèle approprié. Cette tâche est en général très lourde en ce qui concerne la connaissance nécessaire et la quantité de temps exigé pour sa mise en œuvre, à l'échelle d'une installation industrielle. Les méthodes reposant sur des représentations d'état ou l'estimation de paramètres ne peuvent en général tester qu'un nombre réduit d'états ou de paramètres à surveiller.

Afin de surmonter cette difficulté, de nouvelles techniques, -et notamment issues de l'intelligence artificielle (IA) et la modélisation qualitative - ont été adoptées. Néanmoins le calcul des valeurs qualitatives à partir d'un modèle qualitatif mène en général à de multiples solutions et souvent en mélangeant des solutions vraies et fausses. Un autre problème avec ces techniques est que ce n'est pas facile de gérer le temps, ce qui n'est pas compatible avec le diagnostic en temps réel. Un autre outil proposé par l'IA est le raisonnement causal. Le diagnostic est typiquement un processus causal. 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 représentée 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. Le raisonnement de diagnostic peut être purement local, ainsi la taille de l'installation diagnostiquée peut être très grande. La méthode de modélisation causale décrite ici est fondée sur une représentation causale qualitative du fonctionnement normal du procédé et sur des modèles locaux de comportement quantitatifs. L'approche discutée ici se fonde sur le raisonnement causal sur des données quantitatives qui permet une sélection plus précise.

De plus, le raisonnement causal est facilement interprété par un opérateur. Toute le reste de ce chapitre sera consacré aux méthodes de diagnostic mises en œuvre dans l'agent de diagnostic basé sur les modèles causaux (CBDA).

4.3 Génération des symptômes à partir du modèle causal.

Le comportement normal de n'importe quel dispositif industriel de procédé de fabrication peut être partiellement décrit par un graphe causal, composé de variables (nœuds) et de relations continues entre les variables (arcs). 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_o dépend de l'état de x au temps $t < t_o$; 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.

Le modèle causal à construire est consacré à la surveillance, donc le modèle causal doit être basé sur une information avec une signification physique et être facilement compris par l'opérateur. Les variables considérées dans le graphe sont les variables connues, familières à l'opérateur, comme les variables mesurées (températures, débits, pression etc.) ou les sorties de régulateur. La limitation du modèle à ces variables permet d'éviter la surcharge cognitive à l'opérateur.

Un graphe causal peut être obtenu par une analyse physique soignée et une analyse fonctionnelle descendante. 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 portées par les arcs peuvent être obtenus avec des procédures classiques d'identification.

Dans le cas d'un procédé dont un modèle décrit par un système d'équations est disponible, les relations causales parmi les variables sont implicites : les effets sur les sorties ne peuvent pas précéder les variations d'entrée ; définie de cette façon la causalité devient équivalente à la notion de la calculabilité, et est liée à l'ordonnancement implicite des équations par les algorithmes de simulation. Ceci signifie que le système d'équations peut être représenté par un graphe causal qui peut inclure des boucles.

$$\begin{cases} \dot{x}_1 = g_1(u_1, u_n, x_1) \\ \dot{x}_2 = g_2(u_i, x_2) \\ \dot{x}_3 = g_3(u_1, x_2, x_n, x_3) \\ \dots \\ \dot{x}_n = g_n(x_3, x_4, \dots, x_n) \end{cases} \quad (4.1)$$

La sortie d'une équation devient l'entrée de l'équation suivante, correspondant au choix d'une relation d'ordre partiel comme montré sur la **Figure 4-5**. Nous reviendrons sur la construction d'un graphe causal en 4.4 pour décrire les outils que nous avons implémenté.

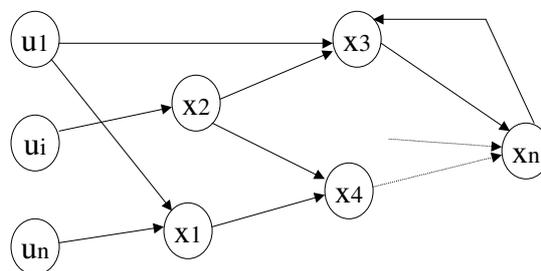


Figure 4-5. Causalité: une relation d'ordre partiel.

Une structure causale est un moyen pour la représentation de la connaissance sur le comportement d'un système avec un niveau d'abstraction élevé. Néanmoins elle est parfaitement compatible avec les outils classiques de l'automatique : des fonctions de transfert peuvent être assignées aux arcs. Ce sera le cas dans l'application MAGIC. La simulation est alors alimentée par des données échantillonnées : les consignes, les signaux de

commande ou les perturbations mesurées (variables exogènes). Le modèle exploité au niveau d'une variable mesurée y est alors de la forme (voir aussi la Figure 4-6) :

$$y_i(k+1) = (1 - F_i(q^{-1})) y_i(k) + \sum_{j \in U_i} q^{-d'_{ji}} B_{ji}(q^{-1}) u_j(k) \quad (4.2)$$

Ici k est la période de calcul, F_i , B_{ji} sont des polynômes en l'opérateur de retard q et les d'_{ji} et d''_{ji} représentent les retards par rapport aux autres variables ou entrées; les valeurs y_j sont les valeurs précédentes de y_i et u_i se rapporte à l'ensemble d'indices j des entrées du procédé affectant directement y_i .

D'autres types de modèles pourraient être implantés, ce qui ne sera pas abordé dans la suite. L'agent DA causal est donc réservé aux procédés complexes de production continue travaillant autour d'un point de fonctionnement.

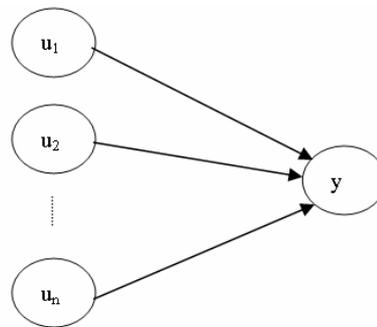


Figure 4-6. Un sous-graphe élémentaire.

4.3.1 Algorithme général d'évaluation du symptôme (cf. définition § 4.2.3).

L'évaluation du test de détection pour la génération de symptômes, appliqué au temps t , est décrite rapidement ci-dessous, chaque point sera explicité plus bas.

1. La couche spécifique reçoit les mesures $y(t)$.

2. Caractérisation du résultat du test :

$$r = \begin{cases} \text{connu,} & \text{Il est possible d'évaluer le test} \\ & \text{(toutes les conditions nécessaires pour} \\ & \text{le test de détection sont disponibles)} \\ \text{inconnu,} & \text{Le test ne peut pas être réalisé} \end{cases}$$

Si cette variable est égale à "connu", les points 3, 4 et 5 sont calculés, sinon l'algorithme s'arrête.

3. La Validité du Modèle est évaluée (§ 4.3.4).

Evaluation du *critère partiel*₁
 Evaluation du *critère partiel*₂
 ⋮
 Evaluation du *critère partiel*₃

validité du modèle = $h(\text{critère partiel}_1, \text{critère partiel}_2, \dots)$

4. Les résidus locale et global son calculés au moyen de la mesure $y(t)$, la propagation globale $y^*(t)$ et la propagation locale $\hat{y}(t)$ (§ 4.3.3).

Le résidu global est calculé comme :

$$\text{résidu global}(t) = y(t) - y^*(t)$$

Le résidu local est calculé comme :

$$\text{résidu local}(t) = y(t) - \hat{y}(t)$$

5. Évaluation de la validité de la décision pour les résidus locaux et globaux (§4.3.5):

$$\text{validité de la décision} = f(\text{résidus})$$

4.3.2 Caractérisation du résultat du test.

Si les valeurs actuelles et passées nécessaires pour l'évaluation du test de détection sont disponibles, et le modèle causal est disponible pour la configuration actuelle, l'agent est donc capable d'effectuer son diagnostic. Dans ce cas-ci :

$$\text{résultat} = (\text{la caractérisation de résultat du test}) = \text{"connu"}.$$

Dans l'autre cas, quand toutes les mesures ne sont pas accessibles ou quand aucun modèle est disponible :

$$\text{résultat} = (\text{la caractérisation de résultat de symptôme}) = \text{"inconnu"}.$$

Ce qui signifie que les tests de détection ne peuvent pas être calculés et que l'état du système ne peut pas être déterminé avec ces tests.

4.3.3 Simulation causale.

Comme mentionné dans la section 4.2, on a associé des fonctions de transfert à chaque arc du graphe causal. Celle-ci peut être discrète comme en (4.2) ou continue. Afin de simplifier la

tâche de simulation et accélérer le calcul, la fonction de transfert continue représentée par chaque arc est transformée en une équation aux différences obtenue par la théorie des systèmes échantillonnés. Pour assurer la précision de la simulation la période utilisée pour la simulation est inférieure à la période d'échantillonnage du procédé.

Si on suppose par exemple que les relations entre les variables montrées sur laFigure 4-5 sont linéaires. Le simulateur discret calcule le comportement de y_i en utilisant (4.2). On notera qu'on travaille uniquement avec des modèles mono-sortie.

La simulation causale est traitée de façon à calculer séparément chaque contribution correspondant à un arc. La simulation au niveau du graphe entier consiste seulement à répéter la simulation pour chaque nœud dans le graphe.

Soit un sous-graphe élémentaire représenté par (4.3) , où y est le nœud de sortie et les u_i sont les nœuds d'entrée. La valeur y d'une variable influencée par plusieurs arcs, dont les nœuds antécédent sont u_i , est la somme des contributions des u_i par les arcs d'entrée :

$$y(k) = \sum_i \frac{B(q)}{F(q)} u_i(k) \quad (4.3)$$

Ici, $\frac{B(q)}{F(q)} \cdot u_i(k)$ représente la valeur calculée avec l'équation aux différences déduite de la fonction de transfert attachée à chaque arc entrant.

Il faut noter que ceci suppose implicitement que la superposition des effets d'entrée est valide, qui est une hypothèse de linéarité (même si quelques contributions pourraient être calculées par des relations non linéaires). Le système de diagnostic travaille avec des modèles de petites variations autour d'un point de fonctionnement : quand le simulateur causal est démarré, on suppose que le système est en état stationnaire et les conditions initiales de 4.2 sont supposées nulles.

4.3.3.1 Simulation au niveau d'un arc.

Un ensemble de fonctions de transfert classiques discrétisées a été pris comme bibliothèque de base pour la configuration du DA (Agent de Diagnostic). Pendant la phase de configuration, le MSE (MAGIC System Engineer) choisit les fonctions spécifiques de la bibliothèque à lier à chaque arc du graphe causal. Il doit donner la valeur numérique des paramètres. Ceci est fait avec l'interface spécifique du PSA (Agent de spécification du procédé); si besoin, les paramètres sont identifiés en utilisant le GPMAS (Serveur d'algorithmes mathématiques génériques); cela sera expliqué en 4.4.

Fonction de transfert de retard pur.

Fonction de transfert.	$H(s) = Ae^{-sTd} \quad (4.4)$
Equation aux différences implémentée.	$y[k] = A \cdot u[k - N - 1] \quad (4.5)$ <p>où</p> $N = \begin{cases} \frac{Td}{Tc} & \text{Si } Td \text{ est multiple de } Tc \\ \text{rond}\left(\frac{Td}{Tc}\right) & \text{Si } Td \text{ n'est pas multiple de } Tc \end{cases}$ <p>$Td =$ retard $Tc =$ pas de simulation $A =$ gain</p>

Dans le cas où le retard n'est pas un multiple du pas de simulation, on prend $N = \text{rond}(Td/Tc)$, où la fonction $\text{rond}(x)$ arrondit l'élément x au nombre entier le plus proche.

Fonction de transfert du premier ordre avec retard

Fonction de transfert	$H(s) = \frac{Ae^{-sTd}}{1 + s\tau} \quad (4.6)$
Equation aux différences implémentée.	$y[k] = y[k - 1] \cdot \alpha + A \cdot (1 - \alpha) \cdot u[k - N - 1] \quad (4.7)$ <p>où:</p> $\alpha = e^{-Tc/\tau}$ $N = \begin{cases} \frac{Td}{Tc} & \text{Si } Td \text{ est multiple de } Tc \\ \text{rond}\left(\frac{Td}{Tc}\right) & \text{Si } Td \text{ n'est pas multiple de } Tc \end{cases}$ <p>$Tc =$ pas de simulation $Td =$ retard $A =$ gain $\tau =$ constant du temps</p>

Fonction de transfert de premier ordre avec retard et zéro

Fonction de transfert:	$H(s) = \frac{Y(s)}{U(s)} = \frac{A \cdot e^{-sT_d} \cdot (1 + s \cdot z)}{(1 + s \cdot \tau)} \quad (4.8)$
Equation aux différences implémentée.	$y[k] = y[k-1] \cdot \alpha + A \cdot (1 - \alpha - \frac{z}{\tau}) \cdot u[k-N-1] + \beta \cdot z \cdot u[k-N] \quad (4.9)$ <p>où :</p> $\alpha = e^{-T_c/\tau}$ $\beta = \frac{A}{\tau}$ $N = \begin{cases} \frac{T_d}{T_c} & \text{Si } T_d \text{ est multiple de } T_c \\ \text{rond}\left(\frac{T_d}{T_c}\right) & \text{Si } T_d \text{ n'est pas multiple de } T_c \end{cases}$ <p>T_c = pas de simulation T_d = retard A = gain τ = constant du temps z = zero de la fonction de transfert</p>

Fonction dérivée avec retard.

Fonction de transfert	$H(s) = A \cdot e^{-sT_d} s \quad (4.10)$
Equation aux différences implémentée.	$y[k] = A \cdot [u[k-N-1] - u[k-N-2]] \quad (4.11)$ <p>where:</p> $N = \begin{cases} \frac{T_d}{T_c} & \text{Si } T_d \text{ est multiple de } T_c \\ \text{rond}\left(\frac{T_d}{T_c}\right) & \text{Si } T_d \text{ n'est pas multiple de } T_c \end{cases}$ <p>T_c = pas de simulation T_d = retard A = gain</p>

Le modèle (dérivateur pur) n'est pas en principe physiquement réalisable. Toutefois il peut constituer une approximation intéressante de certains phénomènes.

Fonction de transfert de premier ordre avec retard et terme dérivé.

Fonction de transfert	$H(s) = \frac{Y(s)}{U(s)} = \frac{A \cdot e^{-sT_d} \cdot s}{1 + s \cdot \tau} \quad (4.12)$
Equation aux différences implémentée.	$y[k] = y[k-1] \cdot \alpha + \beta \cdot (u[k-N] - u[k-N-1]) \quad (4.13)$ <p>where:</p> $\alpha = e^{-T_c/\tau}$ $\beta = A/\tau$ $N = \begin{cases} \frac{T_d}{T_c} & \text{Si } T_d \text{ est multiple de } T_c \\ \text{rond}\left(\frac{T_d}{T_c}\right) & \text{Si } T_d \text{ n'est pas multiple de } T_c \end{cases}$ <p>T_c = pas de simulation T_d = retard A = gain τ = constant du temps</p>

Fonction de transfert de deuxième ordre avec retard et terme dérivé.

Fonction de transfert	$H(s) = \frac{Y(s)}{U(s)} = \frac{A \cdot e^{-sT_d} \cdot s}{(1 + s \cdot \tau_1)(1 + s \cdot \tau_2)} \quad (4.14)$
Equation aux différences implémentée.	$y[k] = (\alpha_1 + \alpha_2) \cdot y[k-1] - \alpha_1 \cdot \alpha_2 \cdot y[k-2] + \beta \cdot [u[k-N-1] - u[k-N-2]] \quad (4.15)$ <p>where:</p> $\alpha_1 = e^{-T_c/\tau_1}$ $\alpha_2 = e^{-T_c/\tau_2}$ $\beta = A \cdot \left(\frac{\alpha_1 - \alpha_2}{\tau_1 - \tau_2} \right)$ $N = \begin{cases} \frac{T_d}{T_c} & \text{Si } T_d \text{ est multiple de } T_c \\ \text{rond}\left(\frac{T_d}{T_c}\right) & \text{Si } T_d \text{ n'est pas multiple de } T_c \end{cases}$ <p>T_c = pas de simulation T_d = retard A = gain τ_1 = constant du temps 1 τ_2 = constant du temps 2</p>

Fonction de transfert linéaire discrète générique.

Fonction de transfert	$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z^{-(nk)} + b_1 z^{-(nk+1)} + b_2 z^{-(nk+2)} \dots + b_{nb} z^{-(nk+nb)}}{1 + f_1 z^{-1} + f_2 z^{-2} + \dots + f_{nf} z^{-nf}}$	(4.16)
Equation aux différences implémentée.	$y[k] = -f_1 y[k-1] - f_2 y[k-2] - \dots - f_{nf} y[k-nf] + b_0 u[k-nk] + b_1 u[k-(nk+1)] + \dots + b_{nb} u[k-(nk+nb)]$	(4.17)

En cas de nécessité, le MSE pourrait introduire d'autres relations spécifiques dans la bibliothèque, mais ceci devrait être programmé sur demande. Cela pourrait être fait par exemple dans le cas où il existe des relations statiques non linéaires. Cela ferait partie de la tâche de personnalisation.

4.3.3.2 Propagation globale.

Pour chacune des variables, sa valeur *globalement propagée* est calculée à l'aide de la valeur *globalement propagée* de chacun de ses antécédents :

$$y^*(k) = \sum_i \frac{B(q)}{F(q)} u_i^*(k) \quad (4.18)$$

La valeur *globalement propagée* d'un antécédent est elle-même fonction des valeurs *globalement propagées* de ses antécédents, jusqu'à ce que les consignes ou les variables exogènes mesurées soient trouvées. Ainsi u_i^* est la valeur de référence de l'antécédent (en l'absence du bruit et des défauts dans le processus), et y^* est la valeur de référence du noeud : la valeur qu'aurait la variable en l'absence du bruit et de défauts, supposant le modèle parfait.

4.3.3.3 Propagation locale.

Pour chacun des variables, sa valeur *localement propagée* est calculée à partir des *mesures* de chacun des antécédents :

$$\hat{y}(k) = \sum_i \frac{B(q)}{F(q)} u_{imes}(k) \quad (4.19)$$

où les u_{imes} sont les mesures de chacun des antécédents. La valeur *localement propagée* est ainsi juste une conséquence des modèles locaux attachés aux arcs entrants et de la valeur mesurée des antécédents les plus proches. Ceci correspond en fait à la prédiction de la valeur du noeud à partir des mesures des antécédentes.

4.3.3.4 Correction des valeurs prédites en présence d'un intégrateur.

Quand le modèle est instable (c'est par exemple le cas dans un système avec un intégrateur pur), la simulation pourrait progressivement dévier du comportement réel parce qu'une petite erreur sur l'entrée de l'intégrateur a comme résultat une erreur sur la sortie augmentant indéfiniment.

Afin d'éviter une déviation progressive des valeurs de simulation en présence d'un intégrateur pur, un algorithme de correction a été implémenté. Dans le cas général, quand un noeud a plusieurs noeuds parents, aucune vraie connaissance (mesures) n'est disponible sur la contribution partielle de chaque arc; par conséquent la correction doit être envisagée au niveau global du noeud.

Comme décrit ci-dessus, la valeur finale est calculée à partir des contributions partielles des arcs entrants. Un terme de correction a été ajouté pour compenser la déviation (équation 4.29) :

$$y_{t+1}^* = \underbrace{\sum_i F_i^*(u_i^*)}_{\text{prédiction}} + \underbrace{k(y_t^* - y_{meas_t})}_{\text{correction}} \quad (4.20)$$

Si une valeur très petite est employée pour le paramètre k , l'influence du terme de correction est trop petite pour compenser la divergence. Si k prend une valeur trop grande, le test de détection peut être peu sensible à quelques défauts lents. Une valeur pratique de $k = 0.05$ a été prise comme valeur par défaut.

4.3.4 La validité du modèle.

La validité du modèle donne la confiance qui peut être attachée au modèle de référence. C'est un nombre entre 0 et 1 tel que plus il est proche de 1, plus on a confiance dans le modèle. Cette valeur influence évidemment la confiance en la décision prise avec ce modèle. Cet indicateur peut être calculé à partir de plusieurs indicateurs partiels appelés *INDi*. Nous définirons des indicateurs intermédiaires et des indicateurs partiels dans l'intervalle [0..1]. Ceci nous permet de les agréger par les méthodes d'agrégation floue, que nous détaillerons ci-dessous.

4.3.4.1 L'indicateur de la validité a priori du modèle.

Chaque modèle (fonction de transfert) associé aux arcs du graphe causal, est obtenu pendant la phase de configuration/personnalisation grâce à certaines méthodes d'identification. Les méthodes d'estimation de paramètres utilisées dans MAGIC donnent une idée de la précision du modèle, ainsi que des paramètres du modèle. Cette précision dépend de l'adéquation entre les hypothèses et les conditions expérimentales.

Pour l'identification de la fonction de transfert, il est possible d'obtenir un critère des moindres carrés global qui évalue l'ajustement entre les données et le modèle ; nous obtenons

également les écarts-type des paramètres du modèle. Ces indicateurs peuvent être transformés en un seul indicateur *a priori* sur la validité du modèle.

De cette façon, $accur_1$ est un premier indicateur associé à la précision du modèle :

$$accur_1 = 1 - \min \left(\frac{\sum_N e_k^2}{\sum_N y_k^2}, 1 \right) \quad (4.21)$$

Ici y représente les mesures, e l'erreur entre le modèle et les mesures, k l'instant de temps (numéro d'échantillon) et N le numéro d'échantillons utilisés. Cette relation représente le rapport d'énergie entre l'intégral de l'erreur de prédiction et la valeur réelle de la variable.

Pour le calcul de l'indicateur, associée à la précision de chaque paramètre du modèle:

$$accur_{\theta i} = 1 - \min \left(\left| \frac{\sigma_{\theta i}}{\theta i} \right|, 1 \right) \quad (4.22)$$

Où θi est un paramètre du modèle et $\sigma_{\theta i}$ son écart-type. Cette relation représente le rapport entre la déviation standard d'un des paramètres et la valeur du paramètre. Elle est de certain façon le pourcentage d'erreur du paramètre.

Tous ces critères partiels peuvent être agrégés dans un seul indicateur qui indique la *validité du modèle a priori*:

$$IND1 = h(accur_1, \dots, accur_{\theta i}, \dots) \quad (4.23)$$

Ici h est un opérateur d'agrégation flou du type compromis qui vérifie les propriétés 4.24. Il correspond à une situation intermédiaire entre une agrégation de type disjonction et une agrégation du type conjonction. Des exemples de cet opérateur sont donnés par les équations 4.25, 4.26 et 4.27.

$$\min(\mu(r_1), \mu(r_p)) \leq h(\mu(r_1), \mu(r_p)) \leq \max(\mu(r_1), \mu(r_p)) \quad (4.24)$$

$$h = \frac{1}{p} \sum \mu(r_i) \quad (4.25)$$

$$h = \sum \frac{1}{\mu(r_i)} \quad (4.26)$$

$$h = \left(\frac{\sum \mu^q(r_i)}{p} \right)^{1/q} \quad (4.27)$$

On a utilisé l'opérateur d'agrégation compromis (4.25) du à la simplicité de calcul:

$$IND1 = \frac{1}{p} \sum accur_i \quad (4.28)$$

Ainsi nous supposons que dans la phase de personnalisation une validité du modèle *a priori* peut être obtenue, il s'agit d'un premier critère partiel *IND1*.

4.3.4.2 Indicateur de l'évaluation en ligne de la validité du modèle.

Une fois en opération, le DA doit évaluer la validité du modèle par rapport à l'état réel du système. Puisque le modèle linéaire n'est qu'une approximation du système non linéaire, la validité de cette approximation affecte forcément la qualité de la détection.

Ainsi, d'autres critères peuvent être évalués. Un premier indicateur correspond à la question: à quelle distance se trouve le point de fonctionnement réel du système des conditions expérimentales dans lesquelles le modèle a été obtenu? Ainsi le point de fonctionnement *OP* autour duquel le modèle a été obtenu devrait être connu (ce peut être un attribut d'un modèle) et le DA peut mesurer la distance entre ce point de fonctionnement et les données réelles qu'il emploie pour le diagnostic. S'ils sont proches, l'indicateur *IND2* correspondant devrait être près de 1.0; sinon cet indicateur devrait être 0.0.

Un exemple d'un tel indicateur est le suivant :

$$IND2_{noeud\ i} = 1 - \min\left(\frac{|y_k - OP|}{OP}, 1\right) \quad (4.29)$$

où y_k représente la valeur de la sortie du modèle en fonction du temps, *OP* le point de fonctionnement, et *k* le temps.

Un troisième indicateur dépend de la pertinence des données. Si l'agent sait que les données viennent d'un capteur très imprécis, ou qu'elles ont été obtenues par interpolation/extrapolation, la confiance en ces données ne sera pas très grande. Ainsi le niveau de confiance des données, cette confiance est calculée par le DAA au moment de l'acquisition et employé comme troisième indicateur *IND3*.

4.3.4.3 Validité globale du modèle.

La validité globale du modèle *vm* est finalement évaluée en utilisant l'opérateur d'agrégation de chaque indicateur partiel *IND_i* :

$$vm = \text{validité du modèle} = h(IND1, IND2, IND3) \quad (4.30)$$

4.3.5 Génération des résidus.

La procédure de diagnostic est exécutée pour tous les nœuds endogènes dans le graphe. Pour chaque nœud, un *résidu local* et un *résidu global* sont générés, ce qui correspond à l'évaluation de deux **tests de détection** différents.

4.3.5.1 Génération des résidus globaux.

Pour chacun des nœuds, la valeur *de la mesure* $y(t)$ au temps t peut être comparée à la valeur *globalement propagée* $y^*(t)$ obtenu comme sortie du simulateur global:

$$y(t) = y^*(t) + \delta^*(t) \quad (4.31)$$

où $\delta^*(t)$ est *le résidu global*. Une valeur élevée de ce résidu signifie que quelque chose d'anormal est détecté dans le procédé. Le défaut peut être primaire ou secondaire. Tous les composants dont le fonctionnement est représenté par les arcs amonts du nœud y sont suspects.

4.3.5.2 Génération des résidus locaux.

Pour chacune des variables, la valeur *mesurée* de la variable $y(t)$ au temps t peut être comparée à la valeur *localement propagée* $\hat{y}(t)$ obtenue par la sortie de la propagation locale. La sortie mesurée est expliquée par la propagation dans le procédé des valeurs des antécédents et par un défaut local:

$$y(k) = f_y + \sum_i \frac{B(q)}{F(q)} u_i^{réel}(k) \quad (4.32)$$

où $u_i^{réel}$ est la valeur réel de la variable u_i dans le système. Ainsi :

$$y(t) - \hat{y}(t) = \left(f_y + \sum_i \frac{B(q)}{F(q)} u_i^{réel}(k) \right) - \left(\sum_i \frac{B(q)}{F(q)} u_i^{mes}(k) \right) = \hat{\delta}(t) \quad (4.33)$$

où $\hat{\delta}(t)$ est *le résidu local*, il est proportionnel au grandeur du défaut. Une valeur élevée de ce résidu signifie que quelque chose d'anormal est détecté dans le procédé. Le défaut est primaire. L'origine du défaut peut être:

- un défaut sur un (ou plusieurs) des capteurs des antécédents (différence entre $u_i^{réel}$ et u_i^{mes})
- un défaut sur le capteur de la variable y
- une perturbation non mesurable sur le système
- un défaut sur le composant (ou composants) décrit par les relations implantées sur les arcs entrants.

Ce résidu a un support réduit par rapport au résidu $\delta^*(t)$, il est donc plus efficace en termes de la localisation.

4.3.6 La validité de la décision.

Le CBDA est basé sur l'hypothèse de comportement normal des composants (l'état du composant est présumé normal). L'existence d'un résidu non nul est donc théoriquement une preuve du comportement anormal d'au moins un composant. En pratique toutefois, il peut être non nul à cause du bruit ou des erreurs de modélisation ; on va donc essayer d'évaluer la validité de la décision « résidu non nul ». Pour cela, la comparaison des résidus, locaux et globaux, avec une valeur fixe $C1$ est faite (**Figure 4-7**). La valeur $C1$ est spécifique pour chaque variable et ainsi a besoin d'être défini par le MSE comme paramétrage de l'algorithme. Quand le résidu a une valeur plus grande que $C1$, il est sûr qu'il y a une contradiction entre la valeur mesurée et la sortie du modèle. Ainsi, la **Validité de la décision** est 1. Si le résidu a le même ordre de grandeur que l'écart type de bruit σ , il est sûr que le composant est dans son état normal. Donc, la **Validité de la décision** est 0. Si σ est inconnu, $C0$ peut être mis à zéro.

Si le modèle est obtenu avec une procédure classique d'identification, σ peut être évalué en utilisant ces expériences et $C0 = \sigma$, l'écart type de l'erreur de sortie de la procédure d'estimation des paramètres, qui tient compte à la fois de l'erreur de modèle et du bruit de mesure. Si aucune valeur de $C1$ ne peut être fixée par le MSE, une idée est de le fixer $C1 = 4\sigma$.

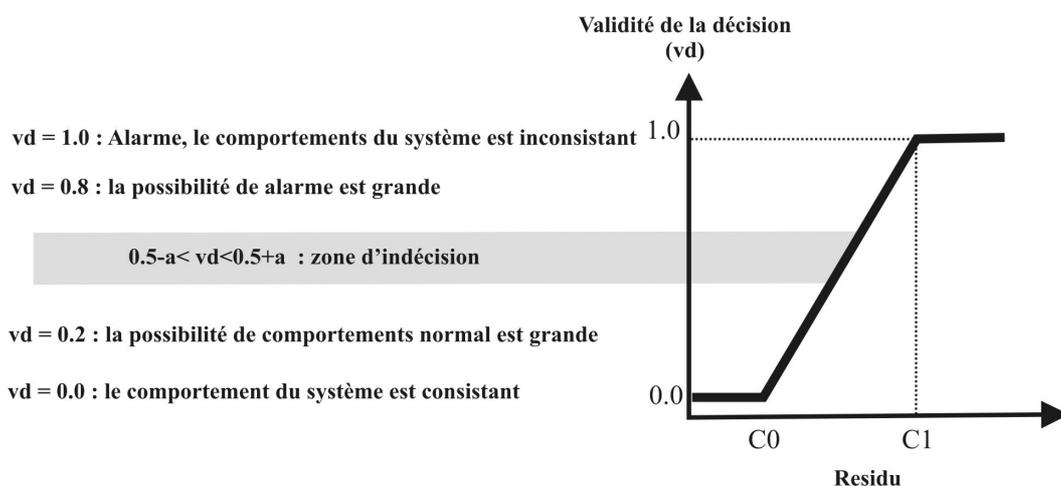


Figure 4-7. Validité de la décision.

D'autres algorithmes peuvent être adoptés pour la prise de décision, par exemple, [Evsukoff, Gentil et al. 2000] décrit un algorithme qui utilise le résidu et aussi la variation du résidu dans une fenêtre du temps pour prendre cet décision.

4.3.7 Les propriétés de localisation de défauts des tests globaux et locaux.

Toutes les méthodes de diagnostic fondées sur des graphes causaux implantent le même principe de base. L'objectif est d'expliquer les déviations détectées dans l'évolution des variables d'un procédé en utilisant un minimum d'hypothèses de défaillance de composant comme source. Un défaut primaire est un changement de l'évolution d'une variable qui est directement attribuable à une défaillance d'un composant ou à une perturbation non mesurable; les défauts secondaires résultent de la propagation dans le procédé de ce changement au cours du temps, causant de nouvelles déviations sur d'autres variables. Le diagnostic fondé sur des graphes causaux consiste à trouver la variable source dont la déviation explique toutes les déviations détectées sur les autres variables.

La décision de localisation peut être énoncée comme suit : si le résidu local est près de zéro, alors l'écart détecté est dû à la propagation des défauts amont dans le graphe; autrement (les résidus globaux et locaux sont élevés), il y a un défaut local (Figure 4-6). Il faut noter que même en présence des défauts multiples, les résidus locaux devraient localiser les défauts multiples.

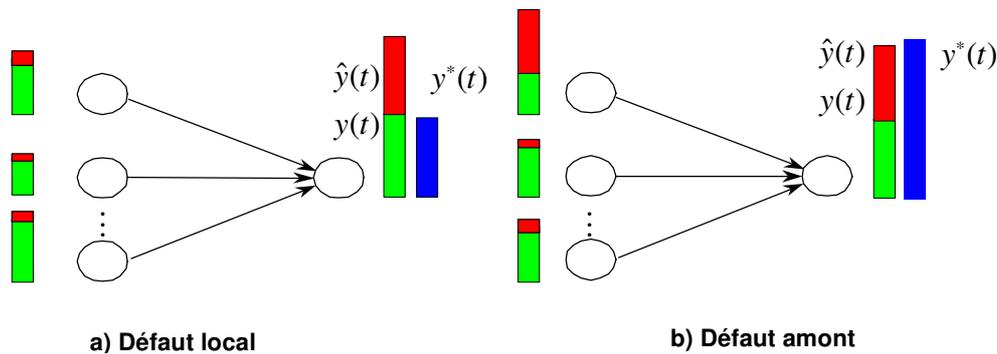


Figure 4-8. Principe de localisation de défauts en utilisant la causalité.

4.4 Configuration et personnalisation de l'agent de diagnostic.

Les DAs sont inactifs pendant la phase de configuration et de personnalisation. Selon le type d'algorithme de diagnostic choisi les besoins de configuration seront différents. En conséquence, pour un DA donné, une procédure consacrée à la définition des propriétés **de(s) test(s) de détection** est commencée dans le PSA pour choisir une méthode de détection spécifique, qui sera appliquée au sous-ensemble de composants couvert par ce DA. Ensuite, les paramètres de la méthode de détection sont choisis ou calculés. À la fin de l'étape de configuration/customisation, les paramètres sont stockés dans la base de données pour un usage postérieur dans le mode de fonctionnement courant et sont maintenus constants jusqu'à ce qu'une nouvelle phase de configuration de cet agent soit initialisée. La

configuration/personnalisation du DA est basée, bien évidemment, sur la construction des graphes causaux pour leur utilisation en ligne.

4.4.1 Construction des modèles causaux.

La modélisation des systèmes est typiquement divisée en trois phases distinctes : la phase de spécification du modèle où la structure du modèle (avec des paramètres libres) est indiquée; l'estimation des paramètres dans laquelle les paramètres libres du modèle sont affectés des valeurs numériques ; la validation du modèle dans laquelle des vérifications statistiques sont employées pour confirmer (ou pas) les deux étapes précédentes, en utilisant de nouvelles données. Il existe une grande variété de techniques d'identification qui peuvent être employées pour identifier chaque arc dans le graphe causal en utilisant des données expérimentales. Les méthodologies pour la validation des modèles sont bien connues aussi. On expliquera plus tard l'utilisation du GPMAS à cet effet.

Il existe divers types de sources de connaissance qui peuvent être employées pour l'obtention d'un graphe causal. Le premier est lié à la description du procédé par un ensemble d'équations algèbro-différentielles qui définissent son comportement. L'obtention de ce genre de connaissance implique tout la problématique de modélisation physique. Néanmoins, une fois que cette connaissance est trouvée, il est relativement facile de la formaliser et de la documenter. Le seul problème est qu'il y a besoin d'une transformation ultérieure pour la mettre sous forme causale. De manière directe, cette connaissance n'est pas vraiment adaptée aux niveaux de supervision, ni adaptée aux explications pour l'opérateur. L'autre genre de connaissance est la connaissance empirique sur le comportement du procédé acquise par les opérateurs et les experts. Cette connaissance est difficile à extraire et formaliser. Mais c'est exactement la connaissance adaptée aux explications du comportement du procédé à l'opérateur.

Quand l'ordonnancement causal des variables et des relations a été trouvé, il est encore nécessaire de simplifier le graphe causal obtenu en éliminant des variables non pertinentes ou non mesurables et en fusionnant les arcs correspondants [Heim, Cauvin et al. 2001]. Puis, les composants physiques peuvent être attachés à chaque arc ; cela permet de faire le lien entre un raisonnement de diagnostic fondé sur l'inconsistance entre valeurs des variables mesurés et calculées et le raisonnement de diagnostic fondé sur les composants, le premier est fait au niveau du DA et le deuxième au niveau du DDA.

4.4.1.1 L'approche formelle pour l'obtention des relations causales.

L'obtention d'un ensemble d'équations mathématiques décrivant le comportement du système est généralement basée sur les bilans de matière ou les bilans énergétiques. Ceci n'est pas l'objectif final, mais seulement une étape intermédiaire pour arriver à un modèle causal qui représente le procédé. Iwasaki est la première à proposer une analyse systématique pour l'obtention de l'ordonnancement causal d'un ensemble d'équations. La causalité est alors équivalente à la calculabilité [Iwasaki and Simon 1986]. Cette méthode a été appliquée et raffinée par différents chercheurs [Travé-Massuyès and Pons 1997]. Par exemple, Thevenon propose dans [Thevenon and Flaus 2000] une prolongation aux systèmes hybrides.

La méthode généralement appliquée consiste d'abord en la séparation des équations différentielles et algébriques. La causalité dans une équation différentielle est simple à trouver : elle résulte directement de la définition de l'entrée de cette équation (la cause) et de la sortie de l'équation (la conséquence) ; par contre, la causalité dans un ensemble de relations statiques n'est pas élémentaire. Des algorithmes d'ordonnancement causal doivent être employés. Ces algorithmes recherchent pour une séquence de variables dans quel ordre elles doivent être calculées. Heim montre dans [Heim 2003] un exemple d'un algorithme de modélisation causale appliqué à un procédé pétrochimique complexe.

4.4.1.2 L'approche « experte » pour l'obtention des relations causales.

L'obtention de la connaissance causale à partir de la description physique du procédé convient aux fins de la supervision parce qu'elle correspond mieux au point de vue de l'ingénieur de procédé ou de l'opérateur [de Kleer and Brown 1986]. Il est facile pour les ingénieurs de décrire le système technique dont ils sont responsables en terme de ses composants (par exemple vannes, réservoirs, contrôleurs) et conduits (par exemple tuyaux), et de le spécifier en termes de processus physiques généraux (par exemple ébullition, chauffage, etc..) et de donner les équations correspondantes. En outre, cette description topologique d'un système est habituellement disponible sous forme de diagrammes synoptiques et de documents techniques sur le système. C'est donc cette approche que nous avons décidé de faciliter par des outils implémentés dans MAGIC pour la personnalisation des agents.

4.4.2 L'éditeur des graphes causaux.

Dans cette section, un outil logiciel appelé Editeur des Graphes Causaux (CGE) est décrit. Cet éditeur est employé pour la création de graphes causaux de façon interactive à partir d'une interface graphique. En cliquant sur les boutons de la souris, il est possible d'ajouter des nœuds, de créer les arcs liant les nœuds et d'assembler des graphes plus compliqués à partir de graphes simples. Le résultat final d'une session d'édition avec le CGE est un ensemble de graphes assemblés. Les résultats de session sont archivés pour être récupérés dans des sessions postérieures. L'éditeur est basé principalement sur DIVA [DIVA]. DIVA est un progiciel libre fondé sur JAVA. DIVA a été conçu pour la visualisation des espaces dynamiques de l'information et de leurs interactions. Le format XML est employé pour coder tous les résultats : configurations, sous graphes, et graphes assemblés (le codage de XML est hérité du paquet de DIVA et a été étendu pour les buts du CGE). Quelques graphes assemblés sont aussi stockés dans la base de données pour leur utilisation ultérieure dans MAGIC.

L'éditeur est un outil pour faciliter la capture de la connaissance théorique et empirique des experts et des opérateurs humains. L'objectif du CGE est d'aider l'expert du procédé dans la construction d'un ensemble de graphe causaux. La visualisation de ces graphes est également employée en ligne pour l'explication à l'opérateur du comportement global du système. Dans ce sens, le CGE est en même temps un outil pour la conception des modèles causaux, qui sont employés ultérieurement par les algorithmes de supervision, et un outil pour la conception des interfaces de surveillance [Garcia-Beltran, Exel et al. 2003].

La conception du graphe causal prend en considération la possibilité d'une construction modulaire des graphes. Le graphe causal est une représentation des relations

entre les variables du procédé et pas des composants physiques. À mesure que le nombre de variables augmente, c'est souhaitable de décomposer le modèle causal en différents sous-modèles. Ceci rend plus facile la compréhension des interactions des procédés complexes et facilite l'association des variables et des composants. Dans l'approche présentée ici, la décomposition d'un système en sous-systèmes est fondée sur une analyse fonctionnelle du processus, pour relier un sous-graphe à une fonction physique et aux composants respectifs du procédé. Cette décomposition commence avec l'identification des fonctionnalités de niveau supérieur d'un système et postérieurement les décompositions successives sont réalisées jusqu'à arriver aux fonctions de base du processus. À ce niveau, les sous-graphes détaillés sont construits. Les composants physiques peuvent être associés à ces sous-graphes. Dans une deuxième phase, l'assemblage des différents sous-graphes permet la description des fonctions de niveau plus élevé. Cette approche est du type descendant et peut être reliée à l'approche proposée dans [Gentil, Dziopa et al. 1999].

Les variables d'un sous-système et les arcs sont conditionnés par un mode de fonctionnement. Inversement, la perte d'un de ces liens peut montrer la perte de validité d'un mode de fonctionnement. Donc, un ensemble de variables et d'arcs est reliée à un mode de fonctionnement, donc au niveau fonctionnel le sous-graphe doit être considéré comme une seule entité. De cette façon, le concept de sous graphe et de la configuration de sous graphe est présenté. Dans ce sens, un sous-graphe est un ensemble de variables et d'arcs qui modélisent le comportement d'un sous-système du procédé dans une configuration donnée. Ainsi, afin de modéliser différents modes d'opération du même sous-système, il est nécessaire de construire différentes configurations de sous graphe pour ce sous-système.

La fenêtre graphique principale de l'interface utilisateur du CGE est divisée en quatre régions (Figure 4-9):

Barre de menu et barre d'outils. Les fonctions définies dans les menus et leurs raccourcis dans la barre d'outils sont employées pour les opérations de création, d'édition et de stockage des graphes dans des fichiers temporaires. Les fonctions typiques du menu de dossier sont *Nouveau sous-graphe, nouvelle configuration, ouvrir, fermer, enregistrer*; les fonctions typiques du menu d'édition sont : *couper, coller et défaire*. Ces fonctions permettent la création d'un nouveau sous-graphe, d'une nouvelle configuration ou d'un nouveau graphe assemblé, à partir de sous-graphes existants.

Vue Hiérarchique. Dans cette zone, la structure hiérarchique des graphes est présentée comme un arbre d'éléments. Un ensemble de graphes contient une liste de sous-graphes et chaque sous-graphe une liste de configurations. Une configuration est une feuille dans cet arbre. On montre chaque graphe (un ensemble de noeuds et d'arcs) dans une vue de l'espace de travail.

Espace de travail. C'est ici que les différentes configurations d'un sous-graphe et les graphes assemblés sont montrés et édités. Le concepteur travaille sur deux types de vues :

- *Vue de configuration* : dans cette vue est montrée et éditée *une configuration de sous-graphe*. La vue contient la configuration, dans sa partie gauche un ensemble de régions de capture pour l'entrée d'information reliée aux arcs et/ou noeuds. Un noeud est décrit par son nom, une brève description textuelle et un ensemble de paramètres qui définissent les seuils pour la décision de diagnostic (§4.3.6 et Figure 4-7). De la même

- L'agent de diagnostic fondé sur les graphes causaux (CBDA). Cet agent utilise le graphe causal comme modèle de référence pour la génération de symptômes.
- L'agent d'aide à décision (DSA) utilise le graphe causal pour la prédiction de la propagation de défauts dans le procédé et pour la recherche de conseils d'action face à un défaut.
- L'agent d'interface pour l'opérateur (OIA) utilise la représentation visuelle des graphes causaux pour l'explication à l'opérateur.

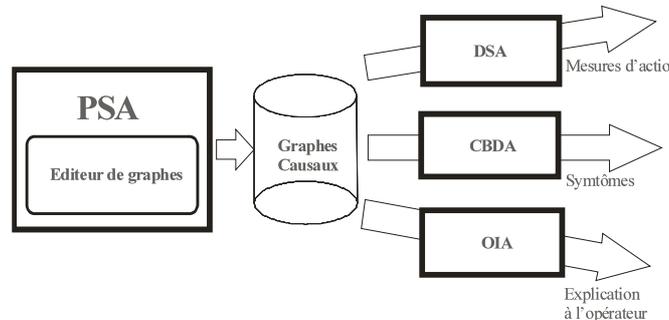


Figure 4-10. Utilisation de graphes causaux dans le cadre de MAGIC

4.4.3 Identification de fonctions de transfert en utilisant le GPMAS.

Afin d'identifier les fonctions de transfert liées aux arcs, on utilise un autre outil de MAGIC, le GPMAS (General Purpose Mathematical Server). Dans le GPMAS chaque agent de diagnostic a ses propres algorithmes pour la création des modèles. Ainsi, dans le cas spécifique de CBDA, on utilise une interface graphique adaptée pour la gestion des structures associées aux graphes causaux (

Figure 4-11). L'identification des fonctions de transfert est fondée sur l'utilisation des méthodes proposées par la boîte à outils d'identification de MATLAB®, particulièrement les méthodes d'identification par erreur de sortie (*oe* : output error).

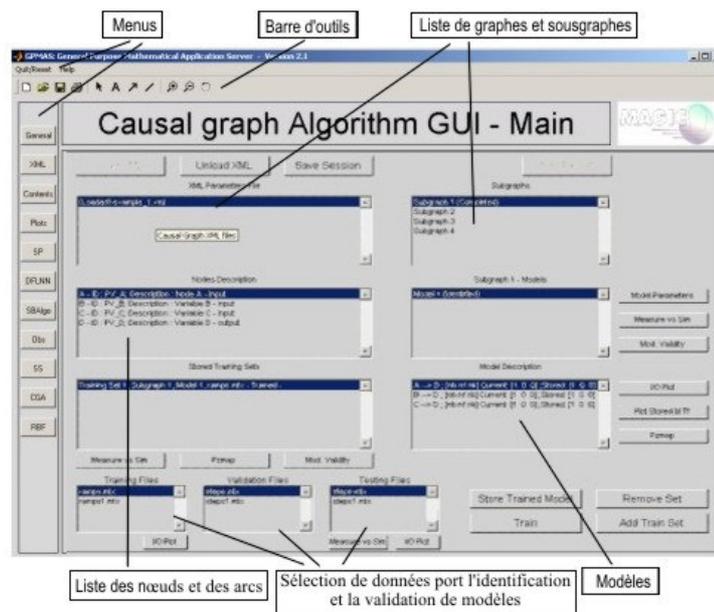


Figure 4-11. Vues globales de l'interface graphique du GPMAS dans l'option d'identification des arcs du graphe causal.

L'ingénieur accède à distance au GPMAS par le moyen d'une interface sur l'écran où le PSA est installé. La communication entre le PSA et le GPMAS permet de transmettre, du PSA vers le GPMAS, les données nécessaires pour l'identification et la validation des modèles et le fichier XML qui contient la structure du graphe causal, mais qui ne contient pas les paramètres des modèles. Le processus d'identification associé au graphe causal est réalisé de façon progressive, car dans le cas général le graphe causal inclut plusieurs modèles locaux, et c'est à ce niveau où les fonctions de transfert sont associées. La première étape dans le processus d'identification est la présentation à l'utilisateur de la liste de sous-graphes et de modèles locaux, dont il fera la sélection un à un. La procédure d'identification proprement dite est similaire à l'interface de la boîte à outil d'identification de MATLAB. On sélectionne les données pour l'identification, et après on sélectionne les ordres du numérateur et dénominateur de(s) la(s) fonction(s) de transfert (selon qu'il s'agit d'un système mono ou multi-entrées).

Différentes vues sont disponibles pour la vérification des modèles, comme le diagramme de pôles-zéros et les traces de comparaison entre la sortie du modèle et la vraie mesure. La validité du modèle *a priori* telle qu'elle est décrite dans la section 4.3.4.1 est calculée ici.

À la fin d'une session d'identification, le GPMAS transmet en retour le fichier XML qui décrit le graphe causal, mais avec les paramètres des modèles et la validité *a priori* de chacun d'eux.

4.4.4 Obtention des tests de détection à partir du graphe causal.

Inversement à ce qu'il arrive avec les autres algorithmes de détection de MAGIC, la création des Tests de Détection est effectuée à partir du graphe causal et non à partir des composants. En conséquence, la première étape consiste en la création d'un graphe causal pour le diagnostic ; à partir de là, il est possible de produire la liste des tests de détection qui peuvent être appelées dans le CBDA. La liste des tests de détection correspond à la liste des tests globaux et des tests locaux au niveau de chaque variable endogène. La quantité totale de tests de détection effectués grâce à un graphe causal est donnée par la formule suivante :

Nombre de tests de détection = (nombre de nœuds - nœuds associés aux variables exogènes)*2.

4.5 Exemples académiques.

Dans ce qui suit, un simulateur d'un système de quatre bacs (Figure 4-12) est employé comme exemple pour illustrer diverses fonctionnalités du CBDA. Les réservoirs sont reliés entre eux par différents tuyaux. L'eau peut remplir les bacs de différentes manières qui peuvent être commandées par des servo-vannes, des électro-vannes du type "tout/rien" ou des vannes manuelles.

4.5.1 Description du système de bacs.

Le simulateur de bacs correspond à un système de bacs réel qui a été conçu pour des besoins académiques : différentes manipulations, orientées vers l'enseignement des systèmes de commande peuvent être effectuées.

Le but le plus sophistiqué est la commande multivariable de débit et température. Pour atteindre ce but, la configuration nominale utilise les deux servo-vannes V_{k1} et V_{k2} . Les mesures disponibles pour les systèmes de commande et de surveillance sont les niveaux d'eau h_1 , h_2 , h_3 et h_4 , le débit de sortie de chaque bac q_1 , q_2 , q_3 , q_4 et les mesures de température de l'eau t_1, t_2 et t_3 correspondant aux réservoirs 1, 2, 3 respectivement. Dans cette configuration, la vanne V_{k1} est utilisée pour réguler le débit d'entrée du Bac 1, qui est rempli avec de l'eau froide, et d'une manière symétrique, le débit de l'entrée au Bac 2 est commandé à l'aide de la vanne V_{k2} , mais dans ce cas-ci de l'eau chaude est employée. Le Bac 3 est employé pour mélanger l'eau froide et l'eau chaude. Donc, dans le cas de cette configuration, le système est commandé pour fournir un débit d'eau continu q_3 à une température t_3 .

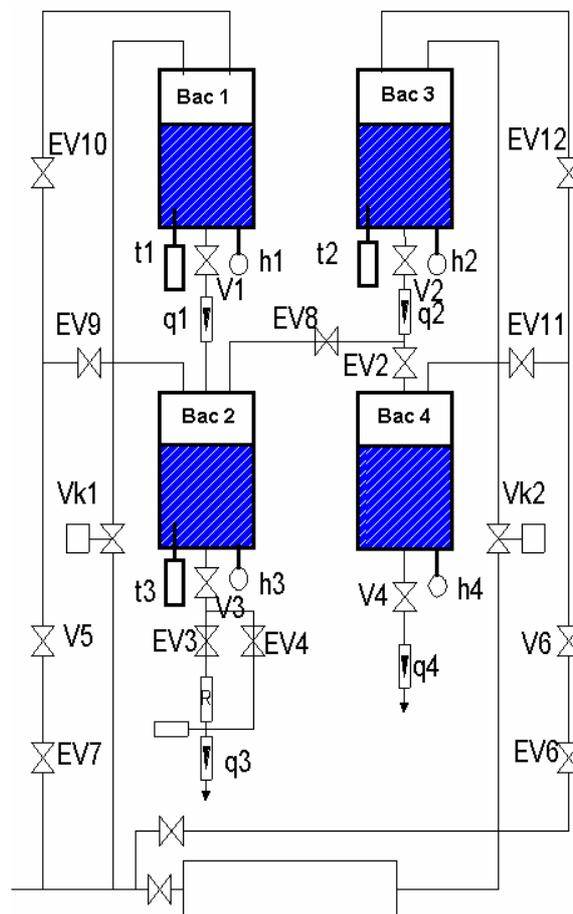


Figure 4-12. Le système complet des bacs.

Chaque configuration est atteinte grâce à la combinaison d'ouverture ou de fermeture des vannes tout/rien et une configuration particulière dans le système de commande. Dans l'opération nominale, le débit et la température de sortie sont commandés par un ensemble de contrôleurs et les servo-vannes V_{k1} et V_{k2} ; seuls les bacs 1, 2 et 3 sont en service. Les vannes manuelles V_1 à V_4 sont totalement ouvertes, et V_5 est fermée. Les électrovannes

EV2, EV7, EV9, EV10, EV6, EV11, EV12 sont fermées et EV8, EV3 sont ouvertes. Du matériel redondant est présent dans le système. Le Bac 4 est employé comme composant de sécurité. Des sous-systèmes alternes d'alimentation ont aussi été considérés.

4.5.2 Tests de détection implémentés dans le CBDA.

Dans les sections suivantes, on montre quelques tests de détection effectués sur une des configurations possibles du système, dans ce cas-là, la configuration de deux bacs en cascade (Figure 4-13), ce qui est une des configurations de base du système.

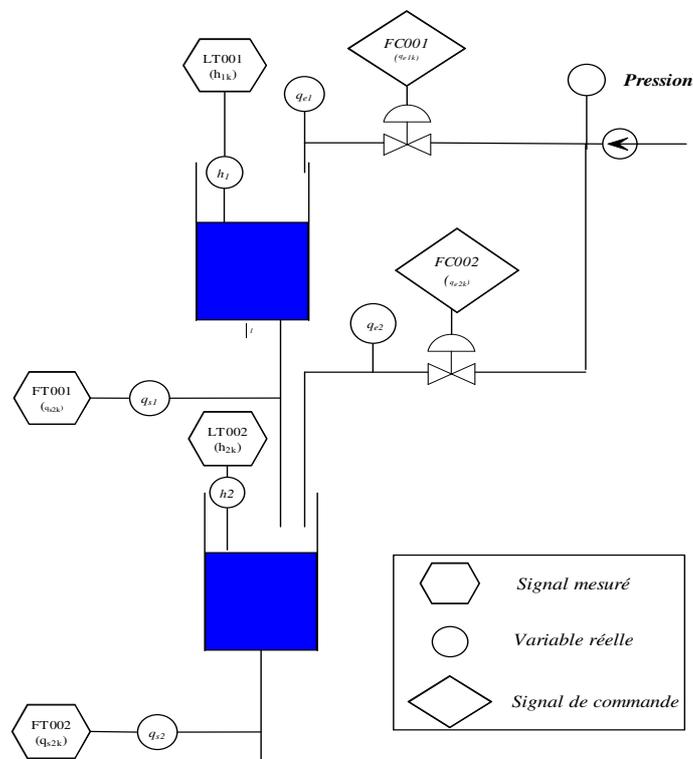


Figure 4-13. La configuration de deux Bacs en cascade.

En suivant la méthodologie de MAGIC, la première étape est l'obtention de la description du système physique liée aux composants physiques, états des composants et comportements des composants.

Tableau 4-1: Composants, états et comportements de la configuration de deux bacs.

Composant	Etats du composants	Description du comportement du composant.
C1 : Servo-vanne du Bac supérieur.	Normal.	$[C1_S1_CB1] CV001 = q_{e1}$
C2 : Bac supérieur (Bac 1)	Normal	$[C2_S1_CB1] S \frac{dh_1}{dt} = q_{e1} - q_{s1}$, valide si $0 < h_1 < 50\text{cm}$.
C3 : Restriction de sortie du Bac 1	Normal	$[C3_S1_CB1] q_{s1} = kh_1$,
C4 : Capteur de niveau du Bac 1	Normal	$[C4_S1_CB1] LI001 = h_1$,
C5 : Servo vanne de Bac inférieur	Normal	$[C5_S1_CB1] CV002 = q_{e2}$
C6 : Bac inférieur (Bac 2)	Normal	$[C6_S1_CB1] S \frac{dh_2}{dt} = q_{e2} + q_{s1} - q_{s2}$, valide si $0 < h_2 < 50\text{cm}$
C7 : Restriction de sortie du Bac 2	Normal	$[C7_S1_CB1] q_{s2} = kh_2$
C8 :Capteur de niveau du Bac 2	Normal	$[C8_S1_CB1] LC002 = h_2$,
C9 : Capteur de débit de sortie du Bac1	Normal	$[C9_S1_CB1] FI001 = q_{s1}$
C10 : Capteur de débit de sortie du Bac 2	Normal	$[C10_S1_CB1] FI002 = q_{s2}$

Un modèle causal qui représente le comportement du système est créé en utilisant les variables physiques connues (FC001, FC002, LT001, FT001, LT002 et FT002). La Figure 4-14 montre le graphe construit, les composants physiques associés aux arcs et aux noeuds.

Le Tableau 4-2 montre aussi les relations, ainsi que les différents tests implémentés grâce à ce modèle causal. On remarquera que dans cette table, on fait apparaître clairement la différence entre variable réelle et valeur mesurée, ce qui met en valeur les composants capteurs.

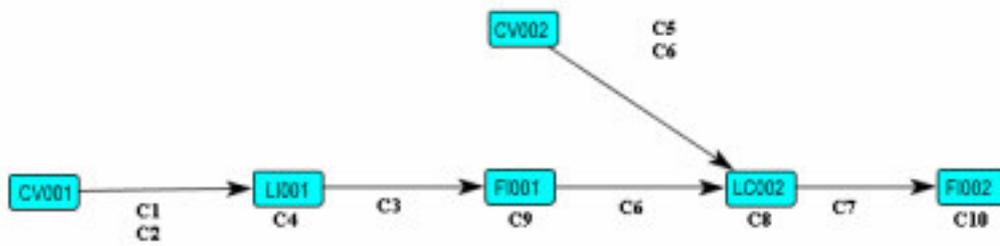
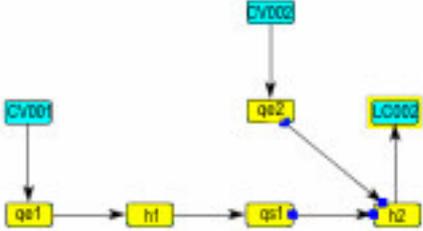
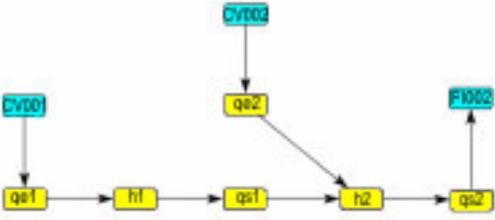


Figure 4-14. Graphe causal associé à la configuration de deux bacs en cascade.

À partir du graphe causal il est possible de générer huit tests de détection différents, chacun avec un support différent.

Tableau 4-2: Liste de tests de détection.

Test de détection	Sous graphe équivalent	Support.
DetCausalLocal_01		C1 C2 C4
DetCausalLocal_02		C3 C4 C9
DetCausalLocal_03		C5 C6 C8
DetCausalLocal_04		C7 C8 C10
DetCausalGlobal_01		C1 C2 C4

<p>DetCausalGlobal_02</p>		<p>C1 C2 C3 C9</p>
<p>DetCausalGlobal_03</p>		<p>C1 C2 C3 C5 C6 C8</p>
<p>DetCausalGlobal_04</p>		<p>C1 C2 C3 C5 C6 C7 C10</p>

Foncé = variables mesurées, Clair = variables non connues.

4.5.2.1 Scénario 1: Simulation d'une fuite progressive dans le Bac 1.

Le scénario décrit par la **Figure 4-15** (et les suivantes) débute avec un changement de point de consigne du niveau LC002, suivi d'un défaut. Une fuite progressive (rampe) est appliquée au Bac1, simulant un défaut progressif appliqué à partir du temps 10000 sec. La Figure 4-15 montre les différentes variables mesurées (LI001, LC002, FI001, FI002) et le signal de commande (CV001).

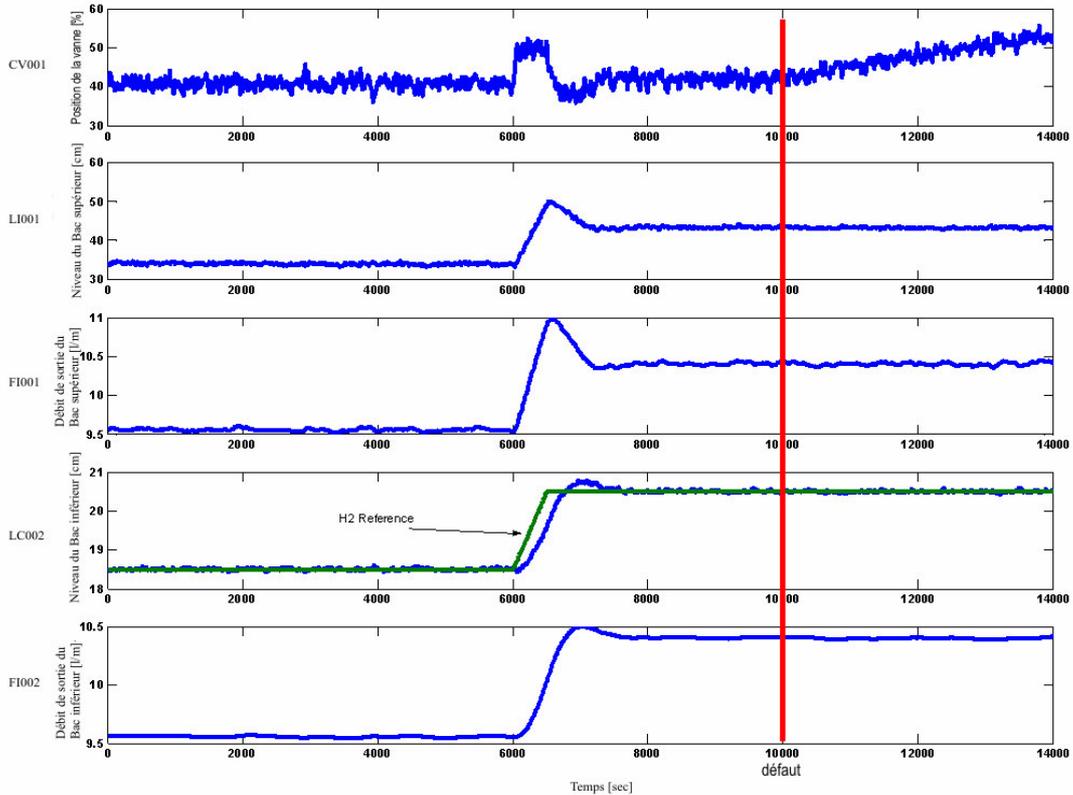


Figure 4-15. Fuite graduellement croissante du Bac 1.

Le niveau du Bac 2 est maintenu à sa valeur de référence grâce à la régulation du système de commande qui augmente l'ouverture de la vanne CV001 pour compenser la fuite. En utilisant les différents tests de détection associés au graphe causal, plusieurs symptômes sont générés (Figure 4-16). On peut observer que les valeurs de la décision des symptômes associées au test *DetCausalLocal_01* arrivent à 1.0 après le défaut (Figure 4-16-e). On peut aussi observer que les tests globaux 1-4 (Figure 4-16 a,b,c,d) suggèrent un comportement anormal du système. Au moyen du raisonnement de localisation, on peut conclure que les composants suspectés sont la servo-vanne 1 et le Bac 1 (voir Tableau 4-2), le capteur de niveau du Bac supérieur est exonéré grâce au test *DetCausalLocal_02* qui est cohérent.

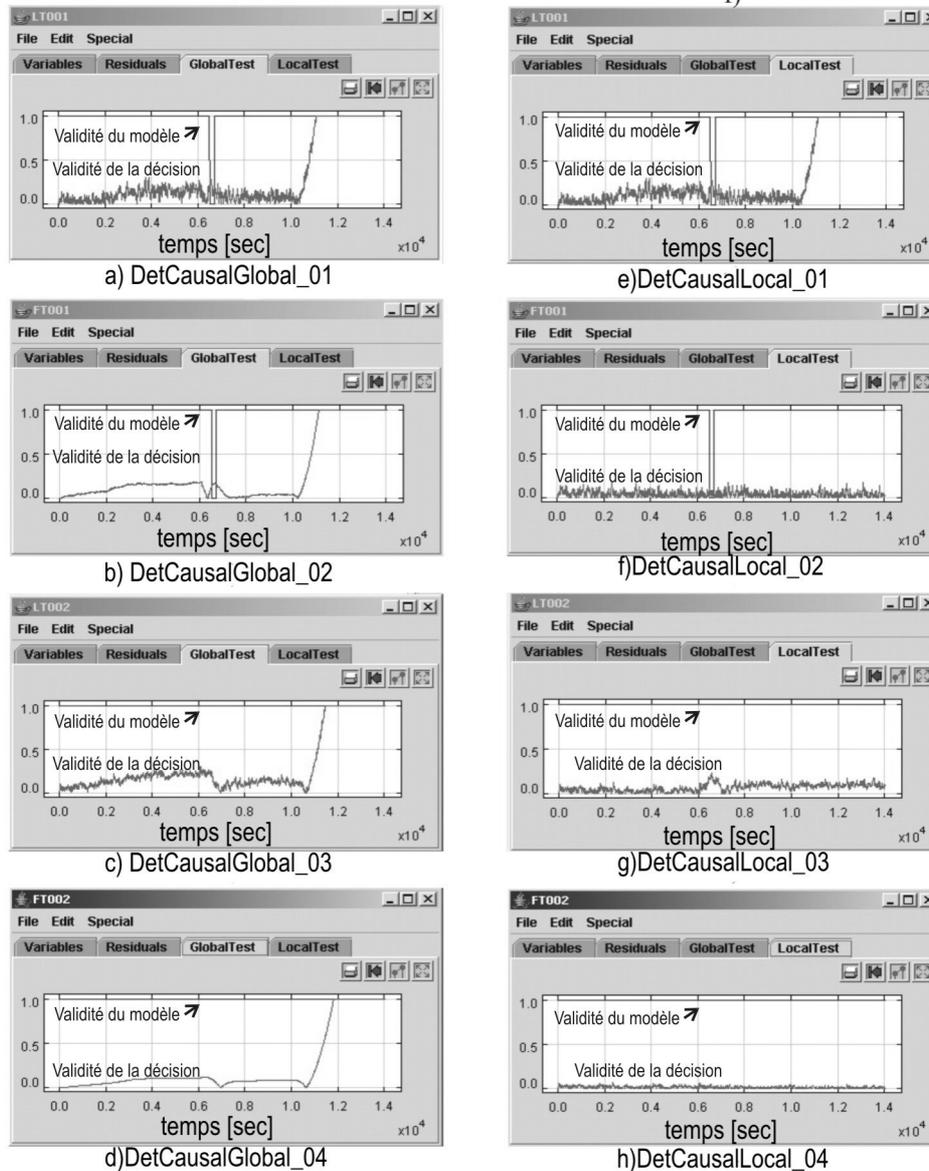


Figure 4-16. Scénario 1: Simulation d'une fuite progressive dans le Bac 1, symptômes générés par le CBDA : de a) à d) tests globaux, de e) à h), tests locaux.

4.5.2.2 Scénario 2 : Simulation d'une fuite brusque en échelon du Bac 1.

Dans ce scénario, une fuite est appliquée au Bac 1 simulant un défaut brusque au temps 10000 sec. La Figure 4-17 montre les différentes variables mesurées (LI001, LC002, FI001, FI002) et le signal de commande (CV001). Comme la valeur réelle de LI001 n'est pas conforme à l'ouverture CV001, le système de diagnostic conclut encore que le Bac 1 ou la servo-vanne 1 sont les composants défectueux.

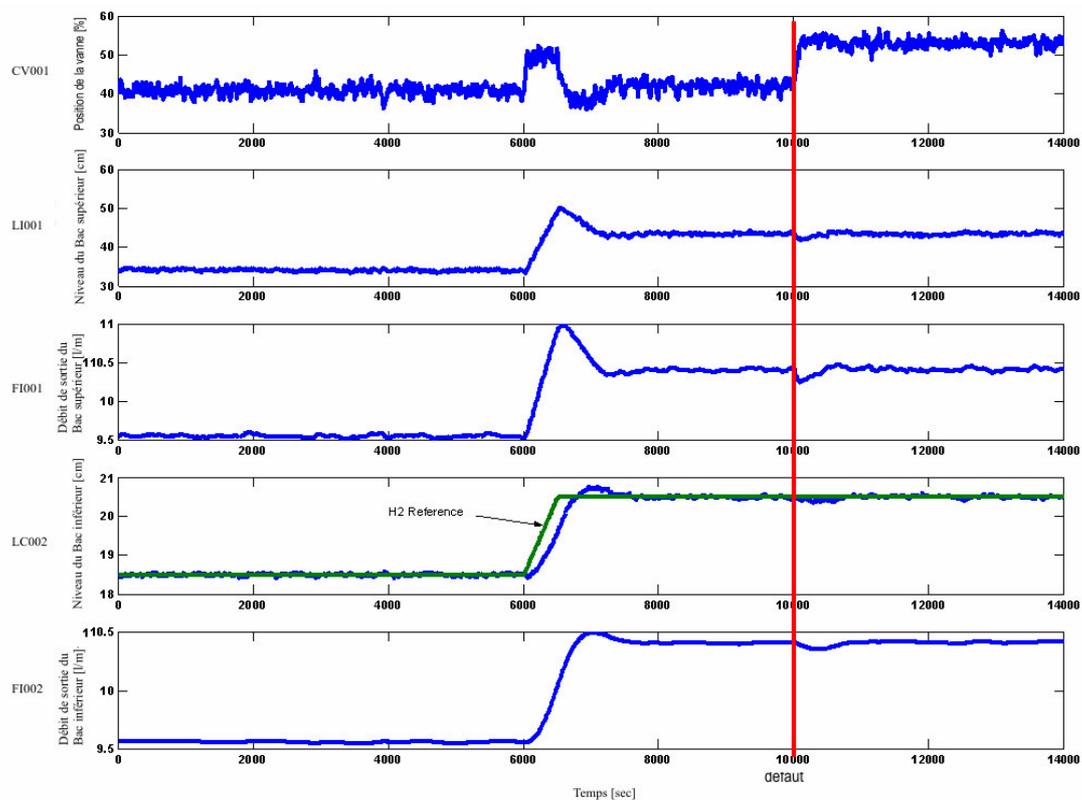


Figure 4-17. Fuite brusque du Bac 1.

Puisque le défaut n'est pas très grand, le système de commande a compensé la fuite par plus de débit d'entrée au Bac 1.



Figure 4-18. Scénario 2: Simulation d'une fuite brusque du Bac 1, symptômes générés par le CBDA: de a) à d) tests globaux, de e) à h), tests locaux.

4.5.2.3 Scénario 3 : Simulation d'une fuite progressive du Bac 2.

Une fuite progressive est appliquée au Bac2 au temps 8000 sec, simulant un défaut progressif. La *Figure 4-19* montre les différentes variables mesurées (LI001, LC002, FI001, FI002) et le signal de commande (CV001).

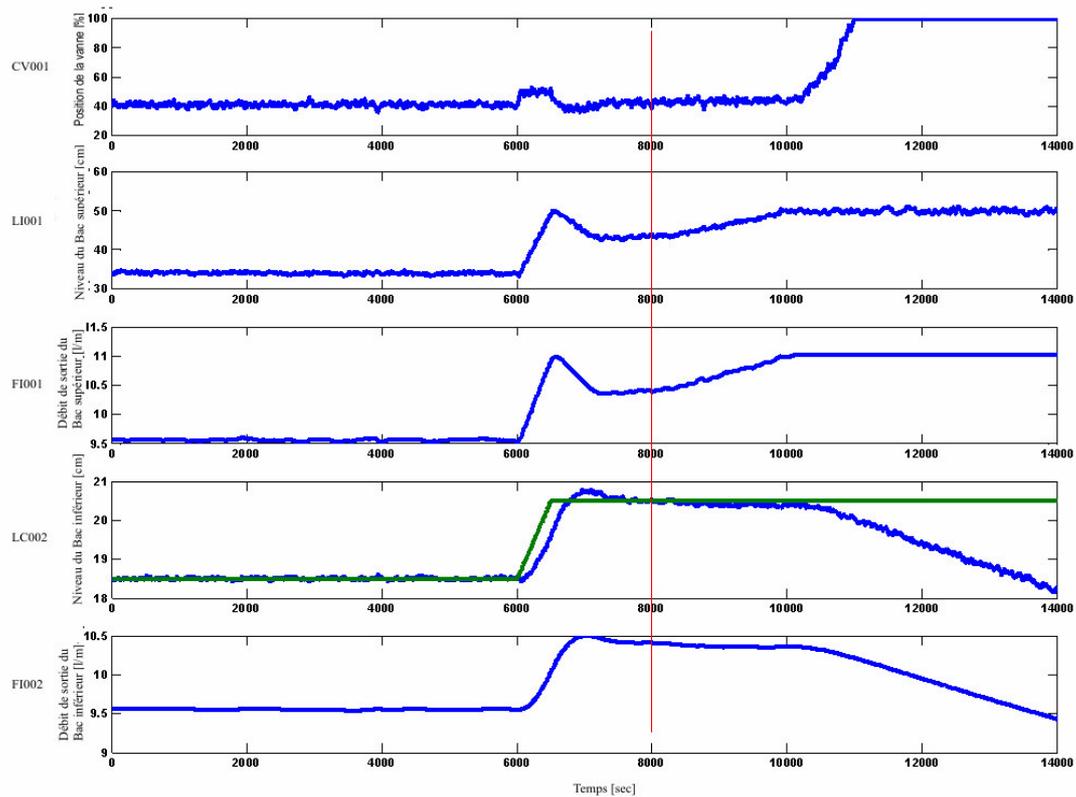


Figure 4-19. Fuite graduellement croissante du Bac 2.

Le niveau du Bac 2 est maintenu à sa valeur de référence parce que le système de commande compense la fuite en augmentant l'ouverture de la servo-vanne CV001. Comme la valeur réelle du niveau LC002 n'est pas conforme au débit FI001, le système de diagnostic conclut que le Bac 2, la deuxième servo-vanne sont les composants défectueux. Le capteur de niveau du Bac inférieur (C8) et le capteur de débit de sortie du Bac inférieur (C9) sont exonérés par le test *DetCausalLocal_04* et le test *DetCausalLocal_02* respectivement.

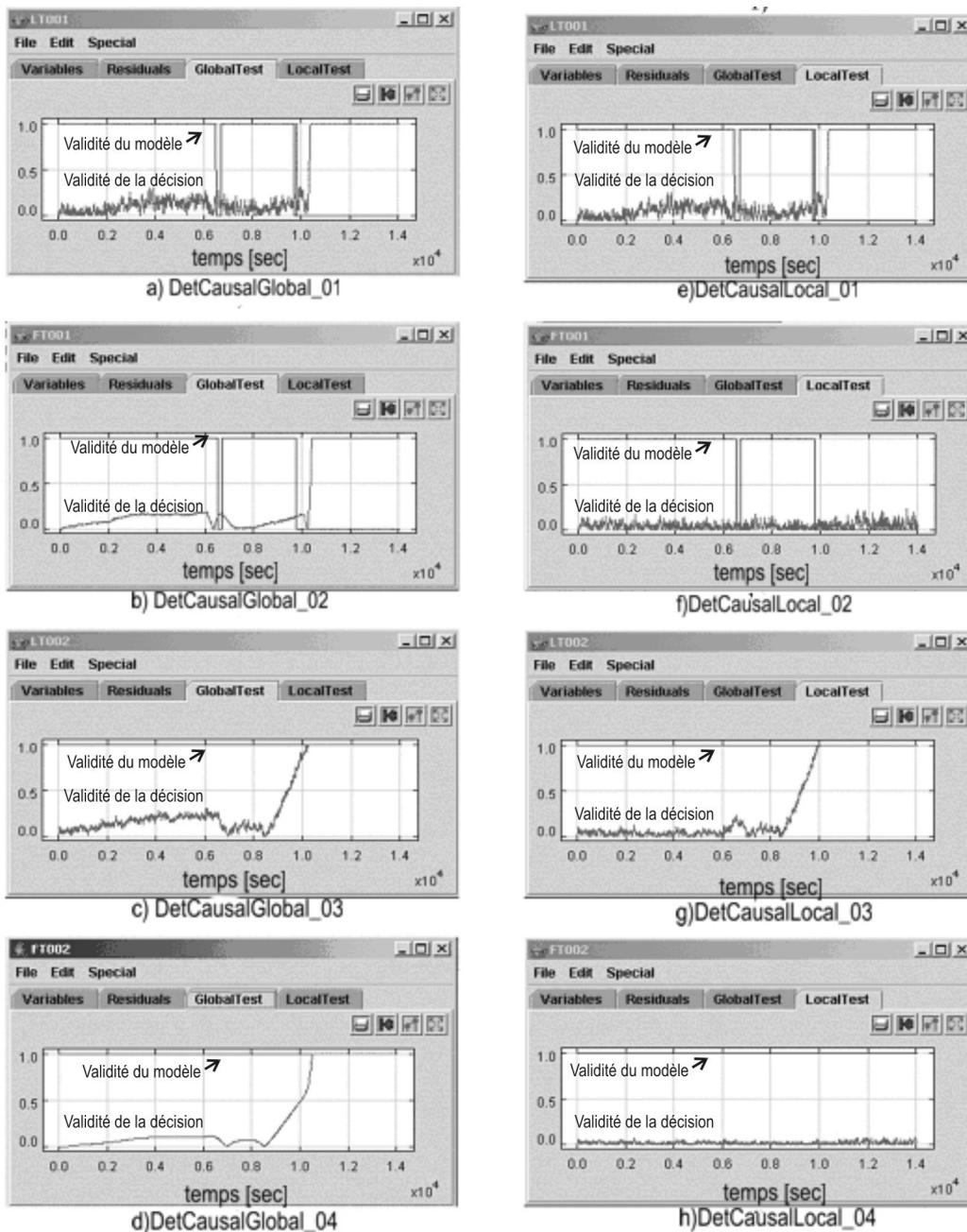


Figure 4-20: Scénario 3: Fuite progressive dans le Bac 2, symptômes générés par la CBDA: de a) à d) tests globaux, de e) à h), tests locaux

4.6 Conclusion.

Les fonctionnalités d'un agent de détection sont distribuées dans deux couches, la couche de communication qui se charge de l'échange de l'information et la couche spécifique, consacrée seulement aux algorithmes de diagnostic.

Dans ce chapitre la notion de symptôme telle qu'elle est utilisée dans MAGIC est d'abord définie. Ensuite la méthode générale pour l'obtention de symptômes à partir d'un graphe causal est décrite. Cet algorithme se base sur la création dynamique des valeurs de référence de bon comportement pour toutes les variables endogènes dans le graphe causal. Deux types de test sont menés pour chaque nœud dans le graphe. Un test local basé sur la comparaison de la valeur mesurée de la variable avec la prédiction locale, produite à partir des valeurs mesurées des antécédents directs de la variable. Un second test compare la valeur mesurée et la valeur de prédiction globale, produite à partir des valeurs des variables exogènes liées causalement avec la variable. La décision finale est obtenue par la transformation des résidus en un ensemble flou représentant le symbole flou « alarme ». Une analyse supplémentaire permet d'évaluer la pertinence du modèle utilisé pour la génération des symptômes. Le même algorithme de génération de symptômes est répété pour tous les nœuds endogènes du graphe.

L'interprétation des résultats des tests globaux et des tests locaux est différente ; les premiers doivent être interprétés comme la détection de défauts, et les deuxièmes comme leur localisation.

Les propriétés de localisation de défaut en utilisant le graphe causal ont été expliquées, bien que cette fonctionnalité n'ait pas été exploitée dans le système MAGIC puisque cette tâche est renvoyée au DDA. Il est intéressant cependant de la prendre en considération, car on a montré que les tests locaux, dont le support comporte un nombre réduit de composants, vont grandement faciliter la tâche de celui-ci.

L'utilité d'un système de ce type au niveau industriel peut fortement être basée sur les outils utilisés pour faciliter sa configuration et son adéquation au procédé spécifique. Par conséquent, un soin particulier a été mis dans la création d'outils de configuration faciles d'emploi.

Les essais effectués avec le simulateur de deux bacs (prototype de laboratoire) permettent d'observer la manière dont fonctionnent les algorithmes causaux pour la détection de défauts.

Chapitre 5

L'agent d'aide à la décision.

5.1 Introduction

L'objectif final d'un système de supervision avancé est d'aider à résoudre les problèmes provoqués par les défauts. Une première étape pour trouver des solutions est de montrer, grâce aux algorithmes de diagnostic, qu'une défaillance est présente sur un composant spécifique. Le diagnostic doit être justifié d'une façon très claire à l'opérateur afin de le convaincre de l'utilité d'un système d'aide à la décision. Une deuxième étape peut être de le conseiller à propos de ce qu'il faut faire pour empêcher que le défaut se développe en défauts plus sérieux qui pourraient arrêter la production ou présenter un risque pour l'environnement. Par conséquent, des outils peuvent être mis en œuvre pour aider la décision de l'opérateur au niveau de la prévision de l'évolution d'une situation et de la conduite qui en résulte ainsi que pour aider le technicien du procédé à gérer sa stratégie de maintenance.

Afin de faire face aux défauts, différentes conditions de sécurité sont exigées du système SCADA (System of Control and Data Acquisition). Un besoin de base pour un système SCADA pour faire face aux situations anormales est l'exécution des procédures d'arrêt d'urgence, consistant à arrêter toutes les unités du procédé suivant un ordre d'exécution des instructions prédéfini. Un autre dispositif de base des systèmes SCADA est l'exécution automatique de la commutation d'équipement, basée sur la redondance physique,

pour faciliter le rétablissement du système après la détection et la localisation du défaut. Ceci consiste en un remplacement automatique d'un composant physique défectueux par un autre sans défaut, par exemple, la commutation entre deux vannes fonctionnant en parallèle. À l'autre extrémité, une fonctionnalité plus avancée pour les systèmes de supervision, est l'accommodation automatique du système de commande après les défauts. L'accommodation consiste à changer des paramètres ou la structure du contrôleur afin d'éviter les conséquences des défauts dans les boucles de régulation. L'accommodation automatique de défauts est encore à un stade de recherche, et n'est pas envisagée dans le projet MAGIC.

Le niveau d'aide à la décision pour l'opération dans MAGIC est situé entre les extrémités mentionnées dans le paragraphe antérieur. L'utilisation d'opérateurs humains spécialisés est une manière efficace de faire face aux problèmes de supervision, surveillance et maintenance dans l'industrie. Les opérateurs sont formés périodiquement pour apprendre et comprendre que faire dans chaque situation de défaut. Néanmoins, dans quelques situations, la tension psychologique peut causer des mauvais choix de la part des opérateurs face à une situation anormale; il est aussi possible que l'opérateur n'arrive pas à prendre sa décision dans le temps requis. Un premier objectif de l'agent d'aide à la décision (**DSA**) est d'aider l'opérateur dans ces situations anormales où il doit faire face à des événements imprévus pendant le fonctionnement du procédé et prendre une décision afin de maintenir la production et éviter un arrêt d'urgence. Un deuxième objectif du **DSA** est d'aider l'opérateur pour la planification d'une politique raisonnable de maintenance.

Après avoir rappelé brièvement l'architecture d'un agent dans MAGIC, nous discuterons les différentes fonctionnalités de l'aide à la décision. Nous discuterons particulièrement la fonction de pronostic. Ensuite nous montrerons comment la connaissance causale implémentée pour le diagnostic peut être utilisée pour l'aide à la décision. Nous évoquerons rapidement la configuration de l'agent DSA et nous terminerons par l'exemple simple des bacs, déjà utilisé au chapitre 4.

5.2 Architecture interne de l'agent d'aide à la décision.

La structure du DSA se compose de trois couches (Figure 5-1) comme tous les agents de MAGIC. La couche "supérieure" est reliée aux couches "inférieures" via une interface IDL (langage de développement d'interfaces) CORBA. Les trois couches sont :

- La **couche spécifique du DSA** qui met en œuvre les algorithmes d'aide à la décision. Ces algorithmes sont responsables des conseils à l'opérateur et/ou l'EUSE dans les situations anormales.
- La **couche de communication spécifique du DSA** fournit les services de communication spécifiques et nécessaires pour la couche "supérieure" du DSA, et pas directement fournis par la couche "inférieure" 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) et permet au DSA d'être intégré dans l'infrastructure MAGIC.

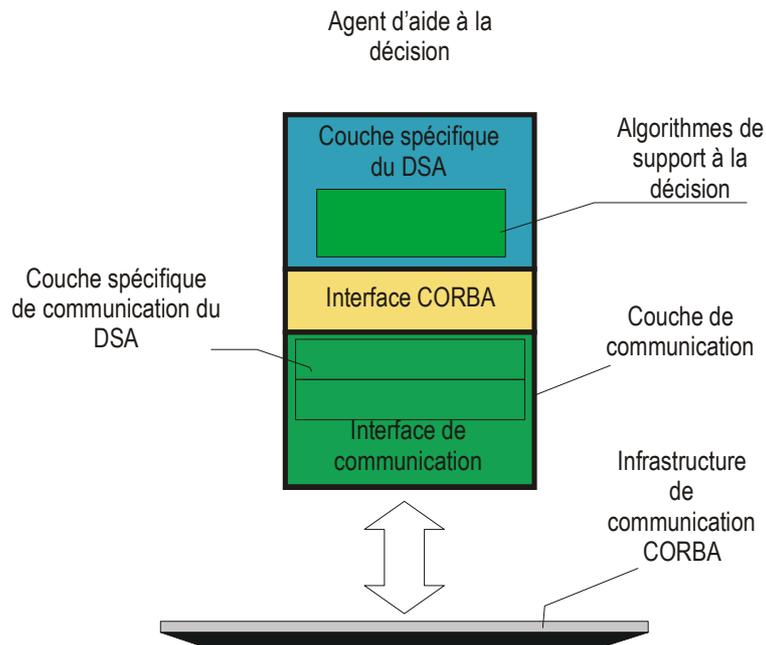


Figure 5-1. Structure interne de l'agent d'aide à la décision

5.3 Fonctionnalités de l'agent d'aide à la décision.

En tant qu'agent d'aide à la décision, le DSA doit fournir diverses fonctionnalités (Figure 5-2). Parmi toutes ces fonctionnalités, on discutera ici seulement celles qui sont liées directement aux algorithmes implantés dans la couche spécifique du DSA ; les algorithmes liés à la communication ont été déjà discutés de façon générale dans le chapitre trois.

Afin d'aider l'opérateur dans la prise de décision, le DSA intègre trois outils d'aide à la décision dans les phases d'anticipation de l'évolution du système, d'interprétation des conséquences des défauts et de définition d'actions à réaliser pour la conduite et la maintenance (voir chapitre 2):

- Pronostic :
 - de la tendance des variables,
 - de propagation de défaut dans le procédé.
- Conseils pour l'aide à la politique de maintenance.
- Conseil d'action pour l'aide à la conduite du procédé.

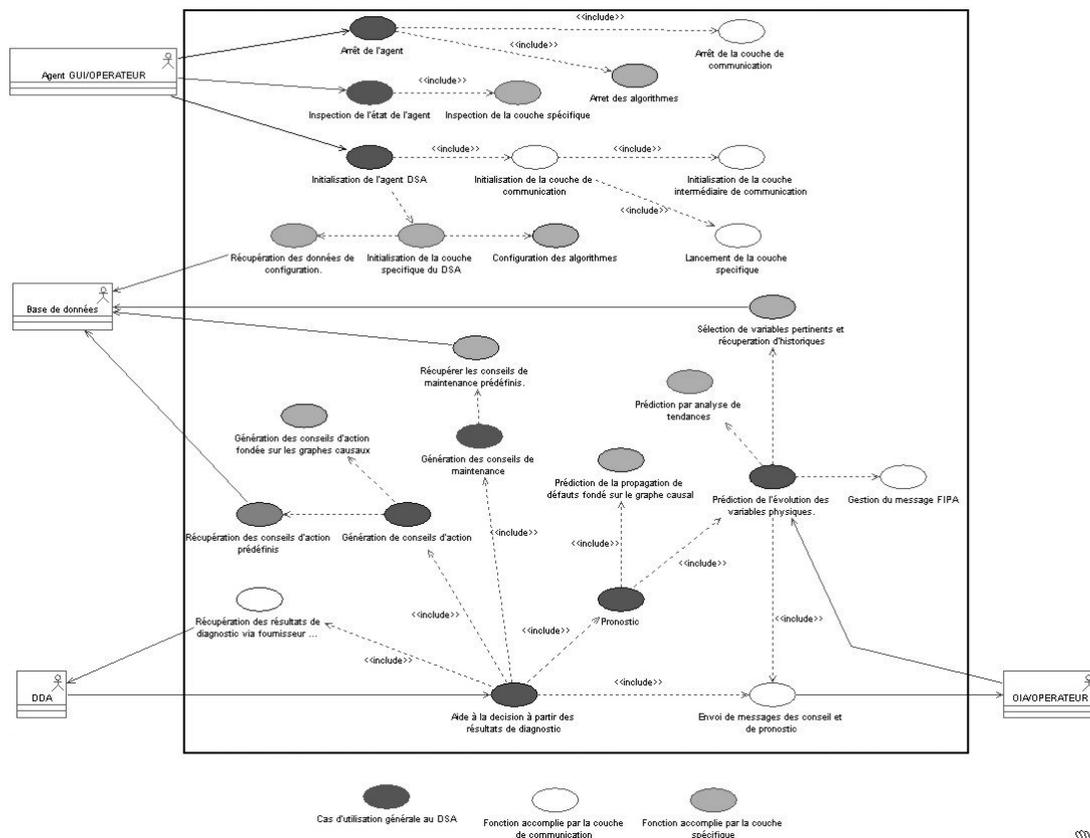


Figure 5-2. Les cas d'utilisation du DSA.

5.4 Pronostic.

Le but principal du pronostic est d'anticiper et d'empêcher le développement de défauts critiques. Les résultats du pronostic sont employés pour la prise de décisions sur les actions préventives avec le but économique de maximiser la durée de vie des composants remplaçables, tout en réduisant au minimum les risques opérationnels [Mathur, Cavanaugh et al. 2001]. On a déjà expliqué l'utilité de la prédiction dans les cadres de la conduite et la maintenance (§2.3.5), donc, dans la suite, on abordera directement les algorithmes de prédiction proposés.

5.4.1 Les réseaux neuronaux pour la prédiction de séries temporelles.

Un ANFIS (Système Adaptatif d'Inférence Neuro-Flou) est un type de réseau de neurones équivalent à un système d'inférence floue. Un ANFIS peut être classifié comme un type de réseau à fonctions de base radiale (RBFN) et c'est en même temps un équivalent du modèle flou de Sugeno [Jang 1993].

5.4.1.1 Structure de l'ANFIS

Considérons un petit système flou de type Sugeno avec seulement deux règles (si-alors):

$$\mathfrak{R}_1 : \text{si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } B_1 \text{ alors } y = p_{11}x_1 + p_{12}x_2 + p_{13} \quad (5.1)$$

$$\mathfrak{R}_2 : \text{si } x_1 \text{ est } A_2 \text{ et } x_2 \text{ est } B_2 \text{ alors } y = p_{21}x_1 + p_{22}x_2 + p_{23}$$

l'architecture correspondante d'ANFIS pour ce système est représentée sur la Figure 5-3 :

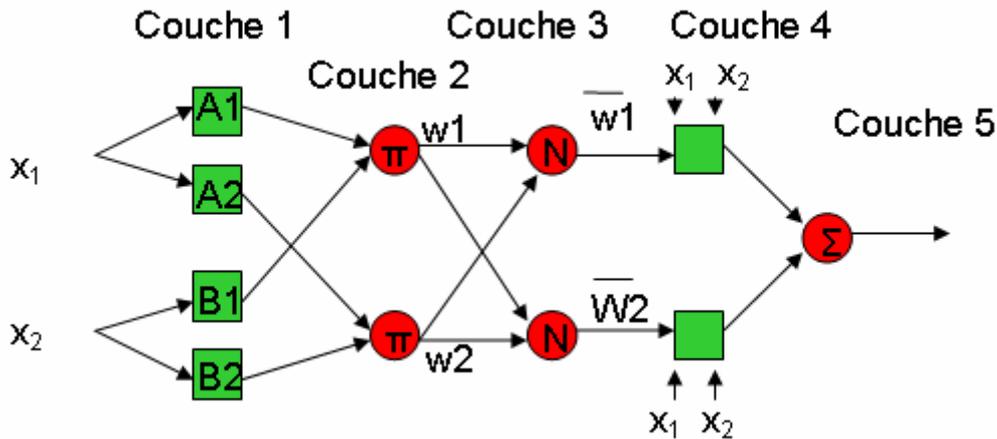


Figure 5-3 L' architecture de l' ANFIS

Tous les nœuds dans une couche donnée exécutent la même fonction. Les équations suivantes sont écrites pour l'exemple de 2 règles de (5.1). O_i^j représente la sortie i des nœuds de la couche j . Ainsi, la sortie des nœuds de la première couche est :

$$O_i^1 = \mu_{A_i}(x) \quad (5.2)$$

où x est l'entrée du i ème nœud de la couche 1 et A_i est l'étiquette linguistique associée à cette fonction de nœud. Une fonction généralisée du type cloche (Figure 5-4) est employée comme fonction d'appartenance $\mu_{A_i}(x)$:

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad (5.3)$$

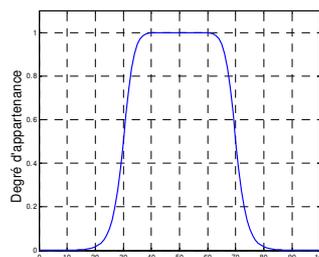


Figure 5-4. Fonction d'appartenance du type cloche.

Les noeuds de la deuxième couche exécutent l'évaluation de la prémisse des règles. Tous les opérateurs du type T-Norme pourraient être employés ; ici, un opérateur de multiplication a été choisi :

$$O_i^2 = w_i = \mu_{A_i}(x_1) \cdot \mu_{B_i}(x_2) \quad (5.4)$$

La troisième couche effectue la normalisation :

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2 \quad (5.5)$$

Les noeuds de la quatrième couche évaluent la partie conséquent de chaque règle. En accord avec le modèle flou de Sugeno, le conséquent de la règle est une combinaison linéaire des entrées:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_{i1}x_1 + p_{i2}x_2 + p_{i3}) \quad (5.6)$$

où $\{p_{i1}, p_{i2}, p_{i3}\}$ est l'ensemble de paramètres modifiables de cette couche.

Le noeud unique dans la couche 5 calcule la sortie globale :

$$y = O_1^5 = \sum_i O_i^4 = \sum_i \bar{w}_i f_i \quad (5.7)$$

5.4.1.2 Le mécanisme d'apprentissage.

Un des principaux avantages de l'ANFIS est la possibilité d'employer un algorithme d'apprentissage hybride pour l'identification de l'ensemble de paramètres $\{a_1, b_1, c_1, a_2, b_2, c_2, \dots, a_n, b_n, c_n\} \cup \{p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, \dots, p_{31}, p_{32}, p_{33}\}$ qui minimisent (5.9). Un estimateur du type Moindre Carrés est utilisé pour l'optimisation des paramètres de (5.6) et une méthode du type gradient conjugué pour les paramètres en (5.3). Cette technique permet une convergence rapide vers un minimum.

Moindres Carrés pour l'évaluation des paramètres de la partie conséquent.

En substituant (5.6) dans (5.7), on obtient:

$$\begin{aligned} Q_1^5 &= \sum_i \bar{w}_i f_i = \sum_i \bar{w}_i (p_{i1}x_1 + p_{i2}x_2 + p_{i3}) \\ &= \bar{w}_1 (p_{11}x_1 + p_{12}x_2 + p_{13}) + \bar{w}_2 (p_{21}x_1 + p_{22}x_2 + p_{23}) \\ &= p_{11}(\bar{w}_1x_1) + p_{12}(\bar{w}_1x_2) + p_{13}(\bar{w}_1) + \\ &\quad p_{21}(\bar{w}_2x_1) + p_{22}(\bar{w}_2x_2) + p_{23}(\bar{w}_2) \end{aligned} \quad (5.8)$$

L'équation (5.8) est un modèle linéaire auquel le problème des moindres carrés est appliqué. Les paramètres à estimer sont $\{p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{23}\}$ et les mesures sont $\{x_1, x_2\}$, les \bar{w}_i sont les sorties de la troisième couche.

Algorithmes du gradient conjugué pour l'évaluation des paramètres de la prémisse.

Définissons l'erreur pour le k-ième patron d'apprentissage de la même façon que dans les réseaux de rétro-propagation classiques:

$$E_k = \frac{1}{2} (y_k - O_{1k}^5)^2 \quad (5.9)$$

où y_k est la sortie désirée de l'ANFIS et O_{1k}^5 est la sortie réelle de l'ANFIS. Seuls les paramètres de la première couche $\theta = \{a_1, b_1, c_1, a_2, b_2, c_2\}$, sont recalculés avec cet algorithme. On définit le vecteur gradient de E comme :

$$g(\theta) = \nabla E(\theta) = \left[\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}, \frac{\partial E(\theta)}{\partial \theta_3}, \dots, \frac{\partial E(\theta)}{\partial \theta_n} \right]^T \quad (5.10)$$

Puisque l'ANFIS est un réseau sans rétro-alimentation, la dérivée partielle de l'erreur par rapport aux paramètres de la prémisse est calculée en (5.11):

$$\frac{\partial E(\theta)}{\partial \theta_i} = \frac{\partial E(\theta)}{\partial O_5} \frac{\partial O_5}{\partial O_4} \frac{\partial O_4}{\partial O_3} \frac{\partial O_3}{\partial O_2} \frac{\partial O_2}{\partial O_1} \frac{\partial O_1}{\partial \theta_i} \quad (5.11)$$

Une fois que le gradient a été obtenu, la direction conjuguée peut être calculée à partir du vecteur gradient (5.12). La direction conjuguée est préférée à la direction du gradient puisqu'elle donne une convergence plus rapide vers un minimum.

$$\begin{aligned} d_k &= -g_k + \beta_k d_{k-1} \\ \text{where} \\ \beta_k &= \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})} \end{aligned} \quad (5.12)$$

Où d est la direction conjugué, g est la direction du gradient et β est le terme de directionnalité.

5.4.1.3 Utilisation de l'ANFIS pour la prédiction.

En utilisant la structure de base de l'ANFIS, on propose ici une méthode de prédiction de séries temporelles. Les variables injectées comme entrées dans l'ANFIS sont les valeurs actuelles et précédentes de la série temporelle, dans cet cas, l'historique de la variable à pronostiquer. En continuant avec l'exemple, à la place des variables d'entrée x_1 et x_2 de la Figure 5-3, on utilise les valeurs présentes et passées de y : $y[k]$ et $y[k-1]$ et on définit la sortie $\hat{y}[k+1]$. Le modèle en conséquence est ainsi Auto-Regressif.

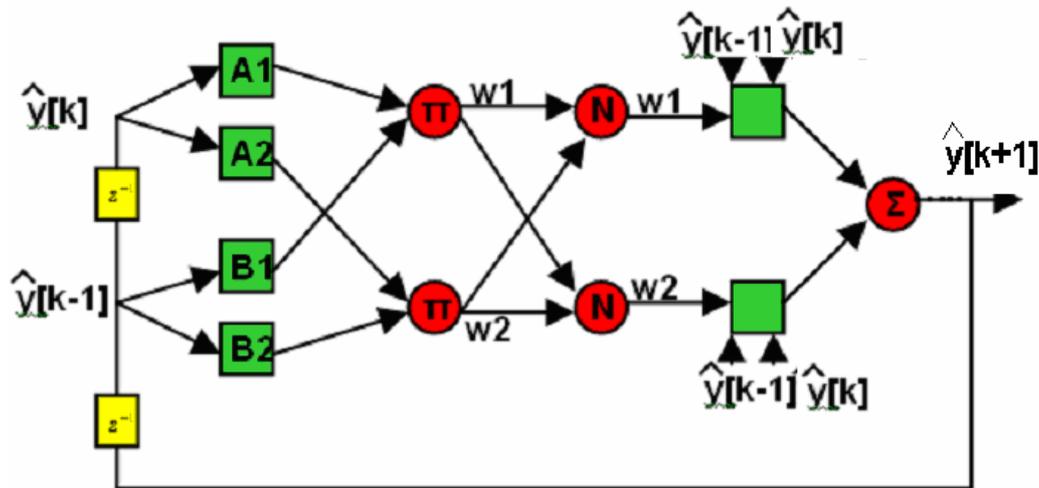


Figure 5-5. Prédiction multi pas.

L'ANFIS est entraîné avec les données disponibles pour faire une prédiction à un seul pas. Une fois que les paramètres ont été optimisés, l'ANFIS est employé de façon récurrente pour le problème de prévision multi-pas. Pour faire cela, la valeur estimée calculée par l'ANFIS est rétro-alimentée à l'entrée jusqu'à ce que l'horizon de prédiction désiré soit atteint, voir la Figure 5-5.

5.4.2 L'analyse de tendances pour la prédiction de séries de temporelles.

L'analyse de tendances est une approche très utile pour l'extraction d'information à partir de données numériques et pour les représenter symboliquement, d'une manière qualitative ou semi-qualitative. Son objectif est de convertir des données numériques obtenues en ligne vers un type de connaissance utilisable pour l'aide à l'opérateur. Le résultat de l'analyse de tendances est une description de l'évolution de l'état qualitatif d'une variable, dans un intervalle de temps, en utilisant un ensemble de symboles appelés les primitives.

De façon générale un petit ensemble de primitives est employé pour représenter la tendance d'un signal et pour faciliter son interprétation par un opérateur humain. L'exemple le plus simple des primitives est le triplet $\{stable, croissant, décroissant\}$. Ces primitives sont employés pour créer des épisodes : un épisode est défini par l'ensemble $\{primitive, temps\}$. Un épisode est ainsi un intervalle de comportement uniforme où toutes les propriétés qualitatives d'une variable sont constantes. Des primitives plus avancées se fondent sur la définition des formes particulières. Une tendance qualitative est par conséquent un historique qualitatif qui est représenté par un suite d'épisodes consécutifs (contigus et non chevauchants). La tendance peut être interprétée par l'opérateur en fonction du système spécifique, grâce à sa connaissance experte. L'extraction de tels épisodes à partir des données quantitatives permet de concentrer l'attention de l'opérateur sur des données sujettes à des modifications [Dash and Venkatasubramanian 2001; Charbonnier, Becq et al. 2002; Colomer, Meléndez et al. 2002]. Cette idée peut être élargie afin de fournir à l'opérateur une idée de comment la variable se comportera à l'avenir.

La méthode d'analyse de tendances proposée ici consiste en 4 étapes :

1. La segmentation en ligne des données.
2. La classification du dernier segment calculé dans une forme temporelle.
3. La transformation de la forme obtenue en modèle semi quantitatifs.
4. L'agrégation du modèle courant de tendance avec les précédents pour former une séquence d'épisodes successifs.

5.4.2.1 Segmentation des données en ligne

Ici, une représentation des données en segments linéaires successifs est employée. L'algorithme de segmentation en ligne détermine le moment où l'approximation actuelle n'est plus acceptable et où un nouveau segment devrait être calculé (Figure 5-6). Chaque segment est un modèle linéaire: $y(t) = p(t - t_0) + y_0$ où t_0 est le moment où le segment commence, p est sa pente et y_0 est la valeur au temps t_0 . L'utilisation de modèles linéaires comme approximation permet l'opérateur de connaître la tendance du signal. Les paramètres sont identifiés en utilisant le critère des moindres carrés.

La technique employée pour détecter si l'approximation est encore acceptable est la somme cumulative (*CUSUM*). Cette technique est fondé sur l'intégration de la différence entre la valeur observée et la sortie du modèle courant. La fonction *CUSUM* est très sensible aux changements de comportement dans les données.

$$\begin{aligned} cusum(t_1 + k\Delta t) &= cusum(t_1 + (k-1)\Delta t) + e(t_1 + k\Delta t) \\ &= \sum_{j=0}^k e(t_1 + j\Delta t) \end{aligned} \quad (5.13)$$

où $e(t_1 + k\Delta t) = y(t_1 + k\Delta t) - \hat{y}(t_1 + k\Delta t)$ est la différence entre la mesure et l'extrapolation.

La décision au temps $t_1 = k\Delta t$ est prise avec deux seuils prédéfinis, th_1 et th_2 ($th_1 < th_2$) (Figure 5-6) :

Si $|cusum(t_1 + k\Delta t)| \leq th_1$: le modèle courant est acceptable.

Si $|cusum(t_1 + k\Delta t)| > th_1$: $y(t_1 + k\Delta t)$ et t_1 sont mémorisés.

Si $|cusum(t_1 + k\Delta t)| \geq th_2$: le modèle n'est plus acceptable. Un nouveau modèle est calculé en utilisant les données mémorisées.

Une fois qu'un nouveau modèle a été calculé la valeur de *CUSUM* est remise à zéro.

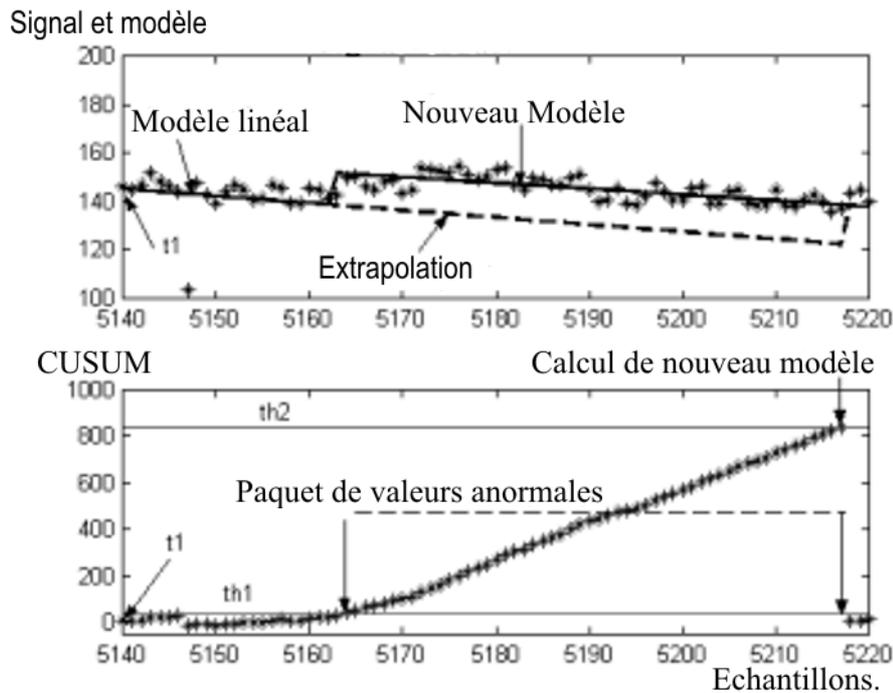


Figure 5-6. Segmentation de données.

5.4.2.2 Classification du nouveau segment dans une forme temporelle.

Le nouveau segment est agrégé avec le précédent pour former une forme composée. Cette forme est classifiée en 7 formes temporelles: « *croissant* », « *décroissant* », « *stable* », « *échelon positif* », « *échelon négatif* », « *transitoire concave* », « *transitoire convexe* » (Figure 5-7). Chacune des formes est décrite par trois caractéristiques :

I = la variation totale produite par la forme composée :

$$I^i = y_e^i - y_b^i \quad (5.14)$$

où les différentes ordonnées sont définies sur la Figure 5-7

I_d = la variation due à une discontinuité entre deux segments:

$$I_d^i = y_o^i - y_b^i \quad (5.15)$$

I_s = la variation due à la pente du nouveau segment :

$$I_s^i = y_e^i - y_o^i \quad (5.16)$$

Une fois que les différentes caractéristiques de la forme temporelle ont été calculées, une classification est effectuée en utilisant deux seuils th_c et th_s prédéfinis et les règles suivantes :

Si $|I_d| > th_c$ alors la forme est « *discontinue* » (c'est un « *échelon* », un « *échelon+pente* » ou un « *transitoire* »).

Si la forme est « *discontinue* » et $|I_s| < t_{hs}$ alors la forme est un « *échelon* », positif ou négatif selon le signe de I_d .

Sinon, si $sign(I_d) \neq sign(I_s)$ c'est un « *transitoire* » sinon c'est un « *échelon+pente* ».

Sinon, si la forme est « *continue* » (elle est « *stable* », « *croissante* » ou « *décroissante* »).

Si la forme est « *continue* » et $|I| < t_{hs}$, alors la forme est « *stable* ».

Sinon, elle « *croissante* » ou « *décroissante* », selon le signe de I .

La classification du segment en une forme temporelle fournit des informations symboliques à l'opérateur.

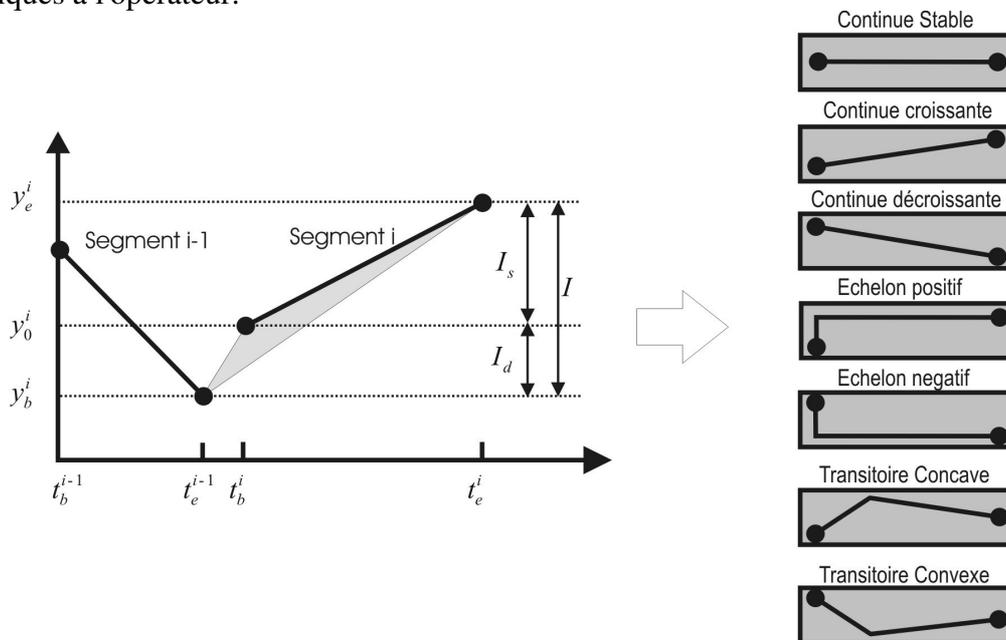


Figure 5-7. Classification en formes temporelles.

5.4.2.3 Transformation des formes en une séquence d'épisodes semi-quantitatifs.

L'information quantitative produite par la segmentation des données peut être stockée et associée à l'information qualitative afin de produire une information semi-quantitative. De cette façon, les 7 formes temporelles peuvent être décrites par 3 modèles de tendance semi-quantitative: « *stable* », « *croissant* », « *décroissant* ». Trois paramètres quantitatifs sont employés pour la nouvelle description: l'instant de début de la forme t_b^i , la valeur de la variable au début de forme y_b^i et la valeur de la variable à la fin de la forme y_e^i . Par exemple:

Un *échelon positif +pente* devient:

$$\{ \text{Croissant}, t_b^i, y_b^i, y_o^i \} + \{ \text{Croissant}, t_b^i + T_s, y_o^i, y_e^i \}.$$

Un *échelon positif* devient:

$$\{ \text{Croissant}, t_b^i, y_b^i, y_o^i \} + \{ \text{Stable}, t_b^i + T_s, y_o^i, y_e^i \}$$

ou T_s est la période d'échantillonnage.

5.4.2.4 Agrégation des modèles de tendance.

L'agrégation des modèles de tendance semi-quantitatives comprend l'association du modèle de tendance courant avec le précédent afin d'établir les épisodes temporels les plus longs possibles. Les agrégations possibles entre 2 épisodes sont:

Croissant + Croissant = Croissant.

Décroissant + Décroissant = Décroissant.

Stable + Stable = Stable si l'augmentation globale de la séquence $|y_e^i - y_b^{i-1}| \leq t_{hs}$ (t_{hs} est un seuil employé pour séparer les formes "stable" et "croissant/ décroissant").

Stable + Stable = Croissante (ou *Décroissante*) si l'augmentation globale de la séquence $|y_e^i - y_b^{i-1}| > t_{hs}$. Ceci permet la détection des dérives lentes dans le signal. La dérive prendra plus de temps pour être détectée puisqu'elle exige l'association d'au moins deux modèles de tendance stables.

L'algorithme d'agrégation suit la logique suivante:

Si l'épisode précédent est $\{ \text{Croissant}, t_b^{i-1}, y_b^{i-1}, y_e^{i-1} \}$ **et** le modèle courant de tendance est $\{ \text{Croissant}, t_b^i, y_b^i, y_e^i \}$

Alors : le nouvel épisode est $\{ \text{Croissant}, t_b^{i-1}, y_b^{i-1}, y_e^i \}$.

Si l'épisode précédent est $\{ \text{Croissant}, t_b^{i-1}, y_b^{i-1}, y_e^{i-1} \}$ **et** le modèle courant de tendance est $\{ \text{Stable}, t_b^i, y_b^i, y_e^i \}$ ou $\{ \text{Décroissante}, t_b^i, y_b^i, y_e^i \}$

Alors : il ne peut pas être agrégé et commence un nouvel épisode: $\{ \text{Croissant}, t_b^{i-1}, y_b^{i-1}, y_e^{i-1} \}$; $\{ \text{Stable}, t_b^i, y_b^i, y_e^i \}$.

5.4.2.5 L'emploi de l'analyse de tendance pour la prédiction.

Les résultats partiels de l'analyse de tendance permettent d'avoir différentes représentations du signal. D'abord, la segmentation de données représente le comportement du

signal à court terme. L'agrégation d'épisodes permet de passer à une représentation à mi-terme ou long terme.

Au temps t_c , le temps courant, quand une nouvelle approximation linéaire a été juste calculée par l'algorithme de segmentation, le segment précédent est complètement défini ainsi par :

$$\text{Segment}_{i-1}: \{ t_0(i-1), y_0(i-1), t_c(i-1), y_c(i-1) \}$$

Les étapes de classification, transformation en épisodes semi-quantitatifs, et agrégation sont réalisées d'abord en employant le segment précédent (i-1). La tendance définitive est alors extraite jusqu'à $t_c(i-1)$.

Pour extraire la tendance jusqu'au temps courant, t_c , le segment courant est défini provisoire par :

$$\text{Current Segment}: \{ t_0(i), y_0(i), t_c, y_c(i) \}$$

où:

$$y_c(i) = p(i) \cdot [t_c - t_0(i)] + y_0(i).$$

Les mêmes étapes de classification, transformation en épisodes semi-quantitatifs, et agrégation sont alors réalisées en utilisant le segment courant (i) pour extraire un épisode transitoire. Plus tard, au moment où une nouvelle approximation linéaire est calculée, le segment courant devient un segment sûr, il est complètement défini. Ce segment est agrégé à la séquence d'épisodes définitive et un nouvel épisode transitoire est calculé en utilisant le nouveau segment courant.

Les résultats de l'analyse de tendance peuvent être employés de différentes manières pour informer l'opérateur sur le comportement d'un signal donné. L'information qualitative telle que la variable est "*Stable*" ou "*Croissant/Décroissant*" est par elle-même un outil intéressant pour aider l'opérateur dans sa prise de décision. De plus, l'information semi-quantitative de tendance peut être employée dans la prédiction des valeurs de la variable surveillée dans une fenêtre de temps. L'algorithme de segmentation fournit le modèle linéaire qui décrit au mieux le comportement de la variable. Si un seuil d'alarme a été choisi, le modèle $y(t) = p(t - t_0) + y_0$ est employé pour résoudre le problème : combien de temps est-il nécessaire pour que le signal $y(t)$ arrive à ce seuil?.

5.4.3 La prédiction de la propagation de défaut à partir des résultats de diagnostic.

En cas de défaut, un outil pour la prise de décision est la prédiction de l'évolution de l'état du procédé. Néanmoins, cette prédiction peut être tout à fait incertaine parce que le défaut a modifié le comportement normal du système. Dans le cas où un graphe causal du système est disponible et ce graphe inclut le modèle de comportement du composant défectueux, le module de pronostic fournit à l'opérateur l'information en ce qui concerne l'ensemble des variables du système qui seront possiblement affectées par le défaut (si aucune action n'est entreprise). Sous cette forme, les conséquences de défaut peuvent être analysées

par rapport aux variables critiques possiblement affectées. C'est un genre de prédiction qualitative.

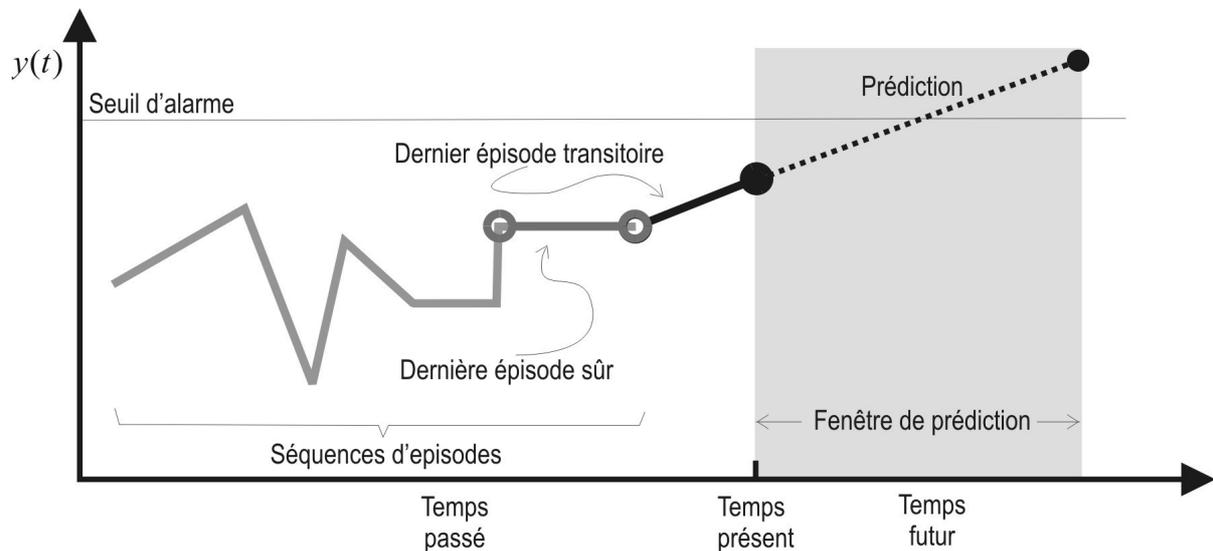


Figure 5-8. Utilisation de l'analyse de tendances pour la prédiction.

L'analyse de la propagation des effets de défauts présentée est une analyse du type pessimiste, puisqu'elle donne la totalité des variables liées causalement au composant défaillant. Cette analyse suppose en particulier que la structure du système n'est pas modifiée par le défaut. Si ce n'était pas le cas, il faudrait disposer de modèles de défauts.

Le but principal de cette analyse est de trouver les variables liées "causalement" avec celles qui expriment le modèle du composant défectueux. L'algorithme utilisé est une recherche en avant des nœuds dans le graphe qui gardent une relation avec soit les arcs, soit les nœuds liés au composant défectueux. Cette information permet principalement de visualiser ces variables critiques pour lesquelles il existe le risque de déviation par rapport au fonctionnement normal.

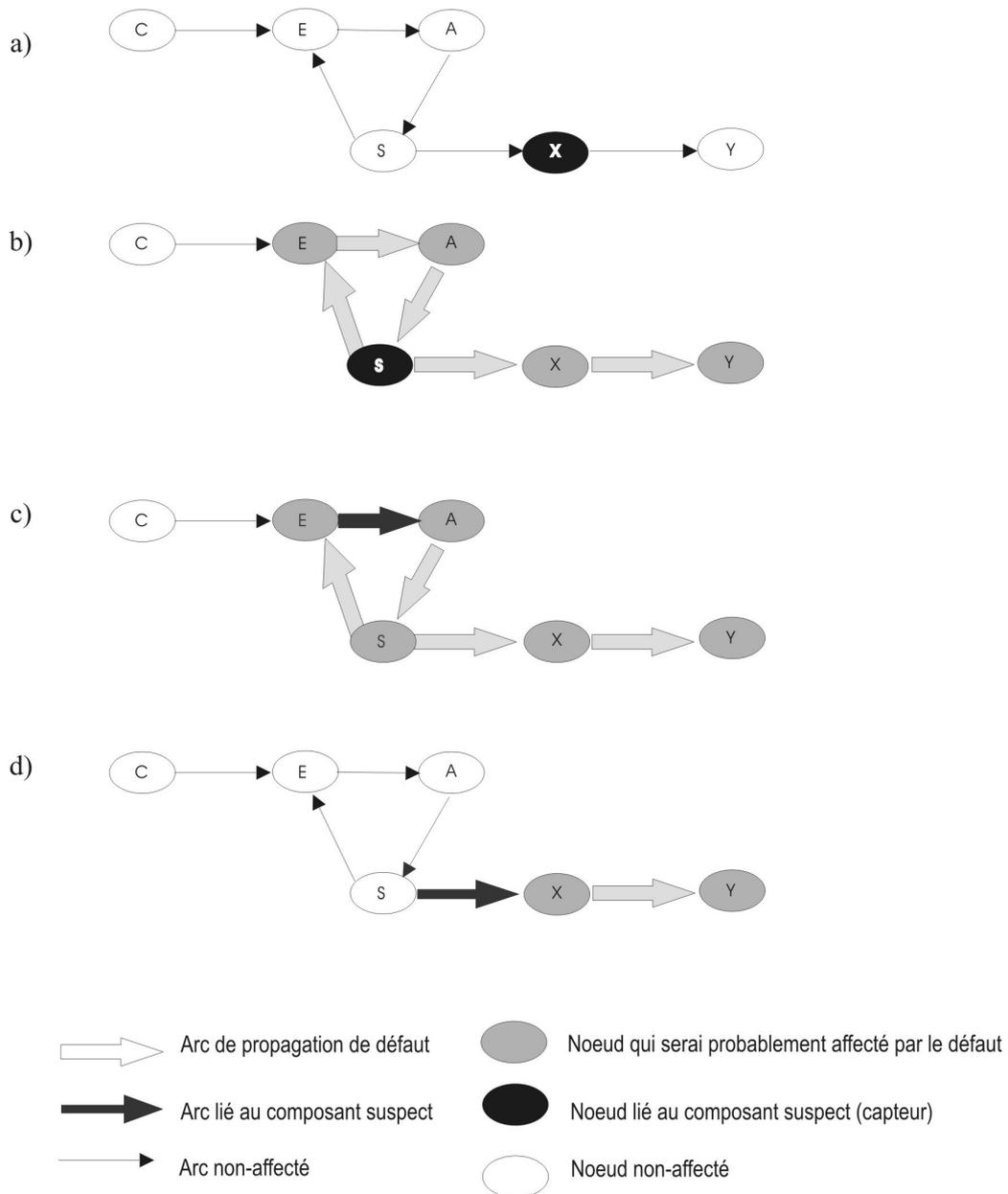


Figure 5-9. Possibilités de la propagation des effets du défaut ; a) défaut de capteur de la variable X hors d'une boucle de régulation (le défaut n'est pas propagé par le procédé) ; b) défaut de capteur de la variable S dans une boucle de régulation (toutes les variables de la boucle sont affectées) c) défaut d'actionneur dans une boucle de régulation (en principe le défaut ne se propage pas en admettant que la régulation fonctionne normalement, toutefois, en cas de blocage ou saturation, le défaut peut se propager dans le boucle) d) défaut du système hors de la boucle de régulation (le défaut est propagé par le procédé).

L'obtention des variables plausiblement affectées est une recherche en avant dans le graphe. L'algorithme prend en considération le fait que les boucles de régulation représentent des cycles particuliers dans le graphe, ce pourquoi l'effet d'un composant en défaut en aval du graphe peut se répercuter sur les variables amont dans le graphe (celles qui appartiennent au cycle).

Algorithme :

I. À partir d'un composant suspect, déterminer quels nœuds ou arcs lui correspondent.

II. Analyser les différentes possibilités :

a) Le composant est un capteur : étant donné que les capteurs sont toujours associés aux nœuds, il y a deux possibilités :

Le capteur n'est pas utilisé dans une boucle de régulation : la défaillance du capteur n'a pas d'effet sur les autres variables (la variable réelle n'a pas une valeur anormale), Figure 5-9-a .

Le capteur est utilisé pour la rétro-alimentation dans une boucle de régulation : les effets de la défaillance du capteur sont propagés dans la boucle et sur les nœuds liées en aval à la boucle, Figure 5-9 b.

b) Le composant est un élément du système ou un actionneur : les nœuds "causalement" liés directement en aval du composant peuvent être affectés, et récursivement tous les nœuds descendants, Figure 5-9 c et d.

5.5 Outils pour l'aide à la conduite face aux défauts.

Un troisième outil pour l'aide à la décision est le conseil d'action. Cette tâche doit être préparée par une étude soignée a priori de la structure du procédé, éventuellement dans plusieurs configurations. Cette étude dépend essentiellement de la connaissance experte disponible sur les diverses possibilités de reprise. Associé à cette analyse, il est également nécessaire d'identifier toutes les variables qui peuvent être directement modifiées par l'opérateur. Des exemples de ce genre de variables sont les consignes par l'intermédiaire du système de conduite et les ouvertures des vannes en configuration de boucle ouverte, ces variables sont définis comme *Moyens d'action*. Toute cette connaissance est acquise hors-ligne par le **PSA** dans la phase de configuration/customisation.

Le **DSA** se sert de la connaissance obtenue pendant la configuration/personnalisation, telle que les graphes causaux, la définition des variables *Moyens d'action* et les procédures de reprise prédéfinies :

Prescription hors ligne des procédures de reprise. Cette façon de faire est très classique dans l'industrie, dans ce cas, l'expérience de conduite est employée comme connaissance pour la surveillance dans une situation anormale. Dans la phase de personnalisation, il est nécessaire de déclarer un ensemble de procédures de repris prédéfinis, liées à chaque défaut possible ou à chaque composant et de les stocker dans la base de données. Un Conseil d'Action Prédéfini est une séquence spécifique de commandes associés aux composants et à leurs modes de défaillance,

devant être exécutées par l'opérateur. Cette connaissance est stockée dans la base de données.

La recherche de *Moyens d'action* hors ligne avec un graphe causal. Si un graphe causal est disponible pour un sous-système (ou un sous-système dans diverses configurations), certaines de ses variables sont prédéfinies comme *moyens d'action* pendant la phase de customisation. Le DSA peut attacher à chaque variable du graphe une liste des différents *moyens d'action* qui agissent sur cette variable dans les différentes configurations. Cette information est obtenue automatiquement, en faisant une recherche dans le graphe. La principale information utilisée dans cette étape est l'organisation causale des variables et la redondance physique dans l'installation représentée par les différentes configurations. Cette façon de recherche de procédures de reprise est par contre originale et ne nécessite pas de connaissances supplémentaires par rapport à celles qui sont requises pour le diagnostic.

Conseils d'action en ligne Quand un défaut a été détecté, la liste de moyens d'action appropriés est présentée à l'opérateur. "Approprié" dénote les moyens d'action qui ont une influence sur les variables affectées par le comportement anormal du composant. Ces conseils d'action correspondent soit *aux moyens prédéfinis d'action*, précédemment stockés dans la base de données, soit *aux moyens d'action* déterminés en ligne en utilisant le graphe causal. Ainsi l'opérateur peut savoir quelles variables d'action ont de l'influence sur les variables qui présentent des valeurs anormales. Ainsi l'opérateur peut maintenir la production dans diverses configurations. Le DSA peut fournir une liste de reconfigurations possibles et des moyens d'action appropriés liés à chaque configuration. Ceci implique l'utilisation d'autres chemins pour agir sur la même variable en évitant l'utilisation des composants défectueux. Dans les prochains paragraphes, on se focalisera sur l'utilisation hors-ligne et en ligne du graphe causal pour trouver les Conseils d'Action puisque c'est cette approche qui est originale.

5.5.1 Les moyens d'action.

Le modèle causal possède toute l'information nécessaire pour la recherche des conseils d'action. Une liste de moyens d'action peut être liée à chaque noeud. En fait, un moyen d'action est une structure liée à chaque variable qui contient l'information suivante :

- la référence au nœud associé à la variable d'action
- le temps de réaction : ce paramètre donne le temps requis par l'action pour avoir un effet sur la variable. C'est la somme du temps de réponse et de retards des diverses fonctions de transfert portées par les arcs sur le chemin entre le moyen d'action et la variable.
- Le sens du moyen d'action : cette information qualitative montre simplement si l'action et la variable évoluent dans le même sens ou dans la direction opposée (l'augmentation de l'action impliquant l'augmentation ou la diminution de la valeur de la variable).

Un moyen d'action est attaché à la variable étudiée et pas à l'action elle-même. De cette façon, s'il existe différents chemins entre un nœud d'action et n'importe quel autre nœud, cela aura comme conséquence différents moyens d'action pour ce nœud.

La première phase pour l'obtention des moyens d'action est une recherche en arrière dans le graphe. L'algorithme considère d'abord les nœuds qui n'appartiennent pas à un circuit. Un nœud est traité d'une façon récursive jusqu'à ce que tous ses nœuds ancêtres aient été traités et que l'algorithme arrive aux nœuds sans nœuds ancêtres. Quand un nœud a été choisi, il existe différentes possibilités :

- le nœud a des nœuds parents : dans ce cas-là, il hérite les moyens d'action de ses parents, modifiés par l'information dans l'arc qui le relie à eux ;
- le nœud est un nœud d'action ou de type consigne : dans ce cas-là, il est son propre moyen d'action ;
- le nœud n'est ni une action ni une consigne et n'a pas des nœuds parents. La variable associée est une perturbation pour le procédé, ainsi il n'y a aucun moyen d'action sur cette variable.

La recherche dans un circuit est basée sur ces hypothèses :

- i) Un nœud d'action ne peut pas appartenir à un circuit.
- ii) Un nœud peut apparaître une seule fois sur un chemin qui relie un nœud d'action et n'importe quel autre nœud.

Le traitement est facilement décrit avec l'exemple montré sur la Figure 5-10. En prenant par exemple le nœud N3, la première étape consiste en une vérification sur l'appartenance du nœud à un circuit au moyen d'une analyse récursive d'ancêtres jusqu'à l'arrivée aux nœuds sans parents ; si le nœud (N3) est trouvé dans cette analyse, alors le nœud appartient à un circuit. Une fois détectée l'appartenance au cycle, les arcs sortants du nœud en question sont ouverts pour casser le cycle (arc a3). À cette étape, l'algorithme essaie d'hériter les moyens d'action des parents du nœud analysé ; s'ils ne sont pas encore disponibles, il cherchera les moyens d'action des parents des parents, et ainsi récursivement jusqu'à l'arrivée aux nœuds sans parents. On trouvera A2 comme Moyen d'action directe de N2, et A1 comme Moyen d'Action «hérité» de N1. A2 et A1 sont passés comme Moyens d'Action au nœud N3. Au moment de refermer le cycle, les Moyens d'Action de N3 sont passés à ses descendants, dans ce cas, N1 hérite A2 à N3. Dans cet exemple, les moyens de l'action sont :

N1 : A1 par le chemin direct.
A2 par le chemin A2-N2-N3-N1

N2 : A2 par le chemin direct.
A1 par le chemin A1-N1-N2

N3 : A1 par le chemin A1-N1-N2-N3
A2 par le chemin A2-N2-N3.

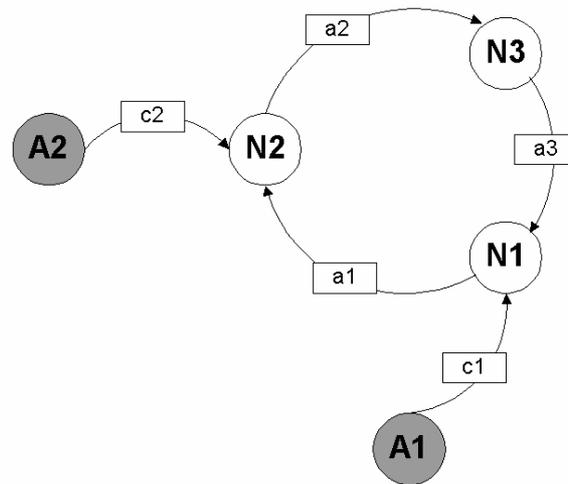


Figure 5-10. Exemple d'un graphe avec un cycle.

Après le traitement des nœuds qui appartiennent aux différents circuits, une "propagation" des moyens d'action est exécutée sur les nœuds enfant qui n'appartiennent pas au circuit.

5.5.2 Recherche en ligne de conseils d'action.

Le traitement en ligne est lancé par le système quand le composant en défaut a été localisé. Le DDA est censé d'avoir informé l'opérateur au sujet de la situation générale du procédé (les composants défaillants). Le système d'aide à la décision donnera à l'opérateur les actions qu'il peut prendre afin de résoudre ou de réduire le problème. Le traitement hors-ligne fournit une liste de moyens d'action potentiels dans diverses configurations agissant sur les variables perturbées.

Si le DDA a localisé un seul composant défaillant, le système de conseil d'action recherche les moyens d'action sur les variables affectées par ce composant en défaut. Dans le graphe causal un lien entre les composants et les arcs et/ou les nœuds est établi. Avec ceci, l'algorithme trouve les nœuds directement associés au composant défaillant et cherche des moyens d'action sur les nœuds enfants les plus proches. Tous les moyens d'action qui utilisent comme chemin les arcs liés au composant défaillant sont supprimés. Si la source est un moyen d'action, elle doit être aussi supprimée de la liste de moyens d'action. Si divers moyens d'action sont trouvés, l'opérateur peut choisir celui qui a le temps de réaction le plus petit. Si un moyen d'action est employé par deux variables différentes et si les directions sont incompatibles, le résultat du raisonnement est traduit comme contradiction (par exemple : « pour corriger la déviation de la variable N1, réduire l'action A1, et pour corriger la correction de la variable N2, augmenter l'action A1 »). Afin de traiter les contradictions, la première possibilité est d'essayer de les éliminer. De cette façon, dans le cas où tous les nœuds affectés ont un ou plusieurs moyens d'action il suffit d'employer différents moyens d'action. Quand on ne peut pas éliminer toutes les contradictions, un algorithme cherche l'existence d'un compromis. La Figure 5-11a) décrit la situation de contradiction la plus simple, aucun

compromis n'est possible et il y aura un défaut. Dans le deuxième cas, Figure 5-11b), le conseil d'action basé sur le noeud A1 crée une contradiction qui ne peut pas être éliminée. Pour résoudre cette situation, la seule manière d'agir sur N1 est A1. Quand il y a besoin d'action sur N2, les deux possibilités de moyennes d'action sont A1 et A2. Un compromis possible est alors d'agir sur A1 pour résoudre N1 et d'agir sur A2 pour résoudre N2, et de compenser, en même temps, l'effet de A1 sur N2.

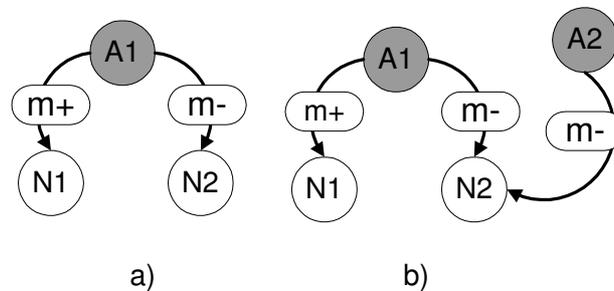


Figure 5-11. a) Contradiction irréductible, b) contradiction réductible.

Les critères de sélection du noeud à utiliser sont : "Parmi tous les nœuds affectés, le premier nœud traité sera celui qui a le plus petit nombre de moyens d'action et celui avec le plus grande nombre de contradictions des conseil d'action créés". Le nœud résultant aura le plus petit nombre de degrés de liberté. Habituellement, la solution du conseil d'action contradictoire produira une suppression de quelques degrés de liberté dans les autres noeuds affectés. Utilisons encore l'exemple Figure 5-11,b), on peut observer qu'agir sur A1 pour résoudre N1 rend impossible d'agir sur A1 pour résoudre N2. Le processus de raisonnement emploie les étapes suivantes afin d'obtenir le meilleur conseil d'action avec un minimum de compromis :

- 1) Faire un choix prioritaire de tous les conseils d'action sans conflits de contradiction.
- 2) Choisir le conseil d'action avec le plus de conséquences dans la bonne direction.
- 3) Choisir le conseil d'action avec le moins de conséquences dans la mauvaise direction.

5.6 Outils pour l'aide à la prise de décision dans les politiques de maintenance

Le conseil de maintenance est un outil important de décision, consistant à conseiller de façon réactive, après qu'un composant défectueux ait été localisé, en donnant à l'opérateur les conseils pour réparer, vérifier, remplacer etc., le composant suspecté. Le conseil de maintenance pour les différents composants doit être prédéfini au moment de la configuration et stocké dans la base de données.

En général, les réparations ne peuvent pas être accomplies pendant la production. Prenant l'exemple de l'industrie métallurgique, seules de petites réparations peuvent être faites entre deux séquences de production, et les réparations plus importantes doivent être

accomplies seulement pendant l'arrêt périodique de maintenance. Ainsi le temps auquel l'entretien devrait être fait est un paramètre important pour le conseil de maintenance.

L'outil proposé ici utilise les Conseils de Maintenance stockés auparavant dans la base de données. Au moment où un message de diagnostic arrive au DSA (liste de composants suspects), il effectue une recherche des conseils de maintenance associés aux composants suspectés. Ces conseils sont montrés à l'opérateur au moyen de l'interface OIA.

Grâce aux résultats de pronostic, l'opérateur ou l'ingénieur de maintenance peuvent envisager un politique de maintenance vis-à-vis des effets des défaillances.

On notera que la maintenance est très dépendante de la nature et de la qualité des composants utilisés. En conséquence, elle doit être adapté à chaque procédé, et peut difficilement faire l'objet d'algorithmes généraux.

5.7 Configuration de l'agent d'aide à la décision.

Les éléments nécessaires au DSA sont configurés par le PSA. Cela inclut l'acquisition de l'information suivante :

Graphes causaux : le DSA utilise les graphes causaux créés par l'éditeur de graphes (chapitre 4). Afin de trouver la totalité des moyens d'action sur les variables, le DSA doit connaître tous les graphes associés aux configurations possibles.

Conseils d'action prédéfinis : dans le cas où il existerait des actions prédéfinies associés à un composant particulier.

Conseils de maintenance : ceci se réfère à l'introduction de conseils de maintenance associés à un composant.

Paramètres pour les algorithmes de propagation : l'algorithme de propagation fondé sur l'analyse de tendances requiert de fixer au préalable certains paramètres. La sélection des paramètres nécessaires pour l'analyse de tendances est une procédure lente qui requiert l'expérience de l'opérateur.

5.8 Exemple académique.

On continue ici avec l'exemple de deux bacs montré dans le chapitre antérieur, et on continue avec les mêmes scénarios.

5.8.1 Scénario 1 : Fuite progressive du Bac 1 (Figure 5-12)

La valeur prévue de chacune des variables est représentée du côté droit de chaque courbe (fond gris). Cette prévision est basée sur le dernier épisode transitoire calculé, qui est extrapolé sur un long horizon de temps. La prédiction permet de voir que le niveau h1 restera

à son niveau de référence longtemps grâce à la boucle de régulation ; mais ce ne sera plus le cas quand l'ouverture de la vanne V_{k1} atteindra son niveau de saturation (100 %). L'opérateur a une idée claire au sujet du temps avant saturation et peut effectuer ainsi une action; l'opérateur a assez de temps pour envisager, par exemple, une réduction de la valeur de la consigne du niveau h_2 , ou un arrêt programmé, ou un appel au service d'entretien.

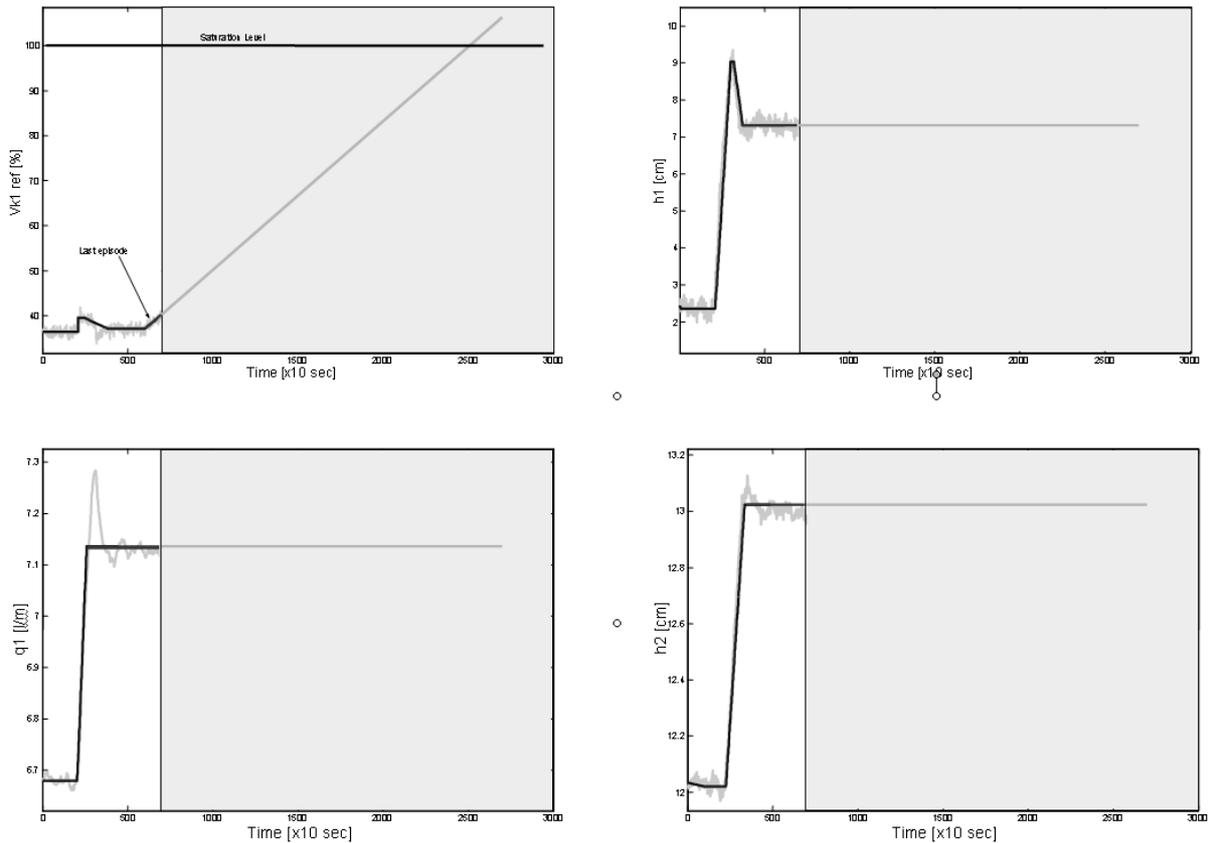


Figure 5-12. Prédiction des variables après une fuite progressive du Bac 1.

5.8.2 Scénario 2 : Fuite brusque du Bac 1 (Figure 5-13).

Puisque le défaut n'est pas très grand, le système de commande a compensé la fuite en augmentant le débit d'entrée du Bac 1. L'opérateur peut voir que cette situation pourrait être maintenue jusqu'à la période suivante d'entretien ou anticiper la politique d'entretien.

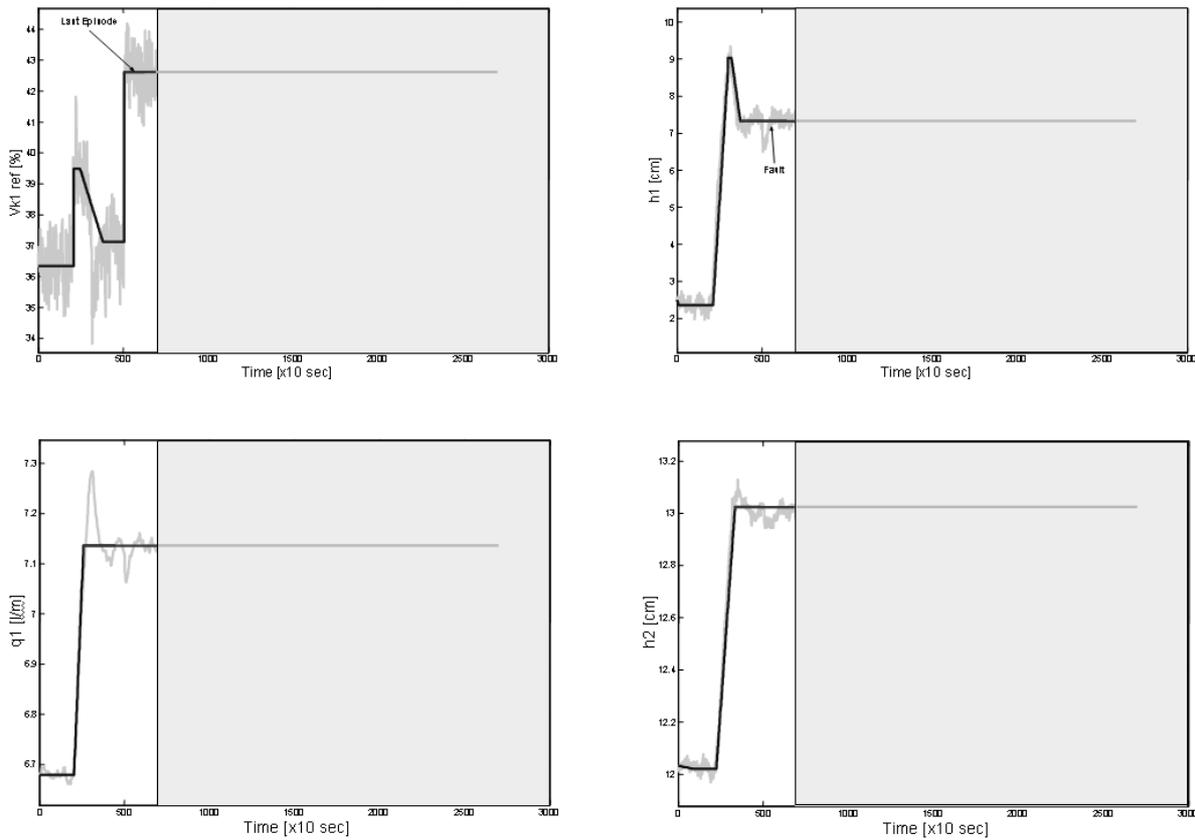


Figure 5-13. Prédiction des variables après une fuite brusque du Bac 1.

5.8.3 Scénario 3 : Fuite progressive du Bac 2 (Figure 5-14).

Dans ce cas-ci, l'opérateur est au courant de ce fait et peut voir que $h1$ atteindra sa limite de sûreté (50 centimètres) avant la saturation de la vanne $Vk1$. Dans ce cas-ci, l'opérateur dispose d'un temps très petit avant la saturation de niveau ; ainsi l'opérateur doit rapidement prévoir une reconfiguration du processus en utilisant le Bac 3 de sécurité, fermant la vanne $EV2$, et ouvrant la vanne $EV1$.

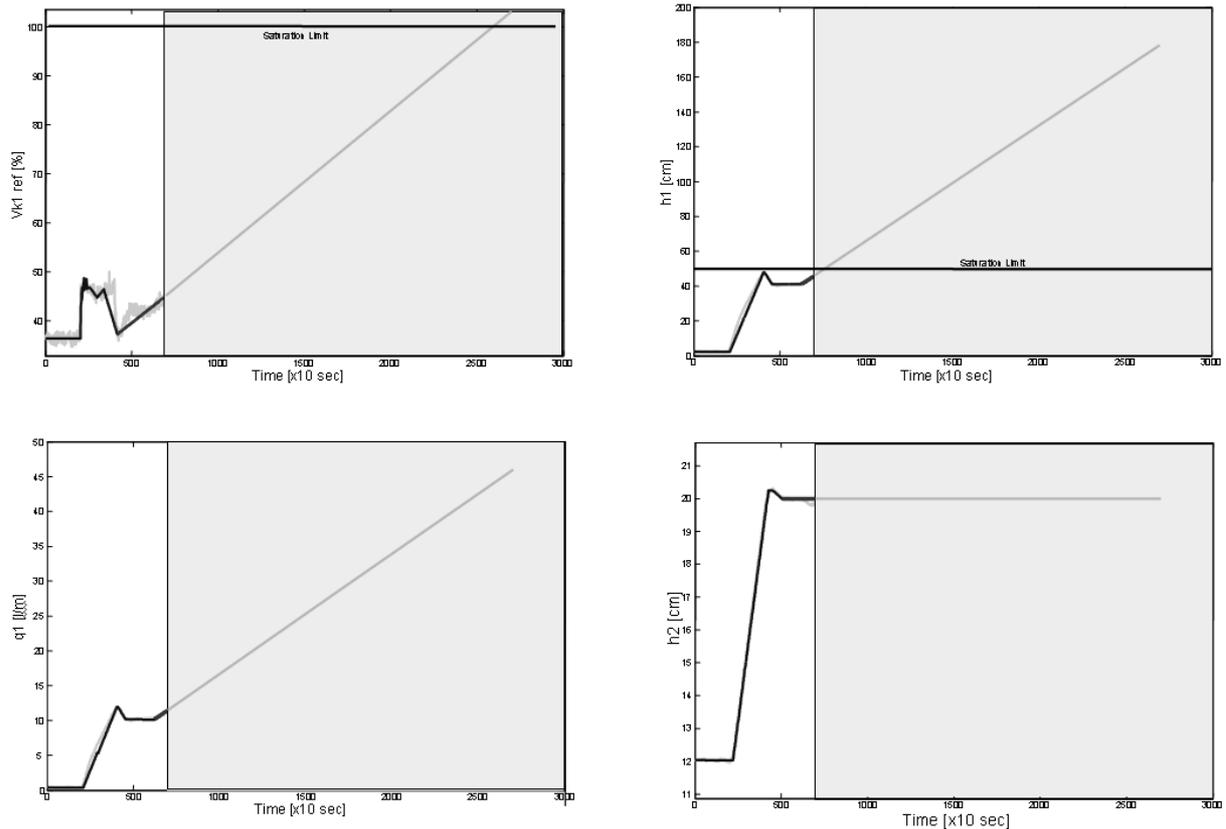


Figure 5-14. Prédiction des variables après une fuite progressive du Bac 2.

5.9 Conclusions.

Afin d'aider l'opérateur dans la prise de décisions, le DSA intègre trois outils d'aide à décision, qui sont utilisés dans les phases *d'anticipation de l'évolution du système*, *d'interprétation des conséquences* et *de définition de tâches* selon le modèle de Hoc [Hoc 1993].

Dans ce chapitre, deux méthodes différentes pour le pronostic de l'évolution des variables ont été présentées. La première est basée sur un réseau de neurones. La prédiction récursive permet la prédiction multi-pas. L'avantage principal de cette méthode est sa capacité d'auto-apprentissage. Cette caractéristique est utile quand aucune connaissance précédente sur le signal n'est disponible. Son inconvénient principal est le temps de calcul nécessaire et la variété de scénarios nécessaires pour effectuer cet apprentissage.

La deuxième méthode présentée est basée sur l'analyse de tendance. La complexité informatique modérée de cette méthode permet son utilisation en ligne. Cependant dans cette méthode les différents seuils qui sont nécessaires pour la segmentation de l'information et la classification doivent être choisis par l'ingénieur de procédé. On peut envisager d'employer les deux méthodes dans des systèmes d'aide à la supervision pour la prévision des variables critiques, en fonction de la précision exigée, des taux de variation des signaux et de la connaissance initiale sur le signal.

Les Conseils d'Action produits par le DSA ont pour but de maintenir les variables non directement impliquées dans le composant en défaut à un niveau proche de leur référence ou à un niveau sûr, en faisant l'usage de la redondance physique du système (dans les cas de reconfiguration) ou de la redondance analytique dans le cas de la compensation des effets de défauts. Le problème de supervision et de reconfiguration est analysé du point de vue des systèmes à grande échelle sous la responsabilité des opérateurs humains. Comme les modèles analytiques ne peuvent pas être envisagés pour représenter l'installation avec des centaines de variables et plusieurs modes de défaut imprévisibles, on propose ici une analyse fondée sur un modèle qualitatif simple. Le modèle proposé est un graphe orienté. Ses noeuds représentent les variables de processus, parmi lesquelles des actions peuvent être trouvées. Ses arcs représentent la causalité entre les variables. Dans le cas le plus simple, quand aucune information quantitative n'est disponible, le graphe causal est un modèle purement qualitatif. Plusieurs graphes peuvent être employées pour représenter diverses configurations.

Enfin, la priorité des Conseils d'Action et de Maintenance produits par le DSA est déterminée par l'opérateur grâce aux résultats de pronostic et à son expertise.

Chapitre 6

Tests sur le tendeur hydraulique de bande

6.1 Introduction

Pendant des décennies, l'industrie sidérurgique a été un symbole de puissance économique. Elle fournit l'une des matières premières pour beaucoup d'autres industries, y compris la production des automobiles, avions, construction, machines-outils, etc. La compétitivité globale exige que les fabricants répondent avec des produits de haute qualité et bas coût et des livraisons du produit à temps. Par conséquent, les aciéries ont mis l'accent sur l'utilisation de technologies avancées en planification et programmation de la production du produit et sur l'automatisation du procédé de production. Pour augmenter leur capacité concurrentielle, les sociétés sidérurgiques ont consacré une partie de leur capacité de recherche au développement de systèmes de fabrication intégrés par ordinateur qui doivent pouvoir s'adapter à la nature dynamique des conditions de production. Un de ces systèmes est le système de supervision qui intègre la capacité de diagnostic de défauts, principalement en vue de la maintenance conditionnelle comme moyen d'atteindre les critères de disponibilité exigés.

Dans ce chapitre, l'utilisation de la méthode de détection de défauts fondée sur le graphe causal, expliquée au chapitre 4, est mise en application pour la détection de défauts

d'un tendeur hydraulique⁸ de bande, un des composants utilisé dans le procédé de laminage au chaud. Ce composant est en contact direct avec le métal chauffé au rouge, nécessaire pour maintenir la bande de métaux à une tension constante.

6.2 Le principe du 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 avec 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 la ductilité nécessaire pour produire les premières grandes déformations. Après que le métal ait été traité dans le stand de dégrossissage à l'épaisseur 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 and Nitta 2000; Montmitonnet P. 2002]. Le traitement entier est montré sur la Figure 6-1.

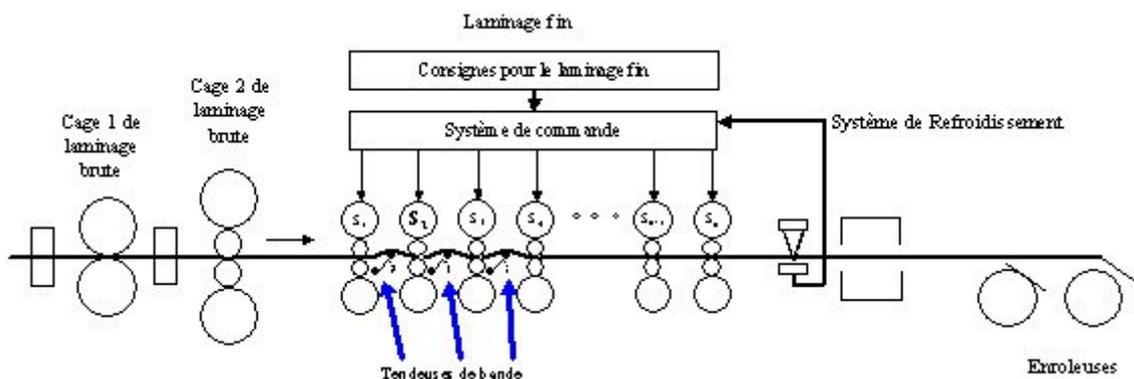


Figure 6-1. Procédé de laminage au chaud.

6.3 Description du tendeur hydraulique de bande et son rôle dans le laminoir.

Les tendeurs de bande sont situés typiquement entre deux cages 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. Le tendeur hydraulique de bande se compose des éléments suivants :

- C1 : La servo-vanne.
- C2 : Le tuyau hydraulique du côté de la tige.

⁸ 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.

- C3 : Le tuyau hydraulique du côté du piston.
- C4 : Le cylindre hydraulique.
- C5 : Le capteur de pression du côté de la tige.
- C6 : Le capteur de pression du côté du piston.
- C7 : L'articulation mécanique.
- C8 : Le capteur de position angulaire.

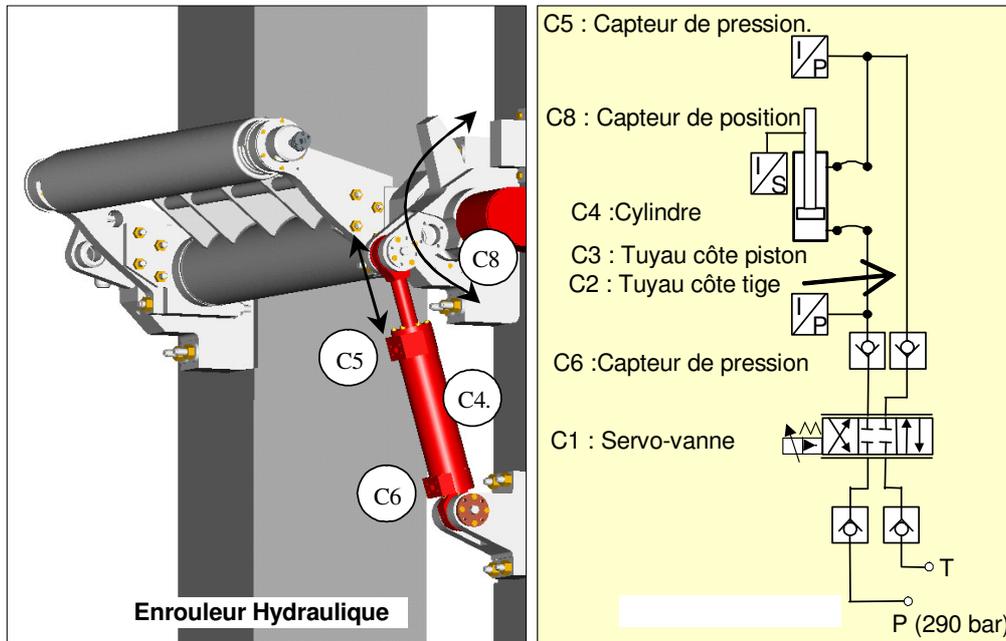


Figure 6-2: Schéma mécanique et hydraulique du tendeur hydraulique de bande.

6.4 Les différents tests d'acquisition effectués sur le tendeur hydraulique de bande en vue du diagnostic.

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 différents composants du procédé de laminage [Deckers 2003]. Les tests effectués par le PPMS sur la tendeuse 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 période fixe. Quatre types d'essai sont réalisés :

- **Test de position maximale** : Dans une configuration de commande de position en boucle fermée, le tendeur hydraulique de bande 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 la tendeuse de bande et de la force f^{max} (mesurée ou calculée à partir des pressions) nécessaire pour faire arriver le tendeur hydraulique de bande à cette position sont stockées.

- **Test de position minimale** : Dans une configuration de commande de position en boucle fermée, le tendeur hydraulique de bande est déplacé jusqu'à sa position minimale ; un interrupteur de position indique au système de commande que la tendeuse de bande 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) nécessaire pour faire arriver le tendeur hydraulique de bande à cette position sont stockées.
- **Test de réponse à un échelon de consigne de force** : le tendeur hydraulique de bande est positionné à l'extrémité supérieure. Dans cette position, et dans une configuration de régulation de la force, une consigne en échelon positif puis négatif est appliquée au système. le tendeur hydraulique de bande agit contre un élément plastique de basse élasticité situé à l'extrémité fixe, 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)** : Le système est déplacé entre une position inférieure de 10° et une position supérieure de 70° en suivant une consigne de position croissante/décroissante. La trajectoire de la position angulaire du tendeur hydraulique de bande et 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.

6.5 Détection de défauts du tendeur hydraulique de bande.

Après un raisonnement de causal, un graphe causal simple a été créé (Figure 6-3). Les variables impliquées dans le graphe causal sont : la variable de commande v_{ref} , la pression du côté du piston p_p , la pression du côté de la tige p_r , la force du côté de la tige f (mesurée ou calculée à partir des pressions) et la position angulaire du tendeur hydraulique de bande x_{meas} . La variable v_{ref} est calculée par le système de commande numérique, convertie en une tension analogique et injectée à la servo-vanne. De l'analyse physique, il est possible de conclure que la variable de commande v_{ref} a une influence sur la position de vanne (non mesurée) et cette position a à la fois une influence sur les pressions p_p et p_r , qui ont aussi influence une sur l'autre. La force appliquée à la tige est le résultat de la différence des deux pressions p_p et p_r . Finalement, c'est la force sur la tige qui provoque le déplacement linéaire de la tige, qui est converti en un déplacement angulaire par le système mécanique, mesuré x_{meas} .

6.5.1 Détection à partir de données simulées.

Des données de simulation ont été obtenues à partir d'un simulateur fondé sur un modèle de connaissance compliqué fournit par SMS-DEMAG [Stuecher and Metzul 2004].

Deux modèles différents ont été construits pour le tendeur hydraulique de bande. Le premier est lié au test PPMS de réponse à l'échelon de consigne de force et le deuxième au test PPMS de réponse à la rampe de consigne de position. Même si la structure du modèle causal est la même pour les deux modèles, la dynamique de l'enrouleur pour chaque test est très différente pour diverses raisons :

- La zone de déplacement de la position du tendeur hydraulique de bande dans le test de position est très grande par rapport au test de force (quelques millimètres ou fractions de millimètre).
- Dans le test de force le tendeur hydraulique de bande agit contre une force externe produite par un élément externe. Cette force n'est pas présente dans le test de position (la seule force externe est le propre poids de le tendeur).

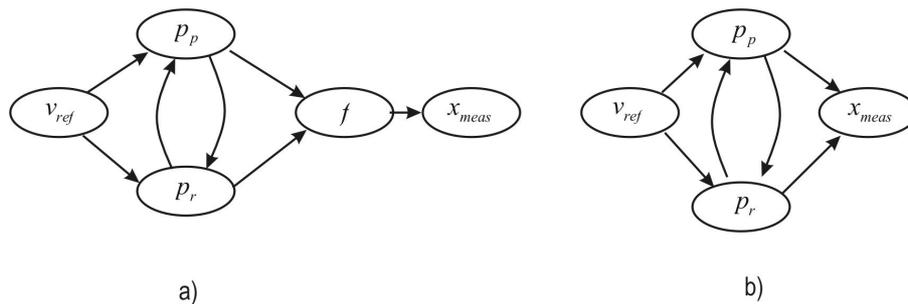


Figure 6-3. a) Graphe causal pour le système avec capteur de force b) graphe causal pour le système sans capteur de force.

Différents tests de détection sont construits à partir du graphe causal. On suppose que la mesure de la force n'est pas disponible, ce qui correspond à la situation industrielle. Chaque test de détection évalue le comportement d'un ensemble de composants :

Tableau 6-1 Test de détection pour le système simulé.

Test de détection.	Support.
DetLocal_pr	C5 : Capteur de pression du côté de la tige. C1 : Servo-vanne. C2 : Tuyau hydraulique du côté de la tige. C4 : Cylindre hydraulique
DetLocal_pp	C6 : Capteur de pression du côté du piston. C1 : Servo-vanne. C3 : Tuyau hydraulique du côté du piston. C4 : Cylindre hydraulique
DetLocal_xmeas	C5 : Capteur de pression de côté de la tige. C6 : Capteur de pression de côté du piston. C4 : Cylindre hydraulique. C7 : Articulation mécanique. C8 : Capteur de position angulaire.
DetGlobal_xmeas	C1 : Servo-vanne. C3 : Tuyau hydraulique du côté du piston. C2 : Tuyau hydraulique du côté de la tige. C4 : Cylindre hydraulique.

	C7 : Articulation mécanique. C8 : Capteur de position angulaire.
--	---

Le simulateur du tendeur hydraulique de bande permet d'appliquer diverses défaillances :

- a. Changement du gain du capteur de position angulaire.
- b. Addition (soustraction) d'une valeur constante sur la mesure de position angulaire.
- c. Jeu dans l'accouplement du capteur de position angulaire.
- d. Changement du gain du capteur de pression du côté du piston.
- e. Addition (soustraction) d'une valeur constante à la mesure de pression du côté du piston.
- f. Changement du gain du capteur de pression du côté de la tige.
- g. Addition (soustraction) d'une valeur constante à la mesure de pression du côté de la tige.
- h. Fuite externe dans les tuyaux hydrauliques.
- i. Fuite interne dans le cylindre hydraulique entre le piston et le mur du cylindre.
- j. Usure de la servo-vanne.
- k. Réduction de la pression du système.
- l. Incrément de la force de frottement du cylindre hydraulique.
- m. Réduction de la rigidité de la tige du cylindre hydraulique.

Le scénario de simulation présenté sur la Figure 6-4 correspond au test de réponse à l'échelon de consigne de force. Une déviation de 1° est ajoutée à la valeur de x_{meas} au temps 0.1sec.

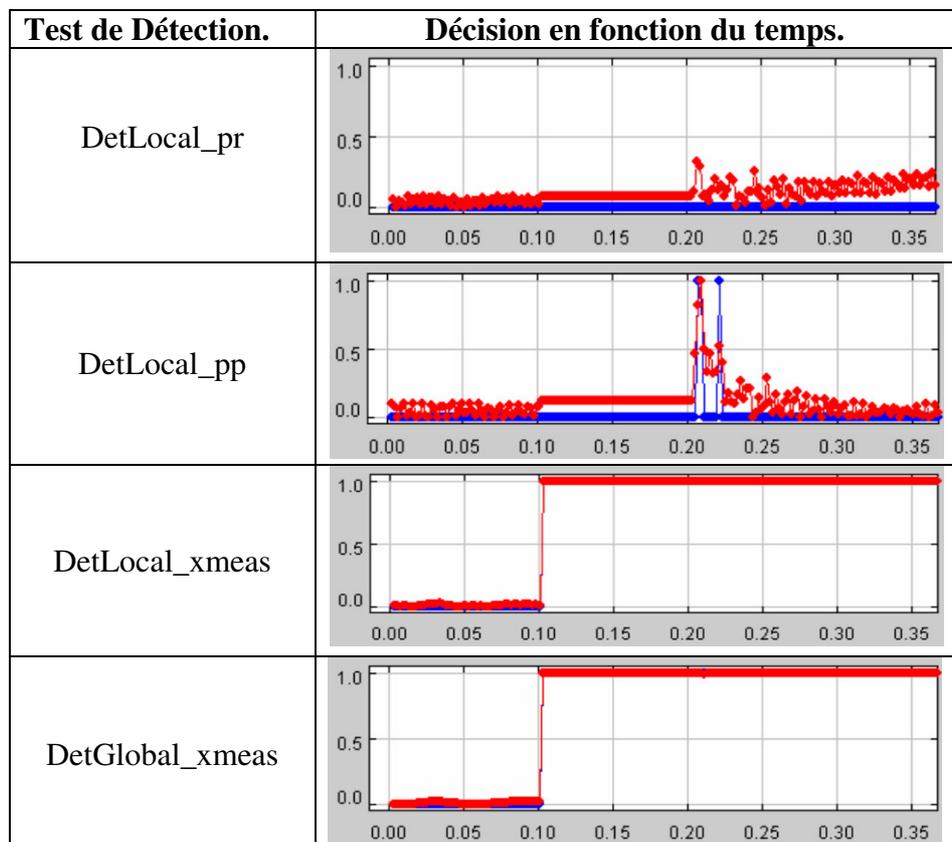


Figure 6-4. Test de réponse à l'échelon de la consigne de force, défaut sur le capteur de position angulaire. Les tests local et global sur x_{meas} montrent un défaut sur l'ensemble de composants { C1, C2, C2, C4, C5, C6, C7, C8 }.

Les tests locaux et globaux réalisés sur le noeud x_{meas} détectent des inconsistances entre les valeurs de x_{meas} et les valeurs de référence de bon comportement générés par le graphe causal. Des contradictions ont été également détectées à 0.2 sec sur la pression du côté du piston. Elles sont probablement dues à une dynamique rapide non modélisée. Comme les tests DetLocal_pp et DetLocal_pr sont consistants, mais DetLocal_xmeas et DetGlobal_xmeas ne le sont pas, une première conclusion est que l'ensemble des composants suivants pourraient être la cause primaire des défauts observés : *cylindre hydraulique, articulation mécanique et capteur de position angulaire*. Les capteurs de pression sont exonérés par les tests DetLocal_pp et DetLocal_pr. L'ensemble des scénarios de simulation de défauts a été utilisé pour évaluer l'algorithme de détection de défauts fondé sur le graphe causal. Le Tableau 6-2 et le Tableau 6-3 résument les résultats de cette évaluation. Un « 1 » indique que le défaut a été détecté et un « 0 » qu'il n'a pas été détecté.

Tableau 6-2. Scénarios de simulation de défauts du tendeur hydraulique de bande à partir des tests PPMS de position.

	Scénarios de défaillance.	detGlobal_pr	detGlobal_pp	detGlobal_xmea	detLocal_pp	detLocal_pr	detLocal_xmeas
1	Système sans défaut.	0	0	0	0	0	0
2	Changement du gain du capteur de position angulaire de 100% à 110%.	0	0	0	0	0	0
3	Addition d'une valeur constante de +1° sur la mesure de position.	0	0	1	0	0	0
4	Jeu de 0,5° sur l'accouplement de capteur de position angulaire.	0	0	0	0	0	0
5	Changement du gain de 100% à 110% sur le capteur de pression du côté de piston.	0	1	0	0	1	0
6	Addition d'une valeur constante de +5bars sur la mesure pression du côté de piston.	0	0	0	0	0	0
7	Changement du gain de 100% à 110% du capteur de pression du côté de la tige.	1	0	0	1	0	1
8	Addition d'une valeur constante de +5bars sur la mesure pression du côté de la tige.	0	0	0	0	0	0
9	Fuite externe de 5e-7 m ² dans le tuyau hydraulique du côté du piston.	0	0	1	0	0	1
10	Fuite externe de 5e-7 m ² dans le tuyau hydraulique du côté de la tige.	0	0	0	0	0	1
11	Fuite interne de 1e-6m ² dans le cylindre entre le piston et le cylindre.	1	1	1	1	1	1
12	Usure des bords de la servo-vanne, chevauchement réduit de 1%	1	1	1	1	1	0
13	Réduction de la pression du système de 270 bars à 200 bars.	0	0	1	0	0	1
14	Incrément de la force de frottement du cylindre hydraulique de 1200N à 2400N.	0	0	0	0	0	0
15	Réduction de la rigidité de tige du cylindre hydraulique de 2e9N/m. à 1.4e9 N/m	0	0	0	0	0	0

Tableau 6-3. Scénarios de simulation de défauts du tendeur hydraulique de bande à partir des tests PPMS de force.

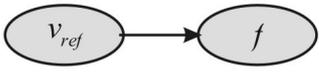
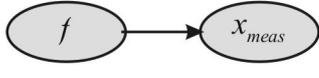
	Scénarios de défaillance.	detGlobal_pp	detGlobal_pr	detGlobal_xmea	detLocal_pp	detLocal_pr	detLocal_xmeas
1	Système sans défaut.	0	0	0	0	0	0
2	Changement du gain en le capteur de position angulaire de 100% à 110%.	1	0	1	1	0	0
3	Addition d'une valeur constante de +1° sur la mesure de position.	0	0	0	0	0	0
4	Jeu de 0,5° sur l'accouplement de capteur de position angulaire.	0	0	0	0	0	0
5	Changement du gain de 100% à 110% sur le capteur de pression du côté de piston.	1	0	0	1	0	0
6	Addition d'une valeur constante de +5bars sur la mesure pression du côté de piston.	0	0	0	0	0	0
7	Changement du gain de 100% à 110% du capteur de pression du côté de la tige.	0	1	0	0	1	1
8	Addition d'une valeur constante de +5bars sur la mesure pression du côté de la tige.	0	0	0	0	0	0
9	Fuite externe de 5e-7 m ² dans le tuyau hydraulique du côté du piston.	1	0	1	1	1	1
10	Fuite externe de 5e-7 m ² dans le tuyau hydraulique du côté de la tige.	1	1	0	1	1	1
11	Fuite interne de 1e-6m ² dans le cylindre entre le piston et le cylindre.	1	1	1	1	1	1
12	Usure des bords de la servo-vanne, chevauchement réduit de 1%	1	1	1	1	1	1
13	Réduction de la pression du système de 270 bars à 200 bars.	1	0	1	1	0	1
14	Incrément de la force de frottement du cylindre hydraulique de 1200N à 2400N.	1	0	0	1	0	0
15	Réduction de la rigidité de tige du cylindre hydraulique de 2e9N/m. à 1.4e9 N/m	0	0	0	0	0	0

6.5.2 Test de détection en utilisant des données réelles.

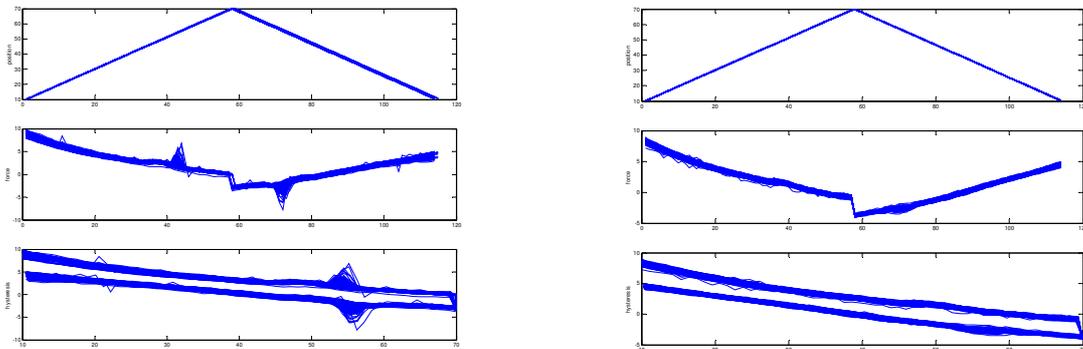
Dans la situation réelle, toutes les données ne sont pas stockées par le PPMS après que les tests de PPMS sont réalisés, par exemple, les pressions p_p et p_r sont mesurées pendant le test, mais la variable stockée est la force, calculée à partir de ces deux pressions. De cette façon, seule la force calculée du côté de la tige f et la mesure de position angulaire x_{meas} sont disponibles à partir du test de position. De la même façon, seule la variable de commande v_{ref} et la force f du côté de la tige sont disponibles à partir du test de force. Étant donné le nombre réduit de variables disponibles, deux graphes causaux très simples ont été construits, me première correspondant au test de force de PPMS (Tableau 6-4,a) et l'autre au test de position de PPMS (Tableau 6-4,b). Puisque ces test sont réalisés à différents instants et avec

différents périodes d'échantillonnage, il n'est pas possible de créer un modèle causal plus grand comprenant v_{ref} , f et x_{meas} .

Tableau 6-4 Test de détection pour le système réel.

Detection_test_1	Support
	C5 : Capteur de pression du côté de la tige. C6 : Capteur de pression du côté du piston. C4 : Cylindre hydraulique. C1 : Servo-vanne.
Detection_test_2	Support
	C5 : Capteur de pression du côté de la tige. C6 : Capteur de pression du côté du piston. C4 : Cylindre hydraulique. C7 : Articulation mécanique. C8 : Capteur de position angulaire.

Près de deux années d'enregistrements de tests PPMS sur des vraies tendeuses hydrauliques de bande ont été fournis par SMD-DEMAG. Les tests ont été exécutés dans la période janvier 2002-novembre 2003. À partir de cette information, il est possible d'affirmer que la plupart des défauts sont liés au cylindre hydraulique et aux articulations mécaniques ; le fonctionnement de ces deux composants est vérifié pendant les tests PPMS de position. Les données ont été analysés mois par mois et comparés avec ceux qui représentent le comportement normal du tendeur ().



a) Enregistrement des tests de position correspondant au mois de février 2002. b) Enregistrement des tests de position correspondant au mois de mars 2002.

Figure 6-5. Test de position du tendeur hydraulique. Dans la figure a) il est possible d'observer une déformation dans la courbe d'hystérésis. La figure b) montre un comportement presque normal.

Comme déjà dit, le test de position de PPMS consiste en un changement en rampe de la référence de la position angulaire de l'enrouleur. La référence va de 10° à 70° en 20 secondes et retourne à 10° en un temps identique. Comparée à la dynamique rapide de l'enrouleur hydraulique, la vitesse de changement de la référence est très lente ; comme

résultat, les états transitoires ne sont pas présents dans les résultats du test, ce test peut être considéré comme un test statique. Ainsi, la relation entre la force et la position a été approximée avec une relation linéaire :

$$x_{meas} = k * force$$

En fait, à cause des caractéristiques asymétriques du cylindre, deux gains différents sont trouvés, k_1 pour le mouvement ascendant de l'enrouleur et k_2 pour le mouvement descendant. La Figure 6-6 et la Figure 6-7 montrent les résultats des tests de détection de défauts de l'enrouleur hydraulique effectués par le CBDA, la première figure correspond au cas sans défaut et la deuxième au cas avec défaut. L'algorithme utilise pour sa décision les fonctions de validité de modèle et de décision montrées sur la Figure 6-8.

La Figure 6-7 montre les résultats du test de détection de défaut. Ce comportement est dû à l'augmentation des forces de friction autour de la position de 50 degrés.

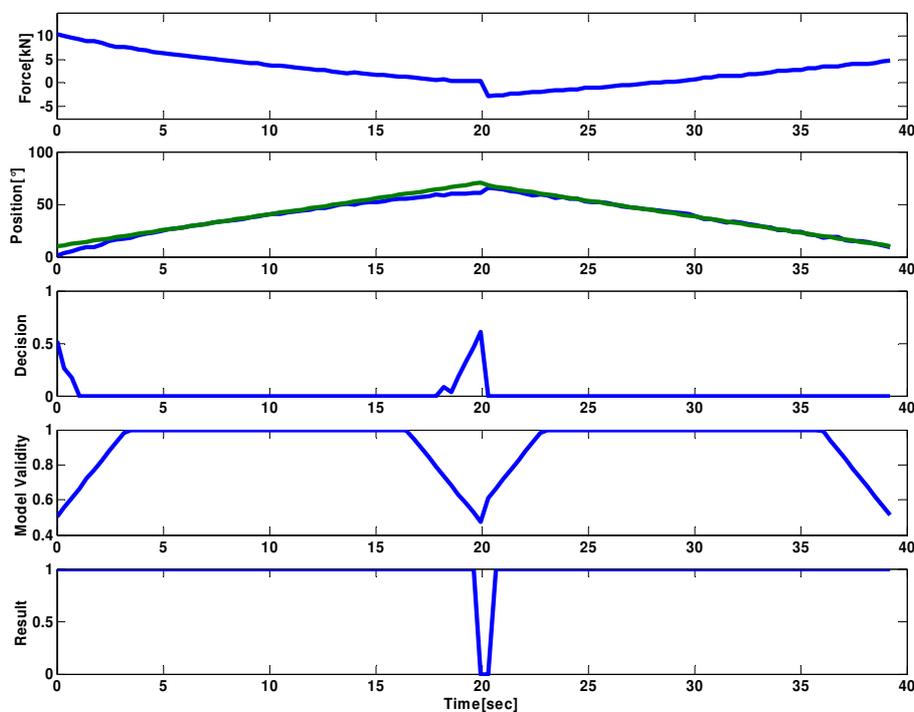


Figure 6-6. Test de position –comportement normal. Le test de détection donne un valeur de défaut autour de 0 et 20 secondes, mais avec un validité de modèle basse (à cause de l'approximation linéaire)

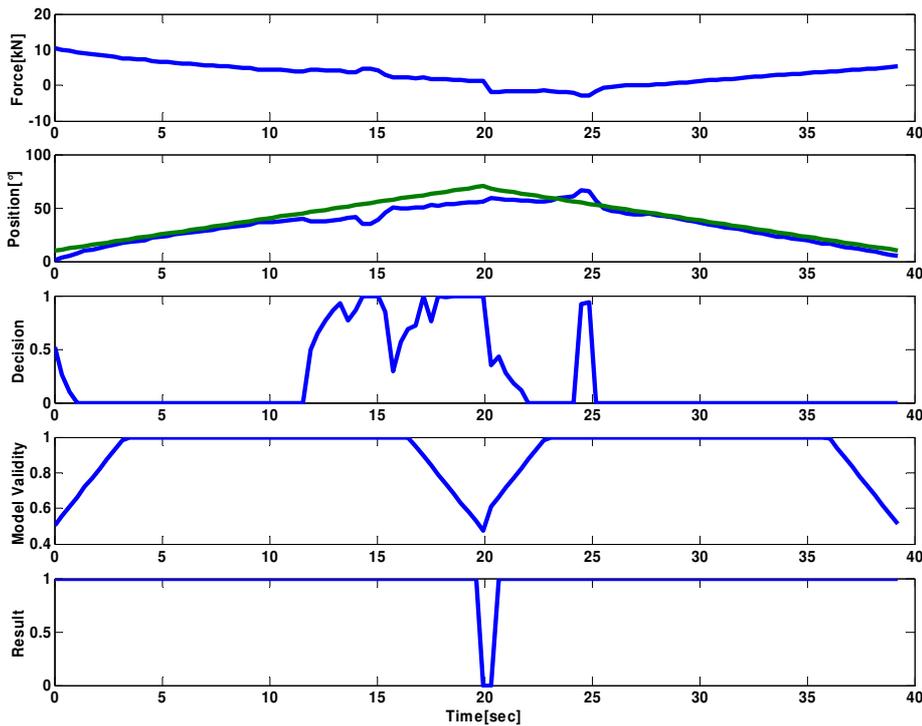


Figure 6-7. Test de position –comportement en défaut. Le test de détection donne une valeur de défaut grande autour de 15 secondes, cette fois avec une validité de modèle grande.

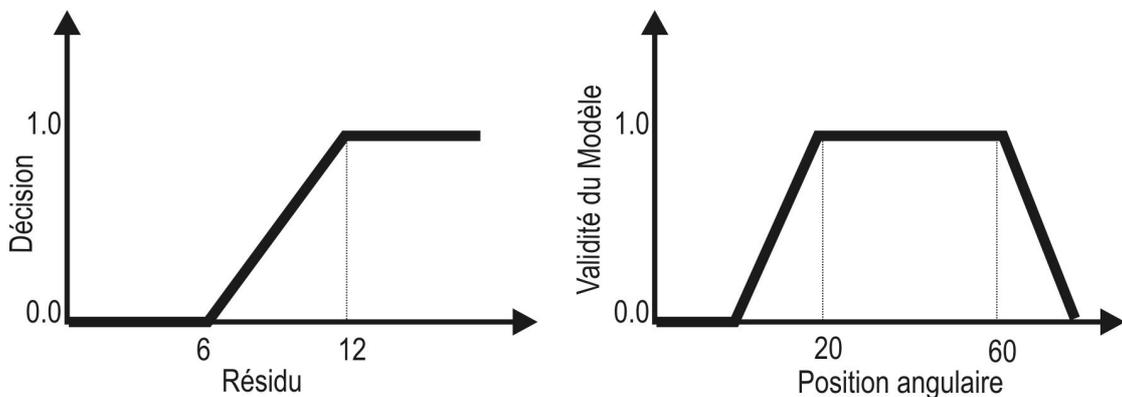


Figure 6-8. Modèles pour la décision et l'évaluation de la validité du modèle.

6.6 Conclusion.

Dans ce chapitre, la tendeuse hydraulique de bande et son rôle dans un procédé de laminage ont été décrits. Ensuite, les méthodes de diagnostic fondées sur le graphe causal ont été appliquées pour la détection de défauts d'une tendeuse hydraulique. Les algorithmes ont été évalués tout d'abord avec des scénarios de défauts provenant d'un simulateur de l'enrouleur hydraulique. Finalement, des données d'un vrai enrouleur hydraulique ont été utilisées comme scénario de test.

Les graphes causaux construits pour ce composant sont très simples. Cependant, il est clair que les algorithmes de détection peuvent être utilisés avec des systèmes plus complexes sans perte de généralité.

Les problèmes de conduite et/ou de maintenance sur ce sous-système ne sont pas assez complexes pour que les outils d'aide à la décision aient pu montrer toutes leurs capacités.

Chapitre 7

Conclusions et Perspectives.

La supervision de procédés complexes représente un problème de prise de décisions par l'opérateur humain pour la solution des problèmes liées à l'opération et la maintenance du procédé face à des situations de fonctionnement anormal. Bien que les techniques d'analyse de sûreté de fonctionnement permettent de réduire ou d'éliminer la présence de certains défauts et leurs effets, ou de créer des séquences automatisées de reprise, il existe toujours la possibilité de la présence de modes de dysfonctionnement non prévus qui doivent être gérés par l'opérateur humain au niveau de la supervision. Dans son rôle de surveillant de procédé, l'opérateur humain a comme tâches :

- la détection de situations anormales,
- la localisation des composants défectueux,
- la prévision du comportement du système en vue des actions à entreprendre et leur exécution comme :
 - la reprise en mode de commande manuelle,
 - la compensation de l'effet des défauts par une action sur le système dans sa configuration actuelle,
 - la sélection d'une configuration différente du procédé, vis à vis de l'isolation physique du composant défectueux,
 - le changement des objectifs du système,
 - la décision de l'exécution d'un arrêt d'urgence, etc.

Dans son rôle de superviseur, l'opérateur humain doit faire usage de systèmes d'aide à la décision qui soient compatibles avec son processus de raisonnement pour la résolution de problèmes.

Dans ce mémoire, nous avons présenté une approche pour l'implémentation des systèmes avancés d'aide à supervision fondés sur une architecture multi-agents, produit d'un projet de collaboration européenne. 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.

MAGIC est un système d'aide à la décision pour la supervision de procédés complexes basé sur une architecture multi-agents. 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.

On a présenté une méthode générale pour l'obtention de symptômes à partir d'un graphe causal. Les nœuds d'un graphe causal représentent les variables utiles pour la supervision. Ses arcs représentent la causalité entre ces variables. Dans le cas le plus simple, quand aucune information quantitative n'est disponible, le graphe causal est un modèle purement qualitatif. Quand une modélisation quantitative est possible, comme nous l'avons supposé dans notre travail, les arcs sont représentés par des équations analytiques, voire des fonctions de transfert dans le cas linéaire. Un de nos apports principaux par rapport aux travaux antérieurs sur la détection de défaut à partir d'un graphe causal est l'application de cette méthode dans un contexte de diagnostic distribué. Un autre concerne l'évaluation de l'incertitude des symptômes dans ce contexte, ce qui nous a conduit à proposer l'analyse séparée de la validité de la décision et de la validité du modèle de référence utilisé pour la détection de défauts. L'utilité d'un système de ce type au niveau industriel est fortement basée sur les outils conçus pour faciliter sa configuration et son adéquation au procédé étudié. Par conséquent, un soin particulier a été mis dans la création d'outils de configuration faciles d'emploi.

On a proposé ensuite des méthodes pour le pronostic de l'évolution des variables. La première est basée sur un réseau de neurones artificiel. La simulation récursive permet la prédiction multi-pas. L'avantage principal de cette méthode est sa capacité d'auto-apprentissage. Cette caractéristique est utile quand aucune connaissance préliminaire sur le signal n'est disponible. Son inconvénient principal est le temps de calcul nécessaire pour effectuer cet apprentissage et la nécessité de disposer d'une base d'apprentissage conséquente. La deuxième méthode présentée est basée sur l'analyse de tendances qualitatives. La complexité informatique modérée de cette méthode permet son utilisation en ligne. Cependant les différents seuils qui sont nécessaires dans cette méthode pour la segmentation de l'information et la classification doivent être choisis a priori par l'ingénieur spécialiste du procédé. On peut envisager d'utiliser ces deux méthodes dans des systèmes d'aide à la supervision pour la prévision des variables ou des symptômes critiques, en fonction de la précision exigée, des vitesses de variation des signaux et de la connaissance préliminaire sur ceux-ci.

Le problème de supervision et de reconfiguration est analysé du point de vue des systèmes à grande échelle sous la responsabilité d'opérateurs humains. Les Conseils d'Action

produits par le DSA ont pour objectif de maintenir les variables non directement impliquées avec le composant en défaut à un niveau proche de leur référence ou à un niveau sûr. Comme les modèles analytiques ne peuvent pas être envisagés pour représenter une installation avec des centaines de variables et plusieurs modes de défaut imprévisibles, on a proposé dans ce mémoire une analyse fondée sur un modèle qualitatif simple qui est le graphe orienté utilisé pour la génération de symptômes. Notre contribution principale a consisté à proposer d'utiliser plusieurs graphes pour représenter diverses configurations d'un même procédé. Les Conseils d'Action font usage de la redondance physique du système (dans les cas de reconfiguration) ou de la redondance analytique, représentée par les liens causaux entre les nœuds affectés et leurs moyens d'action (dans une même configuration). Les nœuds du graphe représentent, comme on l'a dit plus haut, les variables du procédé, parmi lesquelles des moyens d'actions peuvent être définis a priori puis retrouvés en ligne, en fonction de la situation de défaut particulière.

Des moyens de maintenance peuvent être attachés aux composants fautifs. Ils correspondent à de la connaissance experte sur le procédé. Finalement, la priorité des Conseils d'Action et de Maintenance produits par le DSA est choisie par l'opérateur grâce aux résultats de pronostic et à son expertise.

Pour finir, 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 de diagnostic fondées sur le graphe causal ont été appliquées pour la détection 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. Bien que les graphes causaux construits pour ce composant soient très simples, les algorithmes de détection pourraient être utilisés avec des systèmes plus complexes sans perte de généralité. Cette application a montré toutefois que dans certains cas, la détection est impossible parce que le système en défaut n'est pas excité. Un problème similaire est rencontré en identification de systèmes, impossible sur un procédé mal excité. Cela explique tout l'intérêt qui est porté actuellement aux méthodes de *diagnostic actif*. L'Enrouleur Hydraulique n'était toutefois pas assez complexe pour permettre de mettre en valeur correctement l'intérêt des outils d'aide à la décision et à la maintenance, qui ont été illustrés par un exemple pédagogique.

7.1 Dissemination

Les travaux de recherches dans cette thèse ont été disséminés dans des conférences internationales à comité de lecture :

- Charbonier, S., C. Garcia-Beltran, C. Cadet and S. Gentil (Submitted). "Trends extraction and analysis for complex system monitoring and decision support."
- Garcia-Beltran, C., M. Exel and S. Gentil (2003). An Interactive Tool for Causal Graph Modeling For Supervision Purposes. IEEE International Symposium on Intelligent Control (ISIC 2003), Houston, Tx.
- Garcia-Beltran, C. and S. Gentil (2001). Causal reasoning for human supervised process reconfiguration. Proc. IEEE International Symposium on Intelligent Control ISIC 2001, Mexico City, Mexico.

- Garcia-Beltran, C. and S. Gentil (2001). Decision Support System for Supervised Reconfiguration. Proc. of the IAR-ICD Workshop, Strasbourg, France.
- Garcia-Beltran, C. and S. Gentil (2002). Multiagent based decision system for process reconfiguration. Congreso Latinoamericano de Control Automatico, Guadalajara, Mexico.
- Garcia-Beltran, C., S. Gentil and S. Charbonier (2003). Qualitative trends and Neural Network based symptom prognosis. Workshop on Advanced Control and Diagnosis, University Duisburg - Essen , Germany.
- Garcia-Beltran, C., S. Gentil and R. Stuecher (2004). Causal-based diagnosis of a hydraulic looper. Accepted for its publication in IFAC-MMM 04, Nancy, France.
- 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 hot rolling mill. Workshop on Advanced Control and Diagnosis, University Duisburg - Essen , Germany, IAR-ICD Working Group.
- Lesecq, S., S. Gentil, M. Exel and C. Garcia-Beltran (2003). Diagnostic Tools for a Multi-agent Monitoring System. CESA 2003, Lille, France.

7.2 Perspectives

Une première perspective que l'on peut proposer serait de raffiner la méthode de diagnostic causal, bien qu'elle ait déjà été étudiée depuis longtemps par notre équipe. En particulier, les algorithmes de simulation implémentés dans le système MAGIC sont très simples et pourraient être robustifiés numériquement. Il faudrait sûrement, pour bon nombre d'applications industrielles, pouvoir implémenter des relations non linéaires entre nœuds, suivant en cela les travaux préliminaires de Vadam [Vadam, Montmain et al. 1997]. Pour déterminer les différents graphes causaux correspondant à diverses configurations d'un procédé de façon automatique, il faudrait poursuivre les travaux proposés dans [Thevenon and Flaus 2000]. Notre éditeur de graphes suppose en effet que l'ingénieur détermine a priori les relations causales entre toutes les variables du procédé, grâce à sa connaissance physique de celui-ci. Lorsqu'il est en possession d'un système d'équations formelles décrivant le fonctionnement du procédé, il pourrait passer par l'un des outils proposés dans la littérature pour transformer ce système en graphe causal [Heim, Cauvin et al. 2001], même si plusieurs configurations sont possibles, comme évoqué dans [Flaus and Gentil 2003]. Ce type de modélisation causale n'a pas été abordé dans MAGIC. Pour finir, on peut évoquer la possibilité de relier directement le modèle de la structure du procédé (diagramme PID) au modèle causal de comportement, ce qui permettrait d'élaborer directement les algorithmes de diagnostic au moment de la conception d'une installation.

Une réflexion approfondie s'impose sur 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 très intéressante 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 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 de

Diagnostic joue ce rôle de superviseur, à partir des symptômes élaborés par divers Agents de Diagnostic, pour localiser le ou les composants défectueux. De même, l'Agent d'Aide à la Décision adopte 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 connaît 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, mais assez centralisée. 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. Il paraît tout à fait envisageable de réaliser cette coordination au moyen du raisonnement causal, qui permettrait de limiter la communication entre agents.

BIBLIOGRAPHIE

- Albert, M., T. Langle and H. Worn (2004). Real-time Requirements in Diagnostic Systems. To Appear on : IFAC MMM 2004, Nancy, France.
- Alty, J. L. (1997). "Software Agents, Interfaces and Complex Decision Making." The Institution of Electrical Engineering.
- Åström , K., P. Albertos, M. Blanke, A. Isidori, W. Shaufelberger and R. Sanz (2001). Control of Complex Systems. London, Springer-Verlag,.
- 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.
- Avizienis, A., J. C. Laprie and B. Randell (2001). Rapport LAAS N°01145: Fundamental concepts of dependability. Toulouse, LAAS: 19p.
- Bainbridge, P. (1993). The change in concepts needed to account for human behaviour in complex dynamic tasks. IEEE-SMC Congress, Le Touquet, France.
- Bar-Yam, Y. (1997). Dynamics of complex systems, Addison-Wesley.
- Basseville, M. and I. Nikiforov (1993). Detection of abrupt changes, Prentice-Hall.
- Blanke, M. (1996). "Consistent Design of Dependable Control Systems." Control Eng. Practice **4**: 1305-1312.
- Blanke, M., C. Frei, F. Kraus, R. Patton and M. Staroswiecki (2000). What is fault tolerant control? 4th IFAC Safeprocess 2000, Budapest, Hungary.
- Blanke, M., M. Kinnaert, J. Lunze and M. Staroswiecki (2003). Diagnosis and Fault-Tolerant Control, Springer-Verlag.
- Boné, R. and M. Crucianu (2002). Multi-step-ahead prediction with Neural networks: A Review. Proceedings of the ACSEG 2002, Boulogne sur Mer, France.
- Borne, P., B. Fayech, S. Hammadi and S. Maouche (2003). "Decision Support System for Urban Transportation Networks." IEEE Trans. on Systems, Man and Cybernetics-Part C **33**(1): 67-77.
- Box, G. E., P. Jenkins, G. M. and G. C. Reinsel (1994). Time series Analysis, Forecasting and Control, Prentice Hall.
- Brungali, D. (1998). From Objects to Agents: Software Reuse for Distributed Systems. Torino, Politecnico di Torino.
- Carlsson, C. and E. Turban (2002). "DSS: directions for the next decade." J. Dec. Support Systems **33**(2): 105-110.
- Cassar, J. P., R. G. Cocquempot and M. Staroswiecki (1994). "Approche structurelle de la conception de systèmes de surveillance pour des procédés industriels complexes." Revue Européenne de Diagnostic et Sûreté de Fonctionnement **2**: 179-202.
- Chaib-draa, B., I. Jarras and B. Moulin (2001). Systèmes multi-agents: principes généraux et applications. Principes et architecture des systèmes multi-agents. Paris, Hermes science.
- Charbonnier, S., G. Becq, L. Biot, P. Y. Carry and J. P. Perdrix (2002). On line segmentation algorithm for ICU continuously monitored data. 15 th-IFAC World Congress, Barcelona.
- CHEM-Consortium (2000). Advanced decision support system for chemical/petrochemical manufacturing processes: <http://www.chem-dss.org>

- Chen, J. and R. Patton (1995). Design of Robust Structured and Directional Residuals for Fault Isolation via Unknown Input Observers. Proceedings of 3rd European Control Conference, Rome, Italy.
- Colomer, J., J. Meléndez and F. Gamero (2002). Qualitative Representation of process trends for situation assessment based on cases. 15 th-IFAC World Congress, Barcelona.
- Dash, S. and Venkatasubramanian (2001). Offline safety analysis for online fault diagnosis.
- de Kleer, J. and J. S. Brown (1986). "Theories of Causal Ordering." Artificial intelligence **29**(1): 33-62.
- Deckers, J., Jepsen, O.N., Latzel, S., Metz, H., Stuecher, R (2003). Condition monitoring and failure diagnosis in plants of the metals industry. Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2003, Washington, D.C.,USA.
- Demazeau, Y. and J.-P. Briot (2001). Principes et architectures des systèmes multi-agents. Paris., Ed. Hermes.
- DIVA <http://www.gigascale.org/diva/>:
- Dvorak, D. and B. Kuipers (1991). "Process monitoring and diagnosis: A model-based approach." IEEE Expert **6**(3): 67-74.
- Evsukoff, A. (1998). Le Raisonnement Approche pour la Surveillance de Procédés. Laboratoire d'Automatique de Grenoble. Grenoble, INPG.
- Evsukoff, A., S. Gentil and J. Montmain (2000). "Fuzzy Reasoning In Co-operative Supervision Systems." Control Engineering Practice **8**: 389-407.
- Fayech, B., S. Hammadi, S. Maouche and P. Borne (2002). Approche coopérative pour la régulation d'un réseau de transport multimodal. Congrès Int. Francophone d'Automatique, Nantes, France.
- FIPA (2003). Foundation for Intelligent Physical Agents (FIPA). <http://www.fipa.org/>:
<http://www.fipa.org/>
- Flaus, J.-M. and S. Gentil (2003). Hybrid System Automatic Modeling For Diagnostic Purposes. IFAC-SAFEPROCESS 2003, Washington, USA.
- Forbus, K. D. (1984). "Qualitative Process Theory." Artificial Intelligence **24**(1): 85-168.
- Frydman, C., M. LeGoc, M. Torres and L. Giambiasi (2001). The diagnosis approach used in SACHEM. 12th International Workshop on Principles of the Diagnosis, Sansicario, Italie.
- Gandibleux, X. (1993). Système d'aide à la décision pour la conduite de processus perturbés, une approche hybride fondée sur l'intelligence artificielle, la programmation linéaire et l'aide multicritère à la décision. Application à la mobilisation de la réserve tertiaire d'EDF., Université de Valenciennes.
- Garcia-Beltran, C., M. Exel and S. Gentil (2003). An Interactive Tool for Causal Graph Modeling For Supervision Purposes. IEEE International Symposium on Intelligent Control (ISIC 2003), Houston, Tx.
- 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.
- Gentil, S., P. Dziopa and J. Montmain (1999). Hierarchical causal model for complex plant supervision. 3rd IMACS International Multiconference on Circuits, Systems, Communications and Computers (CSCC'99), Athènes, Gr.
- Heim, B. (2003). Approches ensembliste et par logique floue pour le diagnostic causal de procédés de raffinage. Application à un pilote de FCC. Laboratoire d'Automatique de Grenoble. Grenoble, INPG.

- Heim, B., S. Cauvin and S. Gentil (2001). FCC Process Diagnosis Based On Causal And Heuristic Approach. IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries, Seoul, Korea.
- Hoc, J. M. (1993). Mean features of the human supervision of a long response latency process. Proc. of the IEEE Conference on Man, Systems and Cybernetics, Le Touquet, France.
- Hoc, J. M. (1996). "Supervision et contrôle de processus." La cognition en situation dynamique, **Presses Universitaires de Grenoble**.
- Isermann, R. (1997). "Supervision, fault detection and fault diagnosis methods -an introduction." Control Eng. Practice **5**(5): 639-652.
- Isermann, R. and P. Ballé (1997). Terminology in the field of supervision, fault detection and diagnosis. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (Safeprocess '97), Hull, England, Elsevier.
- Isermann, R., R. Schwarz and S. Stolz (2002). Fault-Tolerant Drive-by-Wire Systems. IEEE Control Systems Magazine. **22**: 64-81.
- Iwasaki, S. and H. Simon (1986). "Causality in device behaviour." Artificial intelligence **29**(1-3): 3-32.
- Jämsä-Jounela, S.-L., M. Vermasvuori, P. Endén and Haavisto (2003). "A process monitoring system based on the Kohonen self-organizing maps." Control Eng. Practice **11**.
- Jang, J.-S. R. (1993). "ANFIS: Adaptive-Network-Based Fuzzy Inference System." IEEE Transactions on systems, Man and Cybernetics **23**(3): 665-683.
- Jonsson, E. (1998). An Integrated Framework for Security and Dependability. New Security Paradigms Workshop, Charlottesville, USA.
- Khoualdi, K. and M. Dumas (1994). An alarm processing system using distributed artificial intelligent techniques. Fault Detection, Supervision and Safety for Technical Process, SAFEPROCESS'94, Espoo, Finland.
- Köppen-Seliger, B., T. Marcu, M. Capobianco, S. Gentil, M. Albert and S. Latzel (2003). MAGIC: An Integrated Approach For Diagnostic Data Management and Operator Support. IFAC-SAFEPROCESS 2003, Washington, D.C.
- Lacatusu, C., R. Stücher and M. Capobianco (2002). MAGD001S02: General Specifications for Process Specification and Data Acquisition Agents. Duisburg, UNIDUI.
- Lackinger, F. and W. Nejd (1993). "Diamon: A model-based troubleshooter based on qualitative reasoning." IEEE Expert **8**(1): 33-40.
- Lebold, M. and M. Thurston (2001). Open standards for condition-based maintenance and prognostic. MARCON, Gatlinburg, Tennessee, USA.
- MAGD001S04 (2003). MAGD001S04: Deliverable D5-1: General Specifications for the Process Specification and Data Acquisition Agents. M. Consortium.
- MAGK001R01 (2002). MAGK001R01: Concepts and Features de MAGIC.
- Mandiau, R. (2000a). HDR : Modélisation et Evaluation d'Organisations Multi-Agents. LEMIH, L'universite de Valenciennes et du Hainaut-Cambresis.
- Mandiau, R. (2000b). Modélisation et Evaluation d'Organisations Multi-Agents. Valenciennes, L'universite de Valenciennes et du Hainaut-Cambresis: 128.
- Maquin, D. and J. Ragot (2000). Diagnostic des systèmes linéaires. Paris, Hermès science publications.
- Mathur, A., K. F. Cavanaugh, K. R. Pattipati, P. K. Willet and T. R. Galie (2001). Reasoning and Modeling Systems in Diagnosis and Prognosis. Proceedings of the SPIE Aerosense Conference, Orlando, FL.
- Meystel, I. A. (2001). The tools of intelligence: Are we smart enough to handle them? Proc. European Workshop Intelligent Forecasting, Diagnosis and Control, Santorini, Greece.

- Millot, P. (1996). De la surveillance à la supervision : l'intégration des opérateurs humains. l'Ecole d'Eté d'Automatique de Grenoble, Grenoble, France.
- Montmain, J. (1997). From Diapason Research Program to its Industrial application in nuclear fuel reprocessing. IFAC-SAFEPROCESS '97, Hull, United Kingdom.
- Montmain, J. and S. Gentil (2000). "Dynamic causal model diagnostic reasoning for on-line technical process supervision." Automatica(36): 1137-1152.
- Montmitonnet P., L. (2002). "Objectifs et modélisation." Les Techniques de l'Ingénieur M 3065: 1-12.
- Musen, M. A., R. W. Ferguson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy and S. W. Tu ((in press)). "The Evolution of Protégé: An Environment for Knowledge-Based Systems Development." International Journal of Human-Computer Interaction.
- Orfanidis, S. J. (1996). Optimum Signal Processing, An Introduction. Englewood Cliffs, NJ, Prentice-Hall,.
- Oyeleye, O. O., F. E. Finch and M. A. Kramer (1990). "Qualitative modeling and fault diagnosis of dynamic processes by MIDAS." Chemical Engineering Communications 96: 205-228.
- Patton, R. and J. Chen (1997). "Observer based fault detection and isolation: robustness and applications." Control Eng. Practice 5(5): 671-682.
- 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.
- Pessi, E. and C. Luparia (1992). "MORSAF: un système expert temps réel d'aide aux opérateurs de salle de contrôle d'installation de distillation atmosphérique." Génie Logiciel Temps réel et systèmes à base de connaissances 28.
- Ploix, S., S. Gentil and S. Lesecq (2003). Isolation Decision for a Multi-Agent-Based Diagnostic System. IFAC SAFEPROCESS 2003., Washington, D.C.
- Popieul, J.-C. (2003). HDR: "Diagnostic dans les Systèmes Homme-Machine, Application à la conduite automobile". Laboratoire d'Automatique, de Mécanique et d'informatique Industrielles et Humaines. Valenciennes, L'Université de Valenciennes et du Hainaut-Cambresis.
- Propes, N., S. Lee, G. Zhang, Y. Zhao, G. Vachtsevanos, A. Thakker and T. Galie (2002). A Real-Time Architecture for Prognostic Enhancements to Diagnostic Systems. MARCON, Knoxville, Tennessee.
- Rasmussen, J. (1993). "Diagnostic reasoning in action." IEEE Trans. on Systems, Man and Cybernetics 4(23): 981-991.
- Sanz, R., C. Pfister, W. Schaufelberger and A. de Antonio (2001). Software for Complex Controllers. Control of Complex Systems. London, Springer-Verlag.
- Sanz, R., M. Segarra, A. de Antonio and J. A. Clavijo (1999). ICA: Middleware for Intelligent Process Control. International Symposium on Intelligent Control, Cambridge M.A.
- Staroswiecki, M. (1994). La problématique et les approches de la surveillance des systèmes technologiques. Journées d'Etude S3, Sécurité, Surveillance, Supervision : Détection et Localisation de défaillances, Paris.
- Staroswiecki, M. and A.-L. Gehin (2000). From control to supervision. SAFEPROCESS 2000, Budapest.
- Stolz, S. (1998). Control and Supervision of an electromechanical brake systems. FISITA World Automotive Congr., 2nd Century of the Automobile, Paris, France.
- Stuecher, R. and A. Metzger (2004). Simulations and Measurements of hot rolling mill components for use in diagnostic benchmark tests. IFAC MMM 2004, Nancy, France.

- Thevenon, L. and J. M. Flaus (2000). A hybrid meta-modeling approach suitable for process supervision. IEEE Conference on Management and Control of Production and logistics MCPL 2000, Grenoble, France.
- Trave-Massuyès, L. and R. Milne (1996). "Gas turbine condition monitoring using qualitative model-based diagnosis." IEEE Expert **12**(3): 21-31.
- Travé-Massuyès, L. and R. Pons (1997). Causal ordering for multiple mode systems. 11th Int. Workshop on Qualitative Reasoning about Physical Systems, Cortona, Italy.
- Vachtsevanos, G. and P. Wang (2000). Prognostication: algorithms and performance assessment methodologies. MARCON 2000.
- Vadam, O., J. Montmain and J. P. Cassar (1997). Fault detection using parallel simulations. IFAC Symposium on fault detection, supervision and safety for technical processes, Safeprocess'97, Hull, Grande-Bretagne.
- Venkatasubramanian, V., R. Rengaswamy, K. Yin and S. N. Kavuri (2002). "A review of process fault detection and diagnosis, Part I: Quantitative model-based methods." Computers & Chemical Engineering **27**(2003): 293-311.
- Villemeur, A. (1988). Sûreté de fonctionnement des systèmes industriels, fiabilité- Facteurs humains, Informatisation. Paris, Editions Eyrolles.
- Wakamiya, Y. and I. Nitta (2000). "A quality Control and Diagnostic System for Hot-Rolling Mills." Mitsubishi Electric ADVANCE **92**(December): 30-33.
- Zwingelstein, G. (1995). Diagnostic des Défaillances: Théorie et Pratique pour les Systèmes Industriels. Paris, Hermès.
- Zwingelstein, G. (1996). La maintenance basée sur la fiabilité. Paris, Hermès.