Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes

Pauline Ribot

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :

Systèmes Automatiques

Présentée et soutenue par :
Pauline RIBOT

le : vendredi 4 décembre 2009

Titre :

Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes

---

JURY
Mme Sylviane GENTIL - Présidente
M. Christophe BERENGUER - Rapporteur
M. Vincent COCQUEMPOT - Rapporteur
M. Philippe DAGUE - Examinateur
M. Yannick PENCOLE et M. Michel COMBACAU - Co-directeurs

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
Laboratoire d'Analyse et d'Architecture des Systèmes

Directeur(s) de Thèse :

M. Yannick PENCOLE et M. Michel COMBACAU

Rapporteurs :

M. Christophe BERENGUER et M. Vincent COCQUEMPOT

Remerciements

Ce mémoire marque la fin d'un peu plus de trois années de recherche au sein du Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) du Centre National de la Recherche Scientifique (CNRS). Ce travail n'aurait pu se concrétiser sans l'aide précieuse d'un ensemble de personnes que je tiens à remercier aussi bien pour leur encadrement, leurs compétences que pour leur soutien.

Je remercie tout d'abord M. Malik Ghallab et M. Raja Chatila, les directeurs successifs du LAAS-CNRS, pour m'avoir accueillie au sein de leur laboratoire. Je remercie également Louise Travé-Massuyès pour m'avoir permis d'effectuer ma thèse dans le groupe de recherche DISCO (DIagnostic, Supervision et COnduite).

Je tiens à remercier vivement l'ensemble des membres du jury qui ont accepté d'évaluer ce travail. Je remercie M. Christophe Bérenguer et M. Vincent Cocquempot pour l'examen précis de ce mémoire qu'ils ont réalisé lors de la rédaction de leur rapport, et les questions et remarques lors de la soutenance qui ont permis d'éclaircir certains points difficiles et surtout d'ouvrir le champ des investigations futures. Je remercie également Mme Sylviane Gentil pour sa minutieuse relecture et pour avoir accepté le rôle de Présidente du jury et M. Philippe Dague pour avoir accepté d'être membre du jury.

Je remercie très chaleureusement mes deux directeurs de thèse, Yannick Pencolé et Michel Combacau pour avoir assuré l'encadrement de cette thèse. Leurs compétences complémentaires m'ont permis de mener à bien ce travail. Merci à Yannick pour sa disponibilité, ses précieux conseils et pour m'avoir guidée dans chacune des étapes de cette thèse. Merci à Michel pour m'avoir encouragée et soutenue depuis la Licence et pour m'avoir donné l'envie de devenir enseignant-chercheur.

J'ai depuis toujours voulu travailler dans l'enseignement. Durant ces trois années de thèse, plusieurs de mes anciens professeurs d'université m'ont permis d'enseigner à leur côté. Je remercie particulièrement Gérard Mouney et Yann Labit pour m'avoir fait confiance et m'avoir encouragée dans cette voie dès le Master.

Je tiens à remercier l'ensemble des membres et ex-membres du groupe DISCO pour l'ambiance conviviale qui y régnait tout au long de ma thèse. Merci à Renaud, François, Nuno, Mehdi, Fabien, Nico, Xavier et Hervé pour toutes ces pauses café, ces discussions sur le présent et sur l'avenir de chacun ... Un merci tout particulier à Elodie, ma collègue de bureau devenue une véritable amie, pour m'avoir remonté le moral dans les moments

difficiles et m'avoir fourni la petite dose d'optimisme qui pouvait parfois me manquer.

Un grand merci à mes amis de longue date, David, Emeline, Agnès, Louison et Seb, qui ont su m'apporter leur soutien tout au long de ces trois ans et me changer les idées lorsque j'en avais besoin.

Un énorme merci à ceux qui m'entourent depuis toujours, mes parents et mon frère. Notre petit quatuor sans fausse note m'a toujours donné le courage et la force pour avancer dans la vie. Je conclurai en remerciant celui qui me supporte au quotidien pour sa confiance et pour tout ce qu'il m'apporte.

Table des matières

Introduction

L'efficacité de la maintenance des systèmes industriels est un enjeu économique majeur pour leur exploitation commerciale. Les principales difficultés et sources d'inefficacité résident dans le choix des actions de maintenance. Une action de maintenance consiste à remplacer les équipements du système qui sont en panne qui ne sont plus capables de réaliser leur fonction. Les opérations de maintenance sont coûteuses pour plusieurs raisons.

Tout d'abord, elles nécessitent souvent un arrêt de fonctionnement du système. Dans ce cas, durant toute la phase de maintenance, le système n'est pas opérationnel. Plus la phase de maintenance est longue, plus elle est coûteuse dû à l'indisponibilité du système. Par conséquent la phase de maintenance doit idéalement être réduite aux opérations consistant à remplacer, sans tâtonnement, les équipements réellement en panne. La décision d'une action de maintenance est très complexe et doit reposer sur une surveillance et une analyse intelligente de l'état du système. Un diagnostic de panne est alors nécessaire pour déterminer le plus précisément possible les équipements qui doivent être réparés. Moins le diagnostic est ambigu, plus les opérations de maintenance sont efficaces.

La seconde raison pour laquelle une maintenance peut être coûteuse concerne les cas d'urgence dans lesquels la sécurité ou l'accomplissement de la fonction du système sont mis en jeu. En effet, lorsqu'un équipement tombe soudainement en panne et que le système ne peut plus réaliser sa fonction, des actions de maintenance doivent être automatiquement réalisées pour remettre le système en état de fonctionnement. Ces actions imprévues sont naturellement plus coûteuses car les besoins et services pour la maintenance n'ont pas été anticipés et doivent être rapidement disponibles. Pour minimiser l'occurrence de ce genre de situation, une maintenance préventive peut être envisagée. Les pannes des équipements peuvent être anticipées et corrigées avant de générer de trop importants dégâts qui pourraient provoquer un arrêt imprévu du système.

Le plus souvent, la maintenance préventive repose uniquement sur des analyses de fiabilité qui ne tiennent pas compte des sollicitations influençant réellement les équipements du système tout au long de son fonctionnement. En effet, des sollicitations anormales ou imprévues peuvent accélérer la dégradation des équipements. La maintenance préventive peut être améliorée par un raisonnement de pronostic qui permet d'estimer l'impact de ces sollicitations sur la durée de vie des équipements. En établissant quelle action de maintenance est pertinente à un instant donné, le pronostic aide

également à programmer les futures phases de maintenance.

Cette thèse s'intéresse à l'optimisation de la maintenance des systèmes industriels complexes. Elle propose de mettre en place une architecture de supervision qui intègre des capacités de diagnostic et de pronostic dans l'objectif d'aider à la prise de décisions d'actions de maintenance. Les systèmes complexes qui sont considérés dans cette thèse sont composés d'équipements totalement hétérogènes (matériel, logiciel) et nécessitent plusieurs types de techniques pour être surveillés. Ce travail de thèse présente une description abstraite et homogène d'un système complexe à partir de laquelle il est possible de caractériser un couplage original des problèmes de diagnostic et de pronostic.

Ce manuscrit suit l'organisation suivante :

Le premier chapitre présente le cadre de la maintenance des systèmes complexes et montre l'intérêt d'une stratégie de maintenance préventive. Les concepts relatifs aux problèmes de diagnostic et de pronostic sont introduits et les différentes méthodes existantes pour résoudre ces problèmes sont décrites. Ces méthodes s'appuient, dans chacun des deux domaines, sur une connaissance approfondie du système à considérer.

Le deuxième chapitre propose un cadre de modélisation générique pour un système complexe. Cette modélisation s'appuie sur un ensemble de paramètres caractéristiques des composants, un ensemble de rangs et un ensemble de relations entre ces paramètres pour décrire la structure et le comportement du système complexe. Des modes de fonctionnement pour les composants du système peuvent être définis à partir de ce modèle générique. Le modèle générique et les modes de fonctionnement du système complexe constituent la base de connaissance commune aux problèmes de diagnostic et de pronostic.

Le troisième chapitre présente une architecture générique de supervision qui intègre un module de diagnostic et un module de pronostic ayant pour objectif d'améliorer la maintenance préventive. Le problème de pronostic est défini comme une extension du problème de diagnostic. Ces deux problèmes sont ensuite caractérisés à partir du formalisme générique introduit dans le chapitre 2.

Une fonction de pronostic générique est définie dans le quatrième chapitre. Elle utilise un modèle de Weibull pour représenter de manière unique le résultat de pronostic de chaque composant du système. Cette fonction est adaptative et tient compte des sollicitations influençant réellement chaque composant afin d'estimer leur durée de vie.

Des critères de performance sont définis dans le cinquième chapitre dans le but d'évaluer les résultats obtenus par les fonctions de diagnostic et de pronostic. Une méthodologie de retour sur conception est proposée dans le cadre particulier des systèmes

à événements discrets afin d'améliorer l'efficacité de la fonction de diagnostic par une étude de diagnosticabilité et de précision.

Le sixième chapitre présente le projet ARCHISTIC en collaboration avec Airbus et l'École Nationale d'Ingénieurs de Tarbes (ENIT). L'objectif du projet est de concevoir une nouvelle architecture de surveillance et de diagnostic pour les avions du futur. Les différentes entités impliquées pour fournir un diagnostic de maintenance pour un avion sont présentées. Un cas d'application réelle, le système de génération pneumatique d'un A380, est introduit et permet d'illustrer le besoin d'un raisonnement de pronostic sur de tels systèmes industriels complexes.

Le dernier chapitre dégage les apports de cette thèse et des conclusions, discute les hypothèses formulées pour ce travail et enfin propose diverses perspectives concernant les différentes parties développées.

1 Contexte de la maintenance et état de l'art

Résumé : Le problème de la maintenance des systèmes complexes est défini dans ce chapitre. On montre l'intérêt d'une stratégie de maintenance préventive afin de réduire les coûts de panne et d'indisponibilité du système. Pour optimiser la maintenance préventive, il faut surveiller et analyser l'état du système à l'aide de méthodes de diagnostic et de pronostic. Les concepts relatifs aux problèmes de diagnostic et de pronostic sont introduits et les principales méthodes qui existent pour résoudre ces problèmes sont détaillées. Ces méthodes s'appuient, dans chacun des deux domaines, sur une connaissance plus ou moins approfondie du système à considérer (données historiques, données mesurées en ligne, modèles).

1.1 Introduction

L'efficacité de la maintenance des systèmes industriels est un enjeu économique majeur pour leur exploitation commerciale. La maintenance doit permettre d'améliorer la fiabilité, la sécurité et la qualité des équipements du système industriel pour un moindre coût. Les principales difficultés et sources d'inefficacité résident dans le choix des actions de maintenance. Une action de maintenance consiste à remplacer les équipements en panne qui ne sont plus capables de réaliser leur fonction. Un mauvais choix d'actions peut conduire à une maintenance non satisfaisante et à un surcoût dû à l'immobilisation du système. Optimiser la maintenance consiste à réduire la durée d'immobilisation du système complexe en minimisant la durée des interventions et le nombre d'actions de maintenance.

Une action de maintenance est très difficile à déterminer pour un système distribué. Un système distribué est un système complexe qui résulte d'un assemblage de composants totalement hétérogènes (logiciels, matériels). Il est nécessaire de surveiller de manière graduelle chaque composant pour pouvoir établir une décision d'action de maintenance pour le système global.

A l'aide des nouvelles technologies embarquées, il est possible de mettre en place un système de supervision afin de surveiller les composants du système et de détecter en ligne les problèmes ou les pannes pouvant survenir dans le système. Il est alors nécessaire de fournir un diagnostic de maintenance en ligne qui analyse les différentes sources

d'observation et qui permet d'identifier les équipements en panne à remplacer. Afin d'améliorer la maintenance préventive du système distribué, une fonction de pronostic est intégrée à ce système de supervision permettant de programmer les futures phases de maintenance.

Ce chapitre rappelle toutes les formes de maintenance qui existent pour maintenir un système complexe et montre l'intérêt qu'il est nécessaire de porter à la maintenance préventive. Après avoir défini plusieurs concepts liés au diagnostic, les différentes approches possibles pour fournir un diagnostic de maintenance sont décrites. Le problème de diagnostic a atteint une certaine maturité. De nombreux travaux qui traitent de ce problème peuvent dorénavant servir de références. Le pronostic est, quant à lui, une discipline émergente qui suscite auprès des industriels énormément d'intérêt. Le problème de pronostic n'a jusqu'à maintenant pas réellement été formalisé et encore peu de travaux à ce sujet sont disponibles. Après avoir donné quelques définitions relatives au pronostic, on montre que les méthodes utilisées reposent, comme pour les méthodes de diagnostic, sur les capacités de surveillance et sur une connaissance approfondie de la nature et du fonctionnement du système complexe.

1.2 Maintenance des systèmes complexes

La notion formalisée de maintenance est née dans l'industrie vers la fin des années 1970. Une définition de la maintenance est donnée par la norme AFNOR NFX 13-306 [Afn] :

Définition 1 (Maintenance). *La maintenance est l'ensemble des actions qui permettent de maintenir ou de rétablir un bien dans un état spécifié ou en mesure d'assurer un service déterminé.*

Selon cette définition, la maintenance permet de maintenir ou de rétablir un système dans un état préalablement spécifié afin que le système soit capable de fournir les fonctions pour lesquelles il a été conçu. L'analyse des différentes formes de maintenance repose sur trois concepts.

- **Les événements** qui sont à l'origine de l'action de maintenance : référence à un échéancier, résultat de diagnostic, information de capteur, mesure d'usure, apparition d'une défaillance, etc ...
- **Les méthodes de maintenance** qui leur sont associées : maintenance préventive systématique, maintenance préventive conditionnelle, maintenance corrective.
- **Les opérations de maintenance** : inspection, contrôle, dépannage, réparation, etc ...

Ces opérations de maintenance s'effectuent sur les équipements du système complexe qu'ils soient matériels ou logiciels. Les activités de maintenance, au sens de dépannage

d'un équipement, ont toujours existé. Elles consistent essentiellement à réparer un équipement lorsque celui-ci est déjà en panne. Un équipement en panne n'est plus capable de réaliser les fonctions pour lesquelles il a été conçu, il est dit défaillant.

Définition 2 (Défaillance). *Une défaillance correspond à une cessation de l'aptitude d'une entité à accomplir une ou plusieurs fonctions requises.*

On peut repérer deux formes de défaillance suivant le degré de dégradation du système : la défaillance partielle et la défaillance complète. Leur définition est donnée ci-dessous.

Définition 3 (Défaillance partielle). *Une défaillance partielle correspond à une dégradation de l'aptitude d'un système à accomplir des fonctions requises.*

Dans [ZWI 95], une défaillance partielle résulte de déviations d'une ou plusieurs caractéristiques du système au delà des limites spécifiées, telle qu'elle n'entraîne pas une disparition complète des fonctions requises. Lorsqu'une défaillance entraîne une disparition complète des fonctions du système, il s'agit d'une défaillance complète.

Définition 4 (Défaillance complète). *Une défaillance complète est représentée par une cessation de l'aptitude d'un système à accomplir l'ensemble des fonctions requises.*

Un équipement défaillant nécessite une opération de maintenance pour être réparé. Les activités de dépannage n'intègrent pas d'aspect préventif. Il est bien plus intéressant de prévenir une panne avant qu'elle ne provoque une défaillance d'un équipement du système. La maintenance peut également aider à la reconstitution et à l'amélioration du système. Pour cela elle doit prendre en considération de nombreuses contraintes comme la qualité, la sécurité, l'environnement, le coût, etc...

1.2.1 Méthodes de maintenance

Il existe deux principales familles de maintenance que l'on peut repérer sur la figure 1.1 : la maintenance corrective et la maintenance préventive. La maintenance corrective est celle que le système subit lorsque la panne est déjà présente et qu'il faut la réparer. La maintenance préventive est celle que l'on anticipe pour prévenir les défaillances. Une classification des stratégies de maintenance peut être trouvée dans [KOT 06].

1.2.1.1 La maintenance corrective

La maintenance corrective (ou accidentelle) a pour objectif de rétablir le système après une défaillance (perte de fonction) de manière à ce qu'il soit capable de fournir à

	Maintenance				
Type de maintenance	Maintenance corrective		Maintenance préventive		
	Maintenance palliative	Maintenance curative	Maintenance systématique	Maintenance conditionnelle	Maintenance prévisionnelle
Evénement déclencheur	Défaillance	Défaillance	Date/échéance	Franchissement limite ou seuil	Dérives, Tendances
Action de maintenance	Dépannage	Réparation	Remplacements systématiques	Remplacements Sous condition	Interventions ciblées

FIG. 1.1 – Types de maintenance

nouveau ses fonctions [SHE 94] [EIS 97]. On peut distinguer deux types de maintenance corrective : la maintenance curative et la maintenance palliative.

La maintenance curative

Ce type de maintenance permet de remettre définitivement en état le système après l'apparition d'une défaillance. Cette remise en état du système est une réparation durable. Les équipements réparés doivent assurer les fonctions pour lesquelles ils ont été conçus. Une réparation est une opération définitive de la maintenance curative qui peut être décidée soit immédiatement à la suite d'une défaillance, soit après un dépannage (voir dans le paragraphe suivant). Elle provoque donc une indisponibilité du système.

La maintenance palliative

La maintenance palliative revêt un caractère temporaire, provisoire. Elle est principalement constituée d'opérations qui devront toutefois être suivies d'opérations curatives (réparations). Le dépannage est une opération de maintenance palliative qui est destinée à remettre le système en état provisoire de fonctionnement de manière à ce qu'il puisse assurer une partie des fonctions requises. Les opérations de dépannage sont souvent de courte durée et peuvent être nombreuses. Parce qu'elles ont lieu souvent, elles sont également très coûteuses.

1.2.1.2 La maintenance préventive

Ce type de maintenance effectuée selon des critères prédéterminés, a pour objectif de prévenir les pannes (défaillances) lors du fonctionnement du système [MOB 90]. L'aspect préventif est important pour des raisons de sûreté de fonctionnement mais aussi pour des raisons économiques et parfois pratiques (l'équipement n'est disponible pour la maintenance qu'à certains moments précis). La maintenance préventive a pour

but de supprimer les causes d'accidents graves en réduisant la probabilité de défaillance d'un système ou la dégradation des équipements du système. Elle vise à réduire les coûts des pannes et de la maintenance corrective en minimisant ou en évitant des réparations et indisponibilités coûteuses par un entretien constant et préventif. Il existe trois formes particulières de maintenance préventive : la maintenance préventive systématique, la maintenance préventive conditionnelle et la maintenance préventive prévisionnelle.

La maintenance préventive systématique

La maintenance systématique est élaborée par rapport à un échéancier établi selon le temps de fonctionnement ou le nombre d'unités d'usage [JAR 87]. Même si le temps est l'unité d'usage la plus répandue, d'autres unités peuvent être retenues telles que : le nombre de vols pour un avion, la distance parcourue, le nombre de cycles effectués, etc ... Elle consiste à remplacer systématiquement un certain nombre d'équipements préalablement définis même si aucune panne n'est apparue. Il s'agit donc d'une maintenance programmée. La périodicité des opérations de maintenance est déterminée à partir de la mise en service et est essentiellement basée sur des données de fiabilité. Cette forme de maintenance nécessite de connaître le comportement du matériel, les modes de dégradation (l'usure des équipements) et le temps moyen de bon fonctionnement entre deux défaillances du système (MTBF).

La maintenance préventive systématique assure le remplacement périodique des équipements dont certaines pièces sont anormalement usées. Elle permet également de remplacer les équipements dont la panne risque de provoquer des accidents graves ou les équipements ayant un coût de défaillance élevé. Cette méthode systématique coûte assez cher mais elle assure une grande sécurité en fixant une périodicité de visite qui diminue le risque d'avoir une défaillance avant l'intervention.

La maintenance préventive conditionnelle

La maintenance préventive conditionnelle, également appelée maintenance prédictive, est subordonnée à un type d'événement prédéterminé (résultat de diagnostic, donnée d'un capteur, mesure d'usure, etc...) révélateur de l'état de fonctionnement du système. Elle dépend de l'expérience mais fait également intervenir des données recueillies en temps réel qu'elle analyse de manière à déterminer ou prédire une défaillance [DEL 09]. Une connaissance profonde des équipements du système permet de pouvoir prédire les pannes (défaillances) en observant un certain nombre de paramètres précurseurs de panne, comme par exemple :

- l'usure, visible notamment par des poussières, des débris,
- l'oxydation de pièces,
- les connexions électriques, mécaniques ou hydrauliques relâchées,
- les vibrations anormales, inhabituelles,

- les fuites de fluides, d'air comprimé,
- les échauffements inhabituels,
- les résultats dégradés : dérives des spécifications des pièces, besoins de réglages fréquents, ...

Les mesures et les paramètres suivis sont soigneusement déterminés pour être représentatifs de l'état de fonctionnement du système. Quelle que soit la technique utilisée, les données recueillies ou mesurées sont toujours comparées à une référence. Le franchissement d'un seuil prédéterminé déclenche un événement (alarme) qui permet de décider d'une opération de maintenance sur le système avant que la dégradation n'entraîne une défaillance. Ce type de maintenance peut être appliqué dans le cas où le MTBF du système n'est pas connu. Si le MTBF est connu, il permet d'adapter en ligne le temps restant jusqu'à la prochaine visite de maintenance. Ce temps dépend directement de la surveillance des paramètres précurseurs de panne.

La maintenance préventive prévisionnelle

La maintenance prévisionnelle, également appelée maintenance proactive, est également réalisée à la suite d'une analyse de l'évolution surveillée des paramètres précurseurs de panne qui permettent de qualifier l'état de fonctionnement du système. La maintenance proactive est une forme de maintenance prédictive qui consiste à déterminer les causes à l'origine des défaillances et des usures précoces des équipements du système. La maintenance prévisionnelle permet d'anticiper et de prévoir au mieux le moment où l'opération de maintenance devra être réalisée.

Cette forme de maintenance permet de réduire le nombre de défaillances imprévues, et donc l'indisponibilité du système. Elle permet de planifier les opérations de maintenance de manière à utiliser les équipements au maximum de leurs possibilités. En surveillant les équipements, il est possible de corriger des erreurs de conduite ou des anomalies qui peuvent générer des défaillances plus graves par la suite et d'améliorer la sécurité en évitant des accidents critiques. Par contre, cette forme de maintenance nécessite de mettre en place des techniques de surveillance et de mesure qui peuvent être très coûteuses.

Les opérations de maintenance préventive

Les opérations de surveillance (inspections, visites, contrôles) sont nécessaires pour maîtriser l'évolution de l'état de fonctionnement du système. Elles sont effectuées de manière continue ou à des intervalles prédéterminés calculés sur le temps de fonctionnement ou le nombre d'unités d'usage.

- Les inspections consistent à relever périodiquement des anomalies et exécuter des réglages simples qui ne nécessitent pas d'arrêt de fonctionnement du système.
- Les visites sont des opérations de surveillance qui, dans le cadre de la maintenance

préventive systématique, s'effectuent selon une périodicité déterminée. Elles correspondent à une liste d'opérations définies qui peuvent entraîner une indisponibilité du système.
- Les contrôles sont des vérifications de conformité par rapport à des données préétablies suivies d'une décision.

La révision correspond à l'ensemble des opérations de maintenance préventive effectuées pour éviter toute défaillance majeure ou critique du système pendant un temps ou pour un nombre d'usages déterminé.

1.2.2 Optimisation de la maintenance

Lors du choix de la méthode de maintenance, il faut arbitrer entre les performances attendues du système et les coûts que l'on est prêt à assumer. Par exemple, les économies sur les coûts de maintenance finissent généralement par coûter cher en arrêts de fonctionnement du système. Inversement, au-delà d'un certain seuil, un niveau de maintenance trop important coûte cher sans forcément apporter de supplément de performance au système. Il faut donc arbitrer entre le niveau de disponibilité des équipements que l'on souhaite garantir et le niveau de coûts directs de maintenance acceptable (personnel, matériels) comme le montre la figure 1.2 qui provient de l'ouvrage [HOH 09]. Le niveau minimum des dépenses réelles correspond à un seuil incompressible dû aux pannes imprévisibles qu'il va bien falloir réparer, quels que soient les coûts que cela engendre, sous peine d'arrêt du système.

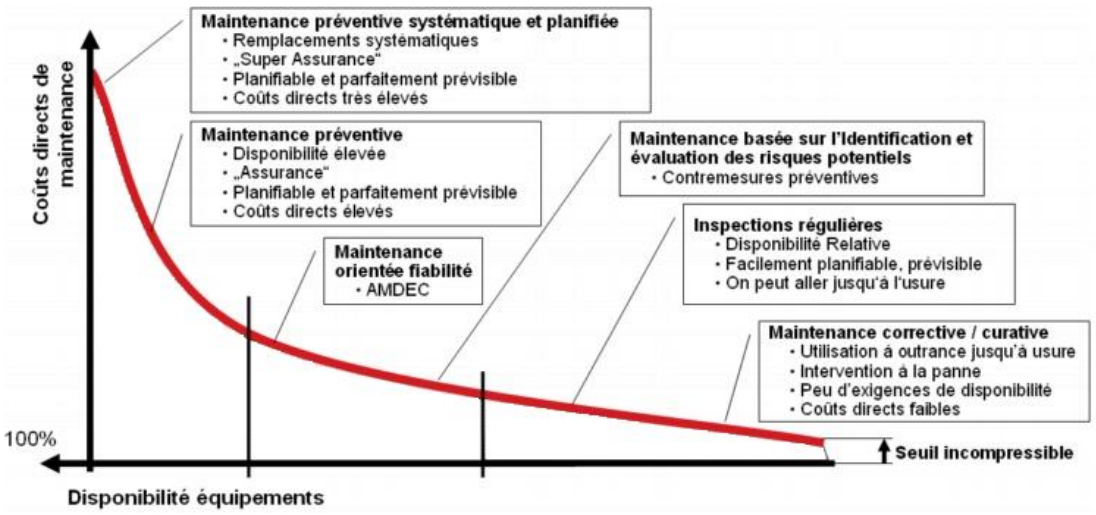

FIG. 1.2 – Disponibilité et coûts de maintenance

Optimiser la maintenance consiste à garantir un certain niveau de disponibilité des équipements du système pour un moindre coût. Afin d'éviter les coûts dus à l'arrêt du

système pour le réparer, il est préférable d'anticiper les pannes potentielles du système et de les éviter. La maintenance prévisionnelle (ou proactive) permet d'identifier les causes des futures défaillances d'un système en tenant compte de l'état réel de dégradation et d'augmenter la durée de vie du système contrairement à :

- réparer un système à chaque panne,
- considérer que les pannes sont inévitables et donc normales,
- mettre en place une maintenance systématique.

Des méthodes de diagnostic sont nécessaires de manière à détecter les signes précurseurs d'une défaillance potentielle dans le système et à identifier les causes des pannes en terme d'équipement à réparer. Des méthodes de pronostic peuvent ensuite s'appuyer sur ces informations pour évaluer et prédire les effets de ces pannes sur les autres équipements du système [JAR 06]. Les méthodes de pronostic permettent également d'améliorer la planification et l'ordonnancement des actions de maintenance.

1.3 Diagnostic pour la maintenance

1.3.1 Concepts généraux

Le fonctionnement d'un système est dit normal lorsque le système est capable de fournir les fonctions pour lesquelles il a été conçu. L'apparition d'une faute sur un équipement du système peut générer une défaillance, c'est-à-dire une perte de fonction du système, qui n'est alors plus dans son fonctionnement normal. Les concepts de faute et de défaillance sont définis dans [VIL 88] et dans [ISE 97].

Définition 5 (Faute). *Une faute représente une déviation non acceptable d'au moins une propriété caractéristique ou d'un paramètre du système.*

Une faute peut générer une défaillance du système. Lorsqu'il y a une défaillance du système, il n'est plus en mesure de remplir les fonctions requises. On rappelle qu'une défaillance correspond à une cessation de l'aptitude d'une entité à accomplir une ou plusieurs fonctions requises.

Après une défaillance, on considère que le système est en panne. Dans [VIL 88], une panne est définie par l'inaptitude d'une entité à accomplir une fonction requise. Une panne est la conséquence d'une faute qui est donc toujours associée à une défaillance.

Afin de déterminer si un système remplit correctement ses objectifs, il est nécessaire de surveiller de manière précise son fonctionnement à l'aide de capteurs positionnés stratégiquement. Des techniques de placement ou de sélection de capteurs sont alors utilisées pour déterminer le nombre et la localisation des capteurs à placer sur le système [NAR 98] [SPA 04] [TOR 07]. La fonction de surveillance permet de détecter le

passage du système en fonctionnement anormal. Elle récupère les informations issues des capteurs et les transforme en indicateurs de défaillance à partir d'une référence illustrant le fonctionnement normal ou anormal du système.

Définition 6 (Indicateur). *Un indicateur de défaillance est une quantité significative et pertinente à partir de laquelle il est possible de détecter une défaillance.*

Lorsque ces indicateurs de défaillance révèlent un fonctionnement (comportement) anormal, ils deviennent des symptômes. Les symptômes traduisent les effets observables des défaillances.

Définition 7 (Symptôme). *Un symptôme est l'effet ou la conséquence visible d'une défaillance.*

Le problème du diagnostic est de déterminer pour un système donné, à partir d'une référence et d'un ensemble d'indicateurs fournis par la fonction de surveillance, les fautes étant apparues sur le système. Dans [ZWI 95], le diagnostic est défini de la manière suivante.

Définition 8 (Diagnostic). *Le diagnostic est l'identification de la cause probable de la (ou des) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test.*

Cette définition résume les deux tâches essentielles du diagnostic : l'observation des symptômes de la défaillance et l'identification de la cause de la défaillance à l'aide d'un raisonnement logique fondé sur des observations du système. Les deux étapes d'une méthode de diagnostic sont donc la localisation et l'identification des fautes sur les équipements responsables d'une ou plusieurs défaillances du système. L'étape de localisation permet d'isoler les équipements en panne, c'est-à-dire dans lesquels une faute est apparue. L'étape d'identification détermine le type de faute apparue. Une fois le type de faute identifié et selon la connaissance disponible sur le système, il est parfois possible de propager les effets d'une faute sur les équipements du système afin de prédire les conséquences de ces défaillances.

1.3.2 Méthodes de diagnostic

Plusieurs techniques permettent de diagnostiquer les fautes survenant dans un système qui provoquent des défaillances. Ces techniques reposent surtout sur la connaissance disponible sur le système. Cette connaissance dépend des techniques de surveillance du système et d'une référence illustrant le fonctionnement normal (comportement nominal) ou le fonctionnement anormal (comportement en présence de faute) du système. Cette référence est représentée soit par un historique, une expérience, soit par un modèle connu ou estimé du comportement du système. Une classification des

méthodes de diagnostic est proposée figure 1.3 .

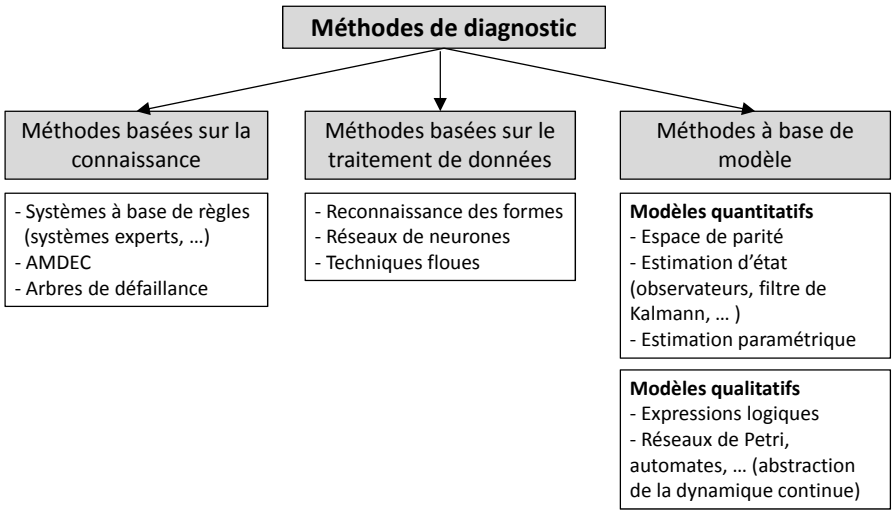

FIG. 1.3 – Classification des méthodes de diagnostic

1.3.2.1 Approche basée sur la connaissance

Ce type d'approche utilise une connaissance explicite de relations causales entre les symptômes, les défaillances et les fautes. Cette approche associe directement un symptôme à la faute qui en est la cause. La connaissance n'est pas extraite d'un modèle explicite structurel ou de comportement du système. Elle est souvent acquise durant la phase de conception et provient d'une analyse fonctionnelle et structurelle du système.

Cette source de connaissance peut résulter d'une analyse des modes de défaillance et de leurs effets (AMDE) ou bien d'un historique des mauvais fonctionnements du système représenté par un arbre de défaillances par exemple. L'analyse des modes de défaillance et de leurs effets et les arbres de défaillances sont des techniques issues du domaine de la sûreté de fonctionnement et d'études de risque dans les systèmes industriels. Elles sont utilisées pour identifier les causes des défaillances possibles d'un système. Cette connaissance, qui se présente sous la forme d'associations entre effets et causes, est dite externe ou de surface. Des approches classiques de diagnostic qui utilisent ce type de connaissance externe sont des systèmes à bases de règles comme par exemple les systèmes experts [BUC 84] [JAC 98] [VEN 03].

Analyse des Modes de Défaillance et de leurs Effets

L'analyse des modes de défaillance et de leurs effets est une méthode très largement utilisée dans de nombreux domaines industriels. C'est une méthode inductive qui permet une analyse systématique et très complète, composant par composant, de tous les modes

de défaillance possibles des composants et qui précise leurs causes et leurs effets sur le système global [VIL 88]. Une méthode inductive consiste, pour un système et une défaillance donnée, à étudier de façon détaillée les effets (ou les conséquences) de cette défaillance sur le système lui-même et/ou son environnement. La base de connaissance de cette méthode est définie à partir d'une analyse structurelle et fonctionnelle du système, c'est-à-dire d'une analyse de ses fonctions et de ses composants. L'ensemble des modes de défaillance possibles des composants doit être établi et pour chaque mode de défaillance, sont recherchées les causes possibles de son apparition. Finalement, une étude des effets sur le système est faite pour chaque combinaison (cause, mode de défaillance) [ZWI 95]. Les résultats sont présentés sous forme de tableaux comme sur la figure 1.4.

Identification du composant	Fonctions Etats	Modes de défaillance	Cause possibles	Effets	Criticité	Moyens de détection	Parades de l'opérateur humain	Observations

FIG. 1.4 – AMDEC

L'analyse des modes de défaillance, de leurs effets et de leur criticité (AMDEC) est une extension de l'AMDE qui inclut une analyse de criticité des modes de défaillance. La criticité permet d'extraire les modes de défaillance les plus critiques.

Les tableaux d'AMDE(C) sont obtenus à partir d'une analyse déductive, à partir des causes (des défaillances), on détermine les conséquences qu'elles peuvent avoir sur le système (les effets). L'utilisation des tableaux d'AMDE(C) à des fins de diagnostic conduit à utiliser une procédure abductive. Supposant qu'un système est défaillant, une démarche abductive consiste à rechercher les causes pouvant expliquer les effets observés de cette défaillance. Les tableaux de l'AMDE(C) sont utilisés comme un outil d'identification des causes de défaillances à partir des effets observés. Les inconvénients d'une telle méthode sont qu'elle nécessite une longue expérience et que toute modification entraîne une ré-écriture du tableau.

Arbre de défaillance

Les arbres de défaillance sont couramment utilisés dans les analyses de fiabilité ou de sécurité des systèmes [VIL 88]. Ils représentent des outils très puissants pour identifier les causes premières conduisant à une défaillance indésirable. Ces arbres sont créés à l'aide d'une procédure déductive optimisée qui détermine des chemins critiques dans un système. Les chemins critiques dans un système correspondent aux diverses combinaisons possibles d'événements qui entraînent la réalisation d'un événement indésirable unique (événement de faute entraînant une panne du système).

La procédure qui utilise les arbres de défaillance à des fins de diagnostic est abductive, elle se focalise d'abord sur les événements indésirables pour identifier ensuite leurs causes. Un arbre de défaillance est établi sous la forme d'un diagramme logique

et comporte au sommet l'événement indésirable. Les causes immédiates qui produisent cet événement sont ensuite hiérarchisées à l'aide de symboles logiques "ET" et "OU".

Pour exécuter un diagnostic correct à partir des arbres de défaillances, ceux-ci doivent largement représenter toutes les relations causales du système, capable d'expliquer tous les scénarios de fautes possibles. Cette méthode déductive s'utilise difficilement pour les systèmes qui dépendent du temps.

Systèmes experts

La technique des systèmes experts est la plus répandue pour la supervision des systèmes complexes. Les systèmes experts sont des outils de l'intelligence artificielle, utilisés lorsqu'aucune méthode algorithmique exacte n'est disponible ou possible. La propriété principale de ces systèmes est de pouvoir représenter et restituer les connaissances acquises par un expert. Dans [ZWI 95], un système expert est définit comme étant "un système informatique destiné à résoudre un problème précis à partir d'une analyse et d'une représentation des connaissances et du raisonnement d'un spécialiste à ce problème." Dans la plupart des cas, les connaissances utilisées pour le développement d'un système expert d'aide au diagnostic, reposent sur l'apprentissage des relations entre les causes et les effets observés pour chaque défaillance du système [AGU 99]. Un système expert est composé de deux parties indépendantes.

- Une base de connaissance qui est elle-même composée d'une base de faits qui contient les informations, les données concernant le cas traité et d'une base de règles connues qui modélisent la connaissance du domaine considéré.
- Un moteur d'inférence capable de raisonner à partir des informations contenues dans la base de connaissance et de faire des déductions.

Le moteur d'inférence utilise les données et les règles pour produire de nouvelles données. Le rôle d'un système expert est donc d'inférer des règles du type :

SI [A="vrai"] ET [A *implique* B] ALORS [B="vrai"].

Au fur et à mesure que les règles sont appliquées, des nouveaux faits se déduisent et se rajoutent à la base de faits. Le diagnostic par systèmes experts se fonde sur l'expérience disponible sur le système pour construire une table de correspondance qui associe les observations aux diagnostics correspondants [BUC 84].

Ces techniques sont efficaces au niveau du temps de calcul, elles sont peu coûteuses et leur implémentation est simple. L'expérience dont dépendent ces approches est cependant difficile à acquérir et si le système évolue, les règles sont à remettre en cause. Une nouvelle expertise est alors nécessaire.

1.3.2.2 Approche basée sur le traitement de données

Dans ces méthodes, les seules informations disponibles sont les signaux issus des capteurs positionnés sur le système. Les capteurs sont supposés fiables et leurs valeurs correctes. L'objectif de ces méthodes est d'associer un ensemble de mesures à des états de fonctionnement connus du système. Ces approches à base de données font appel à des méthodes de reconnaissance de formes qui utilisent des techniques d'apprentissage numérique et de classification afin d'établir un modèle de référence du système fondé sur l'expérience (exploitation des données, des mesures sous la forme d'historique). Le modèle établi ne provient donc pas d'une spécification du système durant la phase de conception [FOU 03]. Il ne repose pas sur une connaissance physique du système. Ce modèle de référence capture le comportement normal du système et est utilisé pour la détection et le diagnostic.

Les principales techniques de classification utilisées pour construire un tel modèle sont les réseaux de neurones et la logique floue. Le principe des méthodes de reconnaissance de formes est décrit dans les ouvrages [DUB 90] et [ZWI 95]. Il est illustré par la figure 1.5.

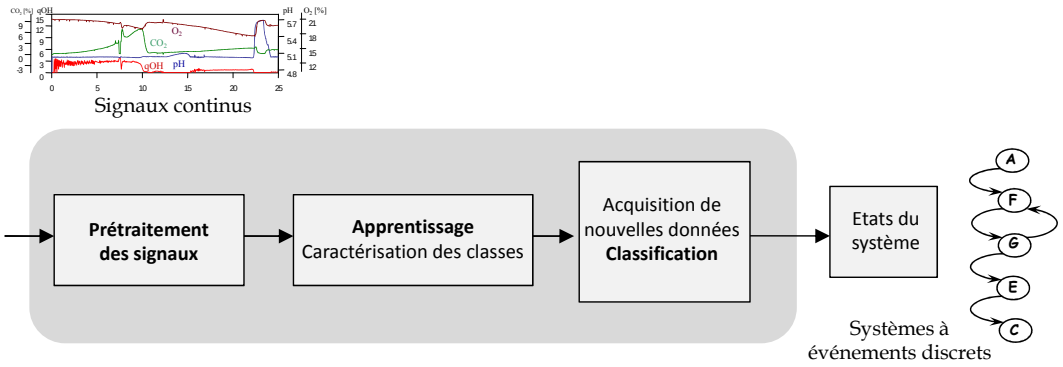

FIG. 1.5 – Diagnostic par reconnaissance de formes

Un prétraitement des signaux issus des capteurs permet d'établir les variables les plus pertinentes pour déterminer les états de fonctionnement du système et d'éliminer les bruits possibles à l'aide de techniques de filtrage.

Les méthodes de reconnaissance de formes consistent à reconnaître des formes parmi différentes possibilités à partir d'observations bruitées dans le but d'identifier les états du système (état de fonctionnement normal, état de défaillance, ...). Une forme est un ensemble de paramètres (ou de caractéristiques) associés à une donnée qui peuvent être numériques ou symboliques. Un prototype est défini par des valeurs précises de l'ensemble des paramètres (ou caractéristiques) d'une situation particulière (défaillance). Une classe est caractérisée par un ensemble possible de valeurs des paramètres. Elle est représentée par son prototype qui définit le mode de fonctionnement du système.

L'objectif d'une méthode de classification est d'identifier à partir de mesures du

système les différentes classes en regroupant les données qui ont des valeurs de paramètres (caractéristiques) similaires. Ces classes contiennent l'information qui caractérise les états et les défaillances du système et sont représentées par leur prototype :

$$\begin{array}{l} \text{Classe } x_1 \rightarrow \text{Prototype } P_1 = \text{ "Fonctionnement normal",} \\ \text{Classe } x_2 \rightarrow \text{Prototype } P_2 = \text{ "Mode de faute 1",} \\ \ldots \\ \text{Classe } x_n \rightarrow \text{Prototype } P_n = \text{ "Mode de faute n".} \end{array} \tag{1.1}$$

Une fois les classes identifiées, la méthode de classification peut évaluer la distance entre une forme particulière et son prototype.

Des techniques d'apprentissage permettent d'établir les valeurs caractéristiques des paramètres de chaque classe et d'évaluer le prototype associé aux nouvelles observations dans le but de déterminer l'état de fonctionnement du système. Elles réalisent donc un partitionnement de l'espace de représentation des données en déterminant les frontières entre les classes. Avec une technique d'apprentissage supervisé, il est possible de créer en ligne de nouvelles classes afin d'identifier des situations non prévues.

L'objectif d'une méthode de reconnaissance de formes est d'associer toute nouvelle donnée à une classe déterminée par la méthode d'apprentissage [DUB 01] [VEN 03] [KEM 04]. Il existe principalement deux types de techniques pour la reconnaissance de formes : la reconnaissance de formes par réseaux de neurones et la reconnaissance de formes par la logique floue.

Techniques neuronales

Les réseaux de neurones sont utilisés pour la classification des données et des formes. Un réseau de neurones est un modèle de calcul dont la conception est inspirée du fonctionnement des vrais neurones. Les réseaux de neurones sont optimisés par des méthodes d'apprentissage par expérience [HER 06] [FEL 07]. Les poids des neurones sont ajustés lorsqu'on leur présente de nouvelles données (observations) à traiter.

Techniques floues

Une autre technique permettant la classification des données utilise la logique floue [NAR 07]. La logique floue introduite par [ZAD 65] est issue de la théorie mathématique des ensembles flous qui considère des ensembles définis de manière graduelle. Le concept de fonction d'appartenance permet de modéliser la définition d'un sous-ensemble [TAK 85]. A l'inverse de la logique booléenne, la logique floue associe à une donnée un degré d'appartenance à un ensemble qui peut être différent d'un état booléen 0 ou 1.

Le modèle de référence utilisé par ces méthodes de classification est le résultat d'un processus d'apprentissage. Aucune garantie ne peut être donnée quant à la complétude, la cohérence et la précision du modèle. De plus, la phase d'apprentissage nécessite qu'un grand nombre de données soient disponibles (données collectées à partir des mesures réelles).

1.3.2.3 Approche à base de modèles

Les approches de diagnostic à base de modèles reposent sur une connaissance physique profonde du système à diagnostiquer. Le système est représenté sous forme d'un ou plusieurs modèles qui décrivent la structure du système et son comportement nominal ou encore son comportement en présence de faute. La méthode de diagnostic s'appuie sur la comparaison du comportement réel observé sur le système physique avec le comportement prédit à l'aide de modèles. La détection d'incohérences permet de conclure sur l'occurrence de faute dans le système. Un modèle de dysfonctionnement (modèle de faute) permet de localiser les fautes et éventuellement de les identifier.

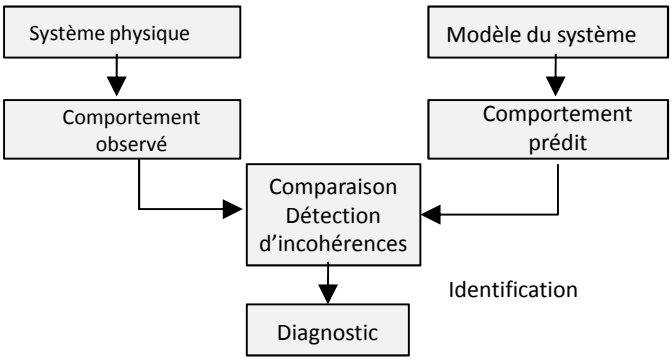

Fig. 1.6 – Diagnostic à base de modèles

Deux principales approches peuvent se distinguer dans les méthodes de diagnostic à base de modèles. L'approche FDI (Fault Detection and Isolation) issue de la communauté automatique utilise des modèles quantitatifs pour décrire le modèle de comportement du système. Des équations algébro-différentielles permettent de représenter le comportement continu du système avec une certaine précision numérique. L'approche DX, fondée sur une théorie logique du diagnostic, provient de la communauté de l'intelligence artificielle. Elles utilisent des modèles qualitatifs qui permettent de représenter de manière efficace les interactions entre composants ou systèmes.

Les approches FDI

Les approches FDI reposent sur une connaissance approfondie du fonctionnement du système. Elles utilisent des modèles de référence quantitatifs qui peuvent être ob-

tenus à partir de lois fondamentales de la physique [FRA 96]. Ces modèles analytiques numériques décrivent le comportement normal du système par un ensemble d'équations différentielles [GER 98].

Le diagnostic consiste à générer des indicateurs de présence de faute dans le système que l'on appelle des *résidus*. Pour obtenir ces résidus, le modèle de référence est restreint aux entrées et sorties observables (mesurées par capteurs sur le système réel). Une valeur non nulle d'un résidu est interprétée comme une modification anormale (ou une déviation inacceptable) d'une propriété ou d'un paramètre caractéristique du système modélisé. Deux classes de méthodes ont été développées dans l'approche FDI.

- Les méthodes basées sur les résidus utilisent le modèle pour prédire les valeurs qui doivent être mesurées. La méthode de l'espace de parité consiste à éliminer les variables inconnues et la méthode d'estimation d'état par observateurs consiste à estimer les variables inconnues.
- Les méthodes d'estimation de paramètres utilisent des techniques d'estimation pour calculer les valeurs d'un paramètre du modèle dont la structure et les paramètres nominaux sont parfaitement connus.

Le modèle Les équations différentielles du modèle qui décrivent le comportement du système peuvent être linéaires ou non linéaires [FRA 87] [STA 89], à temps discret ou à temps continu. Le système modélisé peut également être un système hybride [NAR 03][BAY 08][COQ 05]. Les fautes pouvant surgir dans le système sont représentées dans le modèle par des termes additifs ou multiplicatifs. Une faute additive est représentée par une entrée supplémentaire dans le modèle. Elle correspond à une modification des sorties indépendante des entrées connues du modèle. Une faute multiplicative est représentée par un coefficient qui affecte directement les paramètres du système modélisé. Ce coefficient explique un changement de paramètres qui cause l'évolution des sorties.

Un exemple de modèle linéaire à temps invariant avec des fautes additives est donné ci dessous :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + E_d d(t) + E_f f(t) \\ y(t) = Cx(t) + Du(t) + F_d d(t) + F_f f(t), \end{cases} \tag{1.2}$$

dans lequel

- x représente le vecteur d'état qui décrit l'état du système,
- y est le vecteur de sortie dont la valeur est connue à partir des capteurs positionnés sur le système,
- u correspond au vecteur d'entrée connu et décrit les entrées qui sont mesurées,
- d représente le vecteur d'entrée inconnu et décrit les autres entrées du système comme les bruits, les perturbations, etc ...
- f est le vecteur de fautes pouvant affecter le comportement du système.

Notion de résidu Un résidu correspond à une différence entre le comportement prédit par le modèle de référence et le comportement observé du système. Il peut se définir de la manière suivante [GER 98].

Définition 9 (Résidu). *Un résidu est associé à une différence résultant de la comparaison de mesures de capteurs à des valeurs calculées analytiquement de la variable considérée dans le système modélisé.*

Un résidu correspond à une expression analytique dans laquelle seulement des valeurs mesurables (observables) du modèle du système apparaissent. Prenons l'exemple d'une régulation dans laquelle les valeurs de la vitesse v et de la position x sont mesurables : l'expression $dx/dt - v = 0$ est un résidu. Dans le cas nominal, la valeur du résidu sera considérée comme nulle. Lorsqu'un résidu est non nul, cela révèle la présence d'une faute pouvant provoquer une défaillance dans le système.

A cause des erreurs de modélisation et de la présence de bruit, les résidus ne sont jamais réellement nuls même s'il n'y a pas de faute dans le système. La décision d'une détection nécessite d'évaluer les expressions des résidus obtenus en utilisant les mesures du système afin de déterminer si la différence à laquelle ils sont associés est significative. Pour l'évaluation des résidus, des techniques de reconnaissance de formes, de logique floue, de seuillage sont utilisées ou bien encore des techniques issues de la théorie de décision statistique [BAS 93]. Le principe d'une approche FDI basée sur le calcul des résidus est illustré sur la figure 1.7.

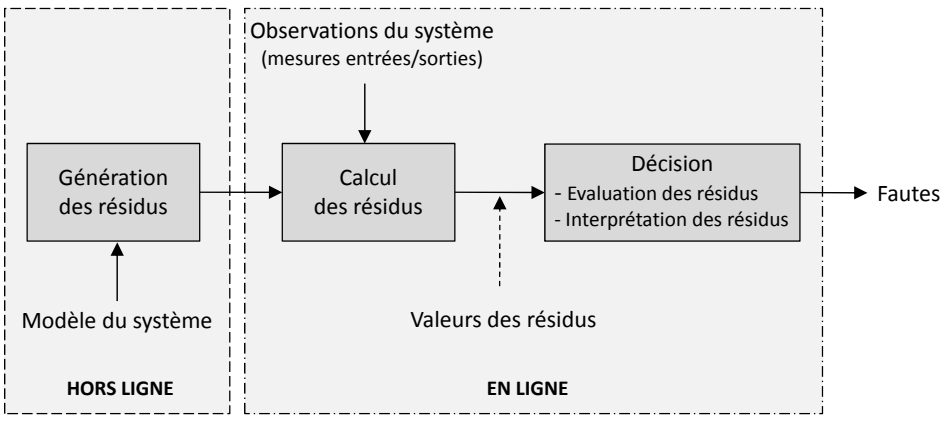

Fig. 1.7 – Approche FDI à base de résidus

Pour obtenir les expressions analytiques des résidus, plusieurs techniques peuvent être utilisées : des relations de parité, des observateurs, des filtres de détection. D'autres méthodes s'intéressent plutôt à l'estimation de paramètres pour la détection de faute. Plusieurs articles de synthèse comme [ISE 97] ou [DUB 01] décrivent l'ensemble de ces méthodes utilisées dans les approches FDI.

Espace de parité Les approches de l'espace de parité reposent sur la génération de résidus [STA 01]. Le comportement normal (nominal) d'un système est caractérisé par les valeurs nulles des résidus. Les mesures sont alors cohérentes avec les propriétés de l'espace de parité. La conception de l'espace de parité est basée sur le développement des expressions analytiques.

L'approche consiste à déterminer des redondances analytiques parmi les entrées et les sorties du système [PAT 91] [MAQ 97]. Les relations de parité utilisent la redondance directe au moyen de relations algébriques liant les différentes grandeurs du modèle du système. Les relations de redondance analytique sont obtenues en éliminant les variables d'état non observables du modèle sur l'espace de parité. On obtient un modèle entrée/sortie du système dans lequel le vecteur d'état x inconnu n'apparaît pas [GER 98] :

$$y(t) = M(\phi)u(t) + S_F(\phi)f(t) + S_D(\phi)d(t), \tag{1.3}$$

dans lequel $M(\phi)$, $S_F(\phi)$ et $S_D(\phi)$ représentent des fonctions de transfert. On tire de ce modèle des relations de redondance analytiques de la forme suivante :

$$r(t) = W(\phi)[y(t) - M(\phi)u(t)]. \tag{1.4}$$

Les équations obtenues n'impliquent donc que des variables entrées/sorties mesurables, donc connues. Les relations de redondance analytiques sont ensuite directement utilisées pour construire des indicateurs de faute pour le diagnostic.

L'utilisation de l'espace de parité pour la détection provient de relations analytiques statiques [DES 84]. Les concepts ont été généralisés ensuite par [MIR 80] et [CHO 84] pour utiliser les redondances dans le cas des systèmes dynamiques. [STA 91] et [DES 81] sont des références d'applications industrielles avec des relations statiques. Des applications industrielles sur des systèmes dynamiques sont développées dans [SCH 03] dans le domaine de l'automobile. Quand certaines sorties ne sont pas directement mesurables, des techniques d'observateurs complètent la méthode de l'espace de parité.

Estimation d'état : Observateurs Les observateurs sont des systèmes dynamiques qui peuvent être utilisés pour la détection et l'isolation de faute dans des systèmes linéaires ou non linéaires. La méthode des observateurs consiste à reconstruire à partir d'un modèle analytique et d'un ensemble d'observations partielles du système (entrées/sorties) les sorties non mesurables du système par une estimation de l'état du système. Les observateurs permettent d'estimer les valeurs des variables d'état. A partir de ces valeurs reconstruites et du modèle dynamique, les valeurs des sorties sont calculées. Prenons l'exemple d'un modèle linéaire à temps discret sous forme de représentation d'état :

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)) \\ \hat{y}(k) &= C\hat{x}(k) + Du(k). \end{aligned} \tag{1.5}$$

La prédiction $\hat{x}$ des composantes du vecteur d'état x du système est corrigée à l'aide d'un terme de correction qui est fonction de l'erreur de sortie $y - \hat{y}$ et d'un gain L.

La présence d'un résidu est évaluée en comparant les variables réelles et les variables estimées. L'erreur d'estimation est définie comme la différence entre l'état réel du système et l'état estimé. Un vecteur de résidus est obtenu représentant la différence entre les mesures des sorties et les valeurs des sorties estimées.

Plusieurs travaux traitent de la détection de faute à l'aide d'observateurs dans le cas des systèmes linéaires [STA 91] [MAG 94]. [FRA 87] et [HAM 99] proposent une solution pour observer des systèmes non linéaires.

Estimation paramétrique L'estimation paramétrique permet d'analyser l'influence des fautes sur les paramètres structuraux du modèle du système. C'est donc une approche naturelle qui permet de détecter et d'isoler les fautes multiplicatives sur les paramètres du modèle dynamique du système. L'estimation paramétrique utilise des méthodes analytiques pour calculer les valeurs des paramètres du modèle dont la structure est connue. Ces paramètres structuraux sont généralement constants.

Pour estimer ces paramètres, des techniques de filtrage sont utilisées. Par exemple, le filtre de Kalman estime de manière récursive les paramètres structuraux d'un modèle linéaire sous forme de représentation d'état à partir de mesures bruitées. Pour les systèmes non linéaires mais linéarisables localement, un filtre de Kalman étendu peut être utilisé et pour les systèmes non linéaires et non linéarisable, un filtrage particulaire est appliqué (simulation et technique statistiques).

La détection d'une faute se fait en comparant les paramètres estimés avec les paramètres nominaux qui caractérisent le comportement normal du système. L'idée principale d'une telle méthode consiste à minimiser la distance entre les valeurs réelles du paramètre et les valeurs estimées. La méthode d'estimation est donc caractérisée par la définition de distance (maximum de vraisemblance, moindres carrés, ...).

Beaucoup de travaux ont été développés dans ce domaine pour des systèmes linéaires [ISE 84] [ISE 93b][ISE 93a] et pour des systèmes non linéaires [BAL 88]. Dans [GER 95], les approches d'estimation paramétrique sont comparées à la méthode d'espace de parité.

Les techniques présentées ici sont les plus connues pour la génération de résidus. Des équivalences entre les techniques utilisant l'espace de parité, les observateurs et l'estimation paramétrique ont été établies dans [PAT 91].

Localisation et identification des fautes Pour localiser les fautes, différents types de résidus sont générés : les *résidus structurés* (propriétés booléennes) et les *résidus directionnels* (propriétés géométriques).

Les résidus structurés sont conçus pour que chaque résidu soit sensible à un sous-ensemble de fautes connues et insensible aux autres fautes. Lorsqu'une faute connue apparaît, la valeur de certains résidus est nulle ou proche de zéro tandis que d'autres résidus seront différents de zéro. L'ensemble des valeurs des différents résidus représente la signature de la faute. L'ensemble des signatures pour les différentes fautes connues pouvant apparaître dans le système est appelé la matrice de signatures :

	f_1	f_2	$\dots$	f_f
r_1	0	1	$\dots$	0
r_2	1	1	$\dots$	1
$\vdots$				
r_r	0	0	$\dots$	1

(1.6)

Dans l'exemple de matrice de signatures donné ci-dessus, le résidu associée à la relation de redondance analytique r_1 n'est pas affecté par la faute f_1, il est par contre affecté par la faute f_2. Le résidu associé à r_2 est quant à lui affecté par les deux fautes f_1 et f_2.

Les deuxièmes types de résidus permettant la localisation des fautes sont les résidus directionnels. Ils sont représentés sous la forme d'un vecteur de résidus orienté selon une direction particulière de l'espace des résidus selon le type de faute apparu. La connaissance sur les différentes fautes ou les modes de défaillances d'un système représenté par un modèle de dysfonctionnement permet l'identification des fautes.

L'approche DX

L'approche DX est une approche qualitative basée sur la cohérence qui provient du domaine de l'intelligence artificielle. La technique du diagnostic de cohérence consiste à comparer le comportement réel du système observé et son comportement attendu tel qu'il peut être prédit grâce à des modèles de bon comportement. Elle repose sur la théorie logique du diagnostic introduite par [REI 87]. Les travaux les plus marquants sur ces approches de diagnostic à base de modèles sont regroupés dans [HAM 92]. Le but du diagnostic logique est de déterminer les équipements (ou composants) du système dont le fonctionnement anormal peut expliquer les incohérences détectées entre les comportements prédits et les observations du système [DEK 87]. Il s'agit d'un diagnostic abductif, le but est de trouver l'ensemble des causes qui expliquent les observations. Le principe de cette approche logique basée sur la cohérence est synthétisé dans l'ouvrage [DUB 01] et est illustré sur la figure 1.8.

L'approche de diagnostic logique repose essentiellement sur la notion de *conflit*. Cette notion sera introduite après avoir défini le modèle qualitatif du système qui sert de référence pour la détection de fautes dans le système.

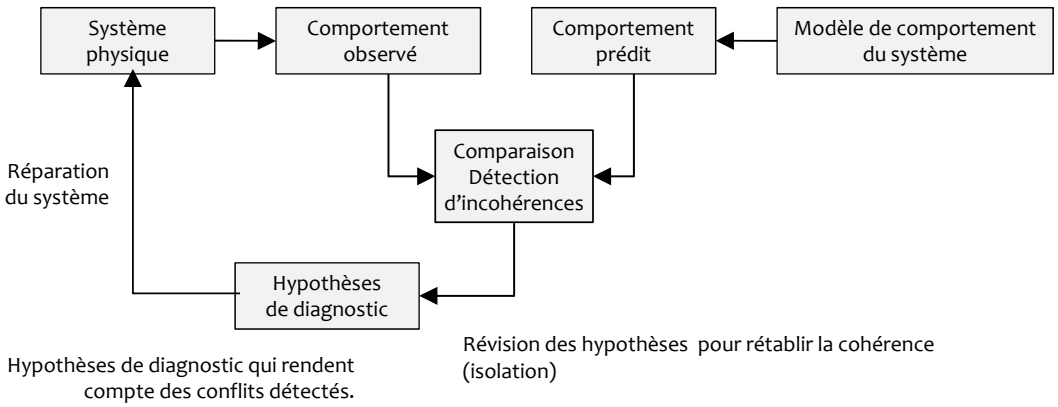

FIG. 1.8 – Approche logique basée sur la cohérence

Le modèle Le modèle utilisé par cette approche est un modèle qualitatif qui décrit la structure du système et le comportement des composants du système. Le système est représenté par la paire $(SD, COMPS)$, dans laquelle

- SD représente un ensemble de formules logiques du premier ordre qui décrivent la structure du système, c'est-à-dire l'ensemble des composants et leurs connexions, ainsi que le comportement des composants du système,
- $COMPS$ représente un ensemble fini de constantes qui représentent les composants du système.

Le modèle de comportement nominal du système est suffisant pour détecter des incohérences. Le prédicat Ab est utilisé pour indiquer qu'un composant ne se comporte pas correctement. La notation $Ab(C)$ signifie que le composant $C \in COMPS$ fonctionne anormalement, C est dit en faute. La notation $\neg Ab(C)$ veut donc dire au contraire que le composant C fonctionne correctement (C est correct).

Un système observé est défini par le modèle $(SD, COMPS, OBS)$, où OBS est un ensemble de formules du premier ordre qui représentent les observations disponibles par les capteurs positionnés sur le système. C'est à partir de ces observations et du modèle de comportement nominal que des incohérences peuvent être détectées.

Notion de conflit Le comportement d'un système est supposé anormal dès qu'une incohérence est détectée entre les observations et les prédictions faites à partir du modèle de comportement nominal du système. Les incohérences détectées ne se contentent pas de manifester la présence de fautes dans le système, elle renseignent également sur la localisation de ces fautes. Il suffit d'utiliser les prédictions qui ont menées à ces incohérences. Si une prédiction a été faite en utilisant les modèles de comportement nominal des composants $\{c_1, \ldots, c_n\} \in COMPS$ et qu'elle entre en contradiction avec une observation, c'est donc que les composants $c_1, \ldots, c_n$ ne peuvent être tous corrects et que

l'un d'eux est nécessairement en faute. On dit que ces composants forment un conflit. La notion de conflit introduite dans [REI 87] est au cœur du problème de diagnostic logique :

Définition 10 (Conflit). *Un conflit pour un système* $(SD, COMPS, OBS)$ *est un ensemble de composants* $\{c_1, \ldots, c_k\} \subseteq COMPS$ *tel que*

$$SD \cup OBS \cup \{\neg Ab(c_1), ..., \neg Ab(c_k)\} \text{ est incohérent.} \tag{1.7}$$

Un conflit correspond donc à un ensemble de composants qui ne suivent plus le modèle de comportement nominal. Plus on accumule d'observations, plus on effectue de prédictions, et plus on a des chances d'obtenir des contradictions, donc des nouveaux conflits. En regroupant ces conflits, on va progressivement affiner la localisation de la faute. La détection des conflits constitue la première phase du diagnostic à base de modèles.

Hypothèse de diagnostic La seconde phase du diagnostic consiste à générer des hypothèses sur les comportements des composants du système. Ces hypothèses rendent compte de tous les conflits, c'est-à-dire de toutes les incohérences détectées. Cela revient à changer l'hypothèse de fonctionnement correct de certains composants en une hypothèse de dysfonctionnement (de fonctionnement anormal), de manière à ce que toutes les contradictions disparaissent, c'est-à-dire qu'il n'y ait plus de conflit. Un ensemble de composants qui, cessant d'être supposés corrects, rétablit la cohérence avec les observations est précisément appelé un diagnostic. C'est le principe du diagnostic basé cohérence.

Le problème de diagnostic consiste à attribuer un mode de comportement (comportement normal ou anormal), représenté par Ab et $\neg Ab$, à chaque composant du système de manière à éliminer les conflits détectés. Un diagnostic est formellement défini de la manière suivante.

Définition 11 (Diagnostic). *Un diagnostic pour un système observé* $(SD, COMPS, OBS)$ *est un ensemble de composants* $\Delta \subseteq COMPS$ *tel que* $SD \cup OBS \cup \{\neg Ab(c), c \in COMPS \setminus \Delta\} \cup \{Ab(c), c \in \Delta\}$ *est cohérent.*

Pour un ensemble de composants donné, il y a généralement plusieurs diagnostics possibles. En appliquant le principe de parsimonie, on ne s'intéresse en général qu'aux diagnostics minimaux. Un diagnostic Δ est minimal s'il n'existe pas de diagnostic Δ' tel que $\Delta' \subset \Delta$.

Modèle de faute Même si l'idée première du diagnostic à base de modèles est de se passer de connaissances sur les fautes et les dysfonctionnements du système, si de telles connaissances sont disponibles, elles peuvent aider à la localisation des fautes.

Elles sont généralement indispensables si l'on souhaite identifier les fautes après les avoir localisées.

Au lieu de n'avoir que deux modes de comportement par composant correct et anormal, dont seul le premier est modélisé ($\neg Ab$), plusieurs modes de dysfonctionnement correspondant aux fautes connues possibles seront modélisés [DEK 89]. Pour des soucis de complétude, un mode inconnu dépourvu de tout modèle est toujours introduit et est censé regrouper tous les comportements de dysfonctionnement associés aux fautes non répertoriées. Un composant $c \in COMPS$ aura ainsi pour modes : $\{N(c), F_1(c), \ldots, F_m(c), I(c)\}$, avec N, le mode de fonctionnement normal (correct), $F_m, \ldots, F_m$, les modes de faute connus et I représentant le mode inconnu.

Un conflit devient dans ce cas une assignation de modes comportementaux à certains composants qui est en contradiction avec les observations du système. Un diagnostic est alors une affectation de modes de comportement à tous les composants du système qui rétablit la cohérence avec les observations [DEK 92] : $N(c_1) \wedge F_1(c_2) \wedge \ldots \wedge F_j(c_n)$.

Diagnostic abductif Par définition, le diagnostic est une procédure abductive : on détecte et on localise les fautes à partir des observations du système. Lorsque des modèles de faute sont disponibles, il est possible d'expliquer ces observations. Le diagnostic est dit explicatif, on cherche les causes possibles qui pourraient expliquer les symptômes (c'est-à-dire les effets observés) des défaillances [CON 91].

L'ensemble des observations OBS du système est divisé en deux sous-ensembles : $OBS = OBS_C \cup OBS_A$, où OBS_A représente les observations que l'on veut expliquer, les symptômes, et OBS_C sont des observations cohérentes avec le modèle décrit dans SD. Un symptôme correspond à un sous-ensemble d'observations de OBS_A. Une observation de OBS peut faire partie d'un ou plusieurs symptômes (OBS_A) ou peut n'en faire partie d'aucun (OBS_C). Un diagnostic abductif Δ explique les observations de OBS_A en étant cohérent avec les observations de OBS_C.

Définition 12 (Diagnostic abductif). *Un diagnostic abductif pour un système observé* $(SD, COMPS, OBS_C \cup OBS_A)$ *est un ensemble de composants* $\Delta \subseteq COMPS$ *tel que* $SD \cup OBS_C \cup \{Ab(c), c \in \Delta\} \cup \{\neg Ab(c), c \in COMPS \setminus \Delta\} \models OBS_A$.

Un diagnostic abductif est une explication (une cause) possible de OBS_A. Le fait que tous les composants de Δ soient anormaux a pour effet l'apparition des symptômes OBS_A. Aucun diagnostic explicatif n'implique l'absence d'une observation de OBS_C : $SD \cup OBS_C \cup \{Ab(c), c \in \Delta\} \cup \{\neg Ab(c), c \in COMPS \setminus \Delta\}$ est cohérent.

Lorsque $OBS_A = \emptyset$, il s'agit d'un diagnostic purement fondé sur la cohérence et lorsque $OBS_C = \emptyset$, c'est un diagnostic purement abductif.

1.3.2.4 Equivalences des deux approches : Bridge FDI-DX

Les communautés FDI et DX ont travaillé en parallèle sur le diagnostic à base de modèles durant de nombreuses années. Un cadre formel présenté dans [COR 04], permet de comparer les deux approches et d'établir un pont entre les deux communautés.

Dans les deux approches FDI et DX, le diagnostic se base sur un modèle explicite du comportement normal du système. L'occurrence d'une faute est détectée à partir des incohérences entre le comportement observé et le comportement prédit à l'aide du modèle du système. L'isolation des fautes repose sur l'analyse des ensembles de composants impliqués dans chaque incohérence détectée. Des preuves théoriques sur des équivalences concernant la modélisation du système, les observations, le concept de faute et la définition du diagnostic sont données. Un cadre unifié est proposé en reliant le concept de relations de redondance analytique aux conflits structurels définis dans le diagnostic logique. Les liens existant entre les deux approches FDI et DX sont également étudiés dans [BIS 04].

Cette étude comparative montre que même si les deux approches de diagnostic s'appuient sur des concepts ou des hypothèses différents, il est possible d'établir des équivalences entre elles et de proposer un cadre générique unificateur quelle que soit la technique de diagnostic utilisée.

1.3.3 Conclusion

Cette section sur le diagnostic pour la maintenance ne constitue pas un état de l'art exhaustif des méthodes de diagnostic existantes. Néanmoins, les techniques décrites ici sont les plus connues et les plus couramment utilisées. Le choix d'une méthode de diagnostic dépend de la connaissance du système, de la présence de capteurs ou de modèles qui permettent de suivre l'état réel du système. Pour optimiser la maintenance préventive et réduire les coûts liés aux défaillances, il est nécessaire de surveiller et d'analyser le plus précisément possible le comportement réel du système. On s'intéressera donc de manière naturelle aux approches de diagnostic à base de modèles qui se basent sur une connaissance profonde de la structure et du comportement du système à diagnostiquer.

1.4 Pronostic pour la maintenance

1.4.1 Concepts généraux

Les travaux concernant le pronostic sont assez récents. On trouve dans la littérature plusieurs définitions qui se rapportent au pronostic [GOH 06], mais deux d'entre elles reviennent principalement [BRO 00].

Définition 13. *Le pronostic consiste à calculer une prédiction de l'état d'un composant ou d'un système dans le futur.*

Cette définition reste très générale et assimile le pronostic à une prédiction d'états de croyance du système. Le temps n'est pas nécessairement quantifié. Or, du point de vue de la maintenance, il est nécessaire de prédire la date à partir de laquelle le composant ou le système évolue d'un état opérationnel (en bon fonctionnement) vers un état non opérationnel, c'est-à-dire en fonctionnement anormal. Une deuxième définition, énoncée ci-dessous, tient compte des objectifs de la maintenance préventive.

Définition 14. *Le pronostic est la capacité de prédire la durée de vie résiduelle (RUL pour "Remaining Useful Life") de composants ou systèmes en service.*

La durée de vie résiduelle (RUL) d'un système correspond au temps restant avant que le système ne puisse plus réaliser avec succès ses fonctions requises et doive être remplacé [ENG 00]. La durée de vie d'un système peut s'exprimer en heures de fonctionnement, en kilomètres parcourus (pour un système automobile), ou en nombre d'utilisations (nombre d'ouvertures pour un relais électrique, nombre d'atterrissages pour des pneus d'avion). Le cas le plus courant est la mesure en heures de fonctionnement. Avec la prédiction du RUL des composants, le pronostic aide l'opérateur de maintenance en recommandant des actions de maintenance appropriées afin de prévenir une défaillance du système [LEB 01]. Les deux définitions données sont en fait liées. Le calcul du RUL repose sur une loi de vieillissement (ou loi d'usure, loi de dégradation) qui peut être obtenue à partir de données issues de bancs de test [KEL 07]. Une loi de vieillissement décrit l'évolution de l'état de santé (ou de dégradation) d'un système tout au long de sa vie. L'état de santé d'un système est défini à partir de propriétés spécifiques aux composants et des sollicitations du système (conditions opérationnelles). Le RUL est ensuite estimé à partir d'un seuil de dégradation fixé a priori comme illustré sur la figure 1.9.

L'état de santé d'un système peut être affecté par des sollicitations anormales ou inattendues du système qui accélèrent ou ralentissent son vieillissement et modifient donc par conséquent son RUL. Afin de déterminer de manière précise le RUL d'un système, il est nécessaire que la loi de vieillissement tienne compte des facteurs de stress qui influencent réellement le système tout au long de sa vie [KIR 04] [WIL 04].

Définition 15 (Facteurs de stress). *Les facteurs de stress représentent toutes les sollicitations (normales et anormales) qui peuvent influencer l'état de santé du système.*

Parmi les facteurs de stress d'un système, on trouvera donc les conditions opérationnelles du système (conditions environnementales telles que l'humidité, la pression, la surtension, la température, les vibrations mécaniques, ...), les fautes pouvant survenir sur les composants et leurs conséquences sur les autres composants du système.

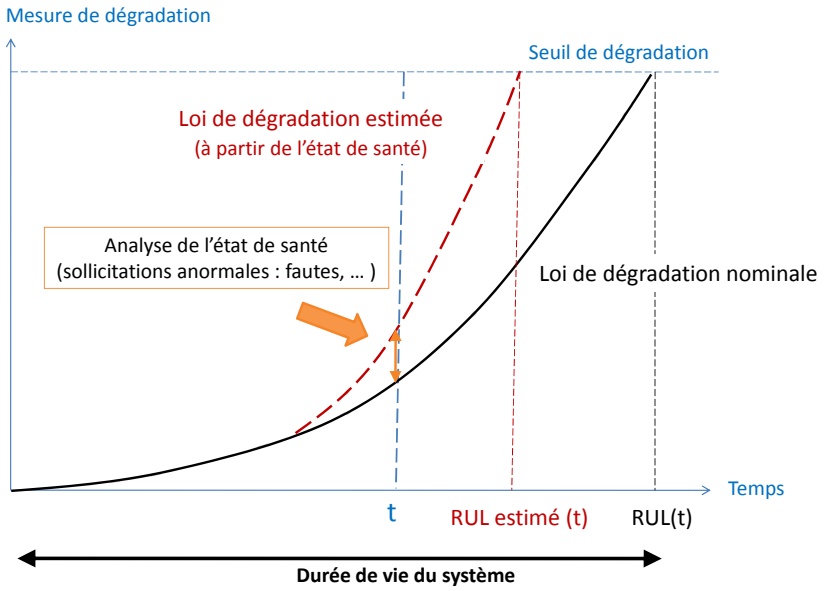

FIG. 1.9 – Loi de vieillissement pour le pronostic

La maintenance préventive prévisionnelle nécessite de connaître et d'analyser les facteurs de stress qui influencent réellement un système. Les décisions de maintenance reposent donc sur une analyse complète et efficace de l'état de santé du système lors de son fonctionnement [BYI 04] [CAM 07] par des fonctions de diagnostic et de pronostic. Les méthodes de diagnostic utilisent une connaissance sur le comportement du système alors que les méthodes de pronostic reposent plutôt sur une connaissance de l'état de santé (état de dégradation) du système. Par contre, tout comme le diagnostic, la précision du pronostic dépend particulièrement de la connaissance que l'on a du système considéré.

1.4.2 Méthodes de pronostic

La connaissance utilisée par le pronostic est contenue dans un modèle qui représente le vieillissement ou l'usure du système. Ce modèle de pronostic est connu a priori et utilisé en ligne. Dans la littérature, il existe déjà plusieurs approches de pronostic qui reposent sur différents modèles [BRO 00][ROE 05][GHE 06][SCH 07b]. Ces approches de pronostic peuvent être hiérarchisées selon l'étendue de leur domaine d'application et leur complexité. Cette hiérarchisation est représentée par une pyramide illustrée par la figure 1.10.

La classification des méthodes pour le pronostic est identique à celle des méthodes pour le diagnostic (cf figure 1.3). Elle dépend de la connaissance que l'on a du système à pronostiquer. Le niveau de connaissance disponible est tout d'abord caractérisé par

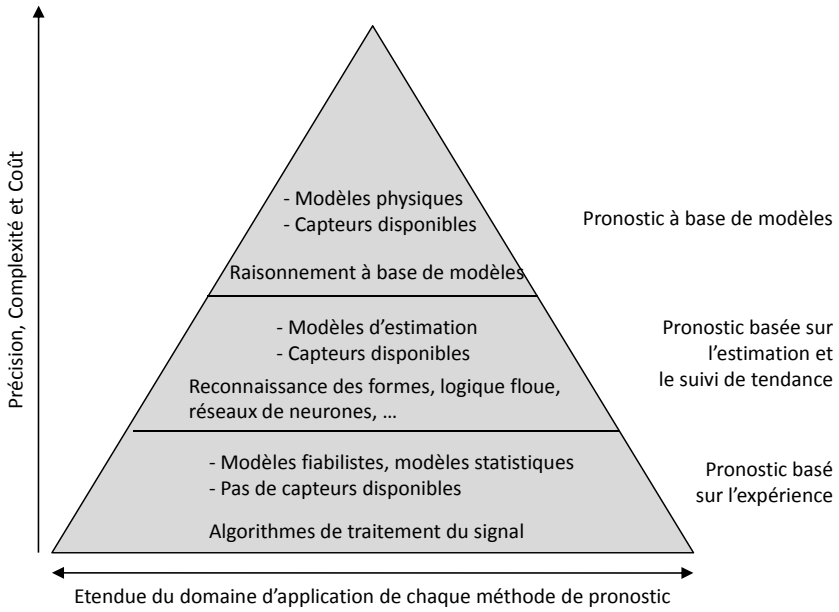

FIG. 1.10 – Classification des méthodes de pronostic

la présence de capteurs sur le système. Selon la qualité et la quantité de capteurs, il est possible d'obtenir en ligne des informations sur l'état du système (données enregistrées, mesures, messages, alarmes, ...). Ces informations sont appelées des observations en ligne du système. Le second type de connaissance dont on dispose est un modèle de pronostic qui décrit l'évolution de l'état de dégradation du système. Ce modèle de pronostic peut être très pauvre en l'absence de capteurs et repose uniquement sur l'expérience. Mais il peut être également très riche dans le cas où des observations en ligne sont considérées et extrapolées par des raisonnements physiques sur les composants susceptibles de tomber en panne dans le futur. La figure 1.10 illustre les trois classes de méthodes de pronostic qui reposent sur différents niveaux de connaissance disponible.

1.4.2.1 Approche basée sur la connaissance

Les méthodes de pronostic basé sur l'expérience se situent en bas de la pyramide. Ces méthodes constituent la seule alternative possible dans la situation où il n'y a aucune connaissance disponible sur la nature physique du système et ses composants. Le nombre de capteurs est limité et n'est généralement pas conçu pour résoudre un problème de pronostic. Une approche de pronostic basée sur l'expérience repose sur un modèle de pronostic obtenu à partir d'une connaissance de surface. Cette forme de modèle est la plus simple et ne nécessite que l'historique des défaillances ou les recommandations de conception des composants dans des conditions opérationnelles similaires afin de déterminer la probabilité de défaillance à un moment donné dans le

futur. Cette technique dérive des systèmes experts basés sur un ensemble de règles et d'associations. Les systèmes experts sont également utilisés à des fins de diagnostic, leur fonctionnement a été décrit dans le paragraphe 1.3.2.1. Des techniques issues du domaine de la fiabilité permettent d'associer une distribution statistique aux données.

Raisonnement à base de cas

Un exemple de pronostic basé sur l'expérience est le raisonnement à base de cas [BER 06]. Le raisonnement à base de cas est un outil de calcul dont la seule source de connaissance est une mémoire des cas importants passés. Quand un problème est résolu, sa solution est retenue pour résoudre des problèmes similaires. Cela permet alors d'éviter de répéter les erreurs commises dans le passé.

Etude de fiabilité

Les prédictions de fiabilité reposent sur des modèles basés sur l'expérience qui permettent d'établir une loi de durée de vie à partir d'un échantillon de population [KAU 75]. Une étude de fiabilité utilise la connaissance a priori et le traitement statistique des données pour prédire la probabilité de défaillance d'un système tout au long de sa vie et donc déterminer sa durée de vie (RUL). Pour cela, la durée de vie d'un système est représentée par une variable aléatoire X. On associe à cette durée de vie X une loi de probabilité établie sur des statistiques. La probabilité que la durée de vie X d'un système soit inférieure à une date t_p s'écrit alors

$$P(X \leq t_p) = \int_0^{t_p} f(t)dt, \tag{1.8}$$

où $f(t)$ représente la fonction de densité de probabilité (pdf pour "probability density function") de défaillance.

Le taux de défaillance d'un système, noté $\lambda(t)$, représente la probabilité pour que le système tombe en panne dans l'intervalle $]t, t+dt]$ sachant qu'il était en bon état à l'instant t :

$$\begin{aligned} \lambda(t)dt &= P(X \leq t+dt | X > t) \\ &= \frac{P(t<X\leq t+dt)}{P(X>t)}. \end{aligned} \tag{1.9}$$

La figure 1.11 représente une courbe idéalisée de la fiabilité qui décrit l'évolution du taux de défaillance, c'est-à-dire du nombre de défaillances par unité de temps (heures, cycles, ...) et illustre les trois principales régions de vie d'un système.

- Le période de jeunesse correspond à la période de mortalité infantile. Il est négligeable pour la plupart des composants lorsque qu'une phase de déverminage ou une phase de test est effectuée.

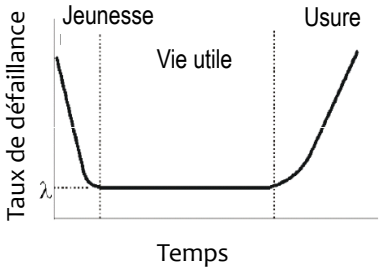

FIG. 1.11 – Evolution du taux de défaillance

- La période de vie utile modélise le comportement d'un système sans vieillissement sujet aux défaillances aléatoires (erreurs humaines, fautes, sollicitations anormales, ...). Le taux de défaillance est considéré constant durant cette phase.
- La période d'usure correspond à la dégradation physique, au vieillissement accéléré du système en fin de vie. Le taux de défaillance est croissant.

Pour représenter la probabilité de défaillance d'un système, on utilise généralement des lois de probabilité paramétrées et connues telle que la loi normale, la loi logarithmique ou exponentielle, la loi de Weibull [HUM 02] [KIR 04] [FER 08].

Le pronostic basé sur l'expérience ne permet pas d'évaluer l'état de dégradation du système. Il ne tient pas compte de la manière dont les composants du système sont utilisés. Ce sont des méthodes hors ligne qui ne permettent pas d'obtenir un pronostic adaptatif.

1.4.2.2 Approche à base de données

Les approches de pronostic à base de données s'appuient sur des méthodes d'estimation en ligne et des méthodes d'apprentissage (méthodes de suivi de tendances) [BYI 02] [SCH 07b]. Elles sont utilisées dans le cas où des observations en ligne sont disponibles mais où on ne possède toujours pas de connaissance physique sur la nature du système. Les modèles de pronostic considérés dans ces approches reposent alors sur des estimateurs (estimateur d'état, estimateur de paramètre) qui sont utilisés en ligne pour évaluer l'état de dégradation du système. Les paramètres considérés ici sont des caractéristiques mesurables du système qui peuvent être utilisées comme indicateur de vieillissement pour prédire les défaillances et donner des estimations du RUL [ROE 05]. Les estimateurs sont obtenus à partir des observations en ligne et de l'historique de défaillances (identification de motifs de faute) en utilisant des techniques d'apprentissage (réseaux de neurones) ou en identifiant des paramètres d'estimateurs classiques tel que le filtre de Kalman par exemple.

Estimation d'état et de paramètres

Les méthodes de pronostic par estimation d'état ou de paramètres sont utilisées pour prédire le comportement d'un paramètre révélateur de l'état de dégradation du système. Il s'agit de minimiser l'écart entre un modèle de référence de ce paramètre significatif et les mesures obtenues à partir des capteurs à un instant donné. Le modèle de pronostic utilisé n'a pas de connaissance physique du système mais il est le résultat d'une analyse mathématique du comportement. Des techniques classiques d'automatique pour la prédiction de mesures sur des paramètres en fonction des mesures précédentes sont utilisées. Une fonction d'extrapolation sur la mesure est donc nécessaire. Par exemple, la fonction d'extrapolation tangentielle calcule une prédiction de mesure sur un paramètre à l'instant $n+1$ en fonction de la mesure de ce paramètre à l'instant n :

$$f(n+1) = f(n) + \dot{f}(n)t + \frac{1}{2}\ddot{f}(n)t^2, \tag{1.10}$$

où $f(n)$ est une mesure à la date n d'un paramètre du système à pronostiquer et t est la période entre deux instants de mesure. Cette prédiction repose seulement sur la mesure à l'instant précédent et sur une équation dynamique. La méthode d'estimation permet d'obtenir un pronostic adaptatif. Le filtre de Kalman peut être utilisé pour minimiser l'erreur entre le modèle de pronostic et les données observées pour prédire le comportement futur d'un paramètre du système.

Méthodes d'apprentissage

La fonction d'extrapolation tangentielle ne considère aucune autre mesure des paramètres du système. Elle repose sur des hypothèses fortes sur le système et peut être totalement imprécise. Pour résoudre ce problème, des techniques d'apprentissage sont utilisées pour enrichir la fonction d'extrapolation f. La fonction d'apprentissage utilise l'historique des mesures du système pour faire des prédictions sur le comportement de chaque paramètre du système pouvant se dégrader. L'apprentissage va inférer des règles générales à l'aide d'un expert capable d'associer aux mesures la dégradation d'un paramètre du système qui a eu lieu après l'observation de ces mesures. Cette association "données-dégradation" est un étiquetage de données. Les nouvelles mesures sont observées et en utilisant ces règles apprises, certaines situations du passé sur le point de se reproduire sont reconnues (à savoir la dégradation d'un paramètre). Des modes de défaillance sont déterminés à partir de l'historique des mesures auquel on se réfère ensuite dans la phase d'estimation pour la détection d'un paramètre dans un mode. Il existe plusieurs techniques d'apprentissage : les techniques statistiques et les réseaux de neurones.

Apprentissage statistique L'idée principale des techniques d'apprentissage statistiques (classification statistique telle que l'analyse en composantes principales ou les machines à vecteurs de support) est de déterminer des sous-espaces dans un espace

vectoriel dans lequel sont représentés des mesures du système [SAM 08]. Lors de la phase d'estimation, une nouvelle mesure d'un paramètre est observé et selon le sous-espace dans lequel elle se situe, un mode de dégradation peut être associé au paramètre considéré.

Les réseaux de neurones La fonction d'estimation est réalisée par un réseau de neurones qui modélise la progression d'un paramètre jusqu'à la défaillance du système [WAN 99]. L'apprentissage va consister à ajuster les poids et les seuils des neurones afin d'obtenir les sorties prédites désirées pour chaque paramètre du système (méthode de rétro-propagation du gradient). Le réseau de neurones est ensuite utilisé pour prédire l'évolution de la dégradation de ces paramètres dans des conditions opérationnelles similaires.

Un exemple d'approche de pronostic à base de données appliquée à un moteur de turbine à gaz est développé dans [BYI 02]. Cette approche repose sur des paramètres spécifiques du système et sur un mode pour l'efficacité du compresseur. Une technique probabiliste est développée. Elle utilise l'information disponible sur la manière dont évolue la dégradation des paramètres dans le temps afin d'évaluer la sévérité du changement de distribution de ces paramètres et de projeter leur état futur.

1.4.2.3 Approche à base de modèles

Les techniques à base de modèles sont représentées tout en haut de la pyramide. Ce sont les techniques les plus précises pour le pronostic, elles reposent sur une connaissance profonde du système à pronostiquer.

Modèle physique

Les méthodes de pronostic à base de modèle physique permettent non seulement d'observer des réalités physiques (conditions opérationnelles environnementales : vibrations, température, l'humidité, etc ...) à l'aide de capteurs positionnés sur le système mais également de retourner, en exploitant un modèle, les causes associées à ces observations [VAL 03] [KIR 04] [VAC 06]. Le pronostic à base de modèles est une approche de modélisation techniquement complète qui est classiquement utilisée pour pronostiquer le mode de défaillance d'un composant. Un mode de défaillance est l'effet par lequel une défaillance est observée. Il est possible de modéliser les dommages accumulés par les cycles de fatigue d'un composant (qu'il soit électrique ou mécanique) et de modéliser l'impact des contraintes environnementales en donnant le nombre de cycles qu'il reste jusqu'à la défaillance du système. Les dommages et la défaillance sont reliés par une expression mathématique. Le modèle de pronostic utilisé est un modèle continu représenté par un ensemble d'équations faisant intervenir des contraintes environnementales [HUM 02][WIL 04].

Cette approche à base de modèle physique est illustrée sur l'exemple d'un module de boîte de vitesse dans [BYI 02]. Un estimateur physique statistique d'une fissure d'une dent du moteur est utilisé. Cette fissure correspond au dommage accumulé par les cycles de fatigue du moteur. Une expression mathématique reposant sur la loi de Miner relie le dommage au nombre de cycles jusqu'au début d'une fissure d'une dent du moteur :

$$Dommage = \left(\frac{n}{N_f}\right)^r \tag{1.11}$$

où

- n est le nombre de cycles effectués,
- r est un exposant non linéaire lié au dommage,
- N_f représente le nombre de cycles jusqu'à l'apparition de la fissure.

Un dommage supérieur à 1 représente le début d'une fissure dans la boîte de vitesse. Une équation différentielle est ensuite utilisée pour prédire la date de défaillance de la boîte de vitesse en estimant l'évolution du craquage par rapport au nombre de cycles effectués.

Un second exemple dans [GHE 06] s'intéresse à l'estimation de la consommation de vie des composants électriques dans des conditions anormales de température et de vibration. L'accumulation du dommage est modélisée par une règle et une défaillance apparaît lorsque le dommage accumulé est supérieur à 1 comme dans le premier exemple du composant mécanique. Les dommages et l'impact d'un paramètre environnemental de température du composant peuvent être modélisés par une équation reposant sur la loi de Coffin-Manson à partir de laquelle on peut obtenir le nombre de cycles qu'il reste jusqu'à la défaillance du composant :

$$N_f = \frac{1}{2}\left(\frac{\triangle W}{2\varepsilon_f}\right)^{\frac{1}{c}}, \tag{1.12}$$

où

- N_f représente le nombre de cycles jusqu'à la défaillance du composant,
- $2\epsilon_f$ est le coefficient de ductilité de fatigue,
- c correspond à l'exposant de ductilité de fatigue,
- $\triangle W$ est la densité d'énergie de tension cyclique maximale, qui est fonction du joint de soudure.

Un critère de décision permet ensuite de prédire la défaillance dans un composant selon le nombre de cycles opérationnels et le nombre total de cycles qui produiraient une défaillance considérant un certain niveau de stress :

$$\sum_i \frac{n_i}{N_i} \geq 1 \text{ en cas de défaillance },$$

où n_i est le nombre de cycles opérationnels et N_i est le nombre total de cycles qui produiraient une défaillance au niveau de stress S_i.

Dans le cas du stress induit par des vibrations, la fatigue du matériel peut être également modélisée par une équation qui détermine le nombre de cycles jusqu'à la défaillance à partir desquels on peut évaluer le dommage :

$$S \ . \ N_f = Cst, \tag{1.13}$$

où S représente le facteur de stress dû aux vibrations qui est une fonction des propriétés matérielles du joint de soudure, Cst est une constance propre au matériau étudié et N_f est le nombre de cycles jusqu'à la défaillance. La sévérité du dommage conduit finalement à l'estimation de la durée de vie résiduelle des composants électriques.

Modèle de simulation

Un modèle de simulation est utilisé dans [BIS 06]. La simulation d'un modèle à partir d'un état de croyance utilise l'historique des observations pour établir les tendances et prédire le comportement futur du système. Elle permet de détecter les déviations à partir du comportement attendu. Un modèle de simulation permet de relier les causes aux performances du système.

Modèle à événements discrets

Il n'existe que quelques travaux sur le pronostic à base de modèles à événements discrets. Comme dans le cas continu, on dispose d'un modèle avec une connaissance sur la nature physique du système et des données observées. Le comportement du système est modélisé par un système de transitions :

$$G = (X, T, \Sigma, x_0), \tag{1.14}$$

où

- X est un ensemble d'états,
- T est un ensemble de transitions,
- Σ est un ensemble d'événements,
- x_0 est l'état initial du système.

Le but est de vérifier si un événement significatif (faute ou événement critique) du SED peut toujours être prédit à partir d'observations partielles du système. Dans ce cas, l'événement est dit prédictible. La notion de prédictabilité a été introduite et analysée dans les systèmes à événements discrets (SED) dans [CAO 89].

Définition 16 (Prédictabilité). *L'occurrence d'un événement significatif s est prédictible dans le système G si après l'observation d'une séquence finie d'événements, un système de surveillance peut dire avec certitude que l'événement s va apparaître.*

Ce problème est proche de celui du diagnostic de faute dans les SED. Il existe la même relation entre la prédictabilité et la prédiction qu'entre la diagnosticabilité

et le diagnostic [GEN 06] [JER 07]. Par cette analogie, les méthodes qui vérifient la prédictabilité sont les mêmes que celles qui vérifient la diagnosticabilité (approche diagnostiqueur).

1.4.3 Conclusion

Pour optimiser la maintenance préventive, il est nécessaire d'intégrer dans le système de supervision une fonction de pronostic afin de surveiller et d'analyser de manière précise l'état de santé (ou état de dégradation) du système. Un système complexe résulte d'un assemblage de composants totalement hétérogènes (composants continus, discrets, hybrides). La connaissance disponible sur chaque composant du système sur laquelle s'appuient les méthodes de pronostic peut être vraiment très différente d'un composant à l'autre. Différentes techniques doivent être utilisées pour surveiller et analyser l'état de santé de chaque composant.

Pour des parties critiques du système, il est possible d'avoir des modèles très précis mais pour d'autres parties, en l'absence de capteurs, la seule connaissance disponible provient de l'expérience (étude de fiabilité, analyse statistique). Les modèles physiques sont les plus précis mais sont également les plus difficiles à obtenir.

1.5 Discussion

Le but de cette thèse est d'optimiser la maintenance préventive d'un système complexe en réduisant les coûts de panne et d'indisponibilité du système. Pour cela des méthodes de diagnostic et de pronostic sont nécessaires.

Un système complexe est un ensemble de composants hétérogènes. La connaissance disponible sur laquelle s'appuient les méthodes de diagnostic et de pronostic peut être très différente d'un composant à l'autre. Notre objectif est de développer un système de supervision qui intègre une fonction de diagnostic et de pronostic et qui fournit la même représentation pour chaque composant surveillé du système complexe quelle que soit la connaissance disponible (expérience, données, modèles).

La difficulté qui s'en suit concerne la description du système complexe. Il faut en effet trouver le bon niveau d'abstraction pour décrire la connaissance du système qui sera utilisée par ce système de supervision. Les méthodes de diagnostic et de pronostic s'appuient chacunes sur des indicateurs de faute et de vieillissement qui sont représentés par des paramètres spécifiques des composants du système. Ces paramètres vont nous permettre d'obtenir une description abstraite mais homogène du système complexe.

Nous allons présenter une caractérisation générique formelle d'un système de supervision puis nous allons caractériser le problème de surveillance comme une combi-

naison des problèmes de diagnostic et de pronostic. En effet le diagnostic doit identifier les composants en faute responsables des défaillances pouvant générer un stress inattendu sur d'autres composants du système, ce qui aurait comme effet d'accélérer l'usure du système global.

2 Cadre de modélisation générique pour un système complexe

Résumé : Ce chapitre propose un cadre de modélisation générique pour représenter la connaissance disponible pour un système complexe. On introduit un modèle structurel et un modèle fonctionnel afin de décrire l'ensemble des composants, leurs interactions et le comportement du système complexe. Cette modélisation s'appuie sur un ensemble de paramètres caractéristiques des composants, un ensemble de rangs pour ces paramètres ainsi qu'un ensemble de relations entre ces paramètres, à partir desquels il est possible de définir des modes de fonctionnement pour les composants du système. Ces modèles et ces modes de fonctionnement représentent la connaissance utilisée pour les problèmes de diagnostic et de pronostic.

2.1 Introduction

La fonction de maintenance doit intégrer des capacités de diagnostic et de pronostic. Pour ces deux problèmes, il est nécessaire de surveiller et d'analyser de manière précise la condition, l'état de fonctionnement de l'objet étudié. Cette analyse n'est possible que si l'on dispose d'un modèle de référence. Il faut donc trouver une représentation pour l'objet d'intérêt. Un système est un ensemble d'éléments physiques (matériel ou logiciel) qui est conçu dans le but d'assurer une fonction d'usage. Cette thèse porte sur la maintenance des systèmes complexes que l'on définit de la manière suivante.

Définition 17 (Système complexe). *Un système complexe est un ensemble structuré d'équipements hétérogènes indépendants qui sont connectés et communiquent entre eux dans le but d'assurer une fonction.*

Un équipement est un élément mécanique, électronique ou informatique. Un système complexe se compose d'équipements autonomes et interconnectés qui sont donc très souvent totalement hétérogènes. Un système complexe peut être décentralisé ou distribué. Les utilisateurs ont cependant l'impression de n'utiliser qu'un seul système. Comme exemple de systèmes complexes, on peut citer des réseaux de machines (ordinateurs standards connectés en réseau, serveur de fichiers, serveur web), ou des logiciels avec plusieurs processus sur une même machine (machine multiprocesseurs avec mémoire partagée).

Cet ensemble d'équipements communique dans le but de réaliser une application, une fonction spécifique. Comme ils proviennent de différents concepteurs, le type de connaissance sur le fonctionnement d'un équipement est variable d'un équipement à un autre (modèle continu, modèle discret, modèle hybride, ...). Les *fonctions élémentaires* mises en œuvre par chacun d'eux pour réaliser la fonction spécifique de l'entité globale complexe peuvent également être de différente nature.

On souhaite définir un cadre de modélisation générique pour un système complexe afin de représenter de manière homogène la connaissance disponible pour chaque équipement. La principale difficulté consiste à choisir le bon niveau d'abstraction pour décrire cette connaissance qui sera utilisée pour les problèmes de diagnostic et de pronostic.

2.2 Modélisation générique d'un système complexe

Cette section propose un cadre de modélisation générique pour un système complexe en s'appuyant sur la notion de *composant* communiquant qui sera définie par la suite. La connaissance disponible est représentée par un ensemble de *paramètres* spécifiques des composants, un ensemble de *rangs* et de *relations* entre ces paramètres qui permettent d'introduire un modèle structurel et un modèle fonctionnel du système complexe.

2.2.1 Système et composants

Un paramètre $p^k \in \mathcal{P}$ représente une variable (continue ou discrète) du modèle du système. Une relation $ar^k \in \mathcal{A}$ entre des paramètres de l'ensemble $\mathcal{P}$ est une équation algébrique, différentielle ou encore une équation logique qui lie des paramètres. Une relation $ar^k \in \mathcal{A}$ est une application de $2^{\mathcal{P}}$ (l'ensemble des parties de $\mathcal{P}$) dans $\mathcal{P}$, qui s'écrit de la manière suivante :

$$p^j = ar^k(p^i, \ldots, p^m). \tag{2.1}$$

Par exemple, l'équation de l'énergie cinétique d'un système représente une relation algébrique entre les paramètres E, m et v :

$$E = \frac{1}{2}mv^2,$$

où E est représente le paramètre d'énergie dont la valeur résulte de la relation algébrique, m est un paramètre de masse et v, le paramètre correspondant à la vitesse du système.

L'équation différentielle donnant la tension d'induction d'une machine à courant continu est un autre exemple de relation entre des paramètres :

$$u(t) = Ri(t) + L\frac{di}{dt} + E_m,$$

où $u(t)$ est le paramètre de tension d'induit fixé par la relation différentielle, R est un paramètre de résistance, $i(t)$ le paramètre d'intensité, L un paramètre d'inductance et E_m, un paramètre représentant la force contre-électromotrice.

Notons $\mathcal{R}$ l'ensemble des rangs des paramètres du système. Le rang d'un paramètre p^k représente l'ensemble des valeurs correctes ou admises pour ce paramètre et il est noté $r(p^k)$. Par exemple, pour un paramètre continu, le rang correspond souvent à un intervalle $r(p^k) = [\alpha^k, \beta^k]$ avec $(\alpha^k, \beta^k) \in \mathbb{R}^2$ et $\alpha^k \leq \beta^k$. Pour un paramètre discret représentant une variable booléenne, le rang est représenté par $r(p^k) = \{0, 1\}$.

Un système à événements discrets peut être également modélisé par un ensemble de paramètres, de rangs et de relations entre les paramètres. Prenons l'exemple d'une alimentation que l'on commande à l'aide d'un interrupteur. L'alimentation doit délivrer un courant continu. Lorsque l'alimentation est "hors-service", elle est incapable de fournir un courant quelle que soit la position de l'interrupteur. Les paramètres du

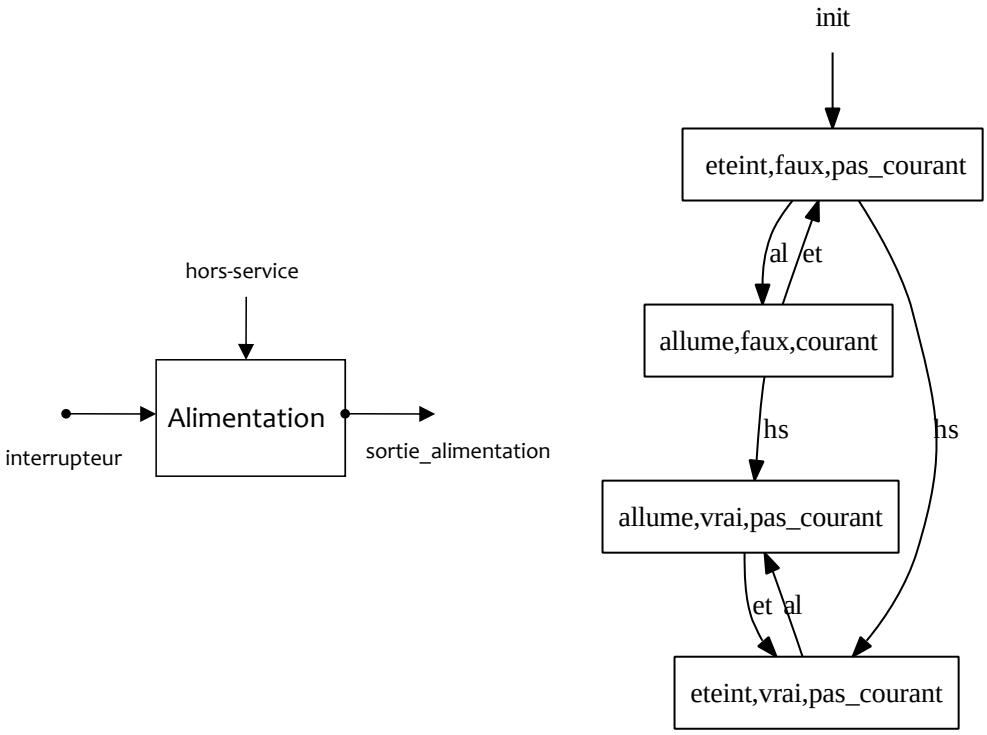

FIG. 2.1 – Modèle à événements discrets d'une alimentation

système d'alimentation sont : $\mathcal{P}$ = {interrupteur, hors_service, sortie_alimentation}, où interrupteur est un paramètre d'entrée, hors_service est un paramètre privé et sortie_alimentation est un paramètre de sortie du système d'alimentation. Les valeurs possibles pour chaque paramètre sont définies par leur rang :

- r(interrupteur) = { allume, eteint},
- r(hors_service) = {vrai, faux},
- r(sortie_alimentation) = {courant, pas_courant}.

Un courant est fourni par l'alimentation lorsque l'interrupteur est allumé et que l'alimentation n'est pas hors-service. On en déduit facilement la relation logique suivante :

$$\forall t_i, \quad (\text{sortie\_alimentation}(t_i) = \text{courant}) \equiv (\text{hors\_service}(t_i) = \text{faux} \wedge \text{interrupteur}(t_i) = \text{allume}), \quad (2.2)$$

où t_i est un instant. Une relation de transition pour le paramètre interrupteur serait par exemple :

$$\forall(\text{interrupteur}(t_i+1) = \text{allume}) \equiv (\text{interrupteur}(t_i) = \text{eteint} \wedge \text{evenement}(al, t_i+1)) \quad (2.3)$$

Le comportement de l'alimentation est défini par un automate représenté sur la figure 2.1 :

$$G = (X, E, T, x_0). \quad (2.4)$$

- L'ensemble des états de l'automate est noté X. Chaque état $x_i \in X$ est un triplet (interrupteur,hors_service, sortie_alimentation) qui prend ses valeurs dans l'espace {allume, eteint} × {vrai, faux} × {courant,pas_courant}.
- L'ensemble des événements est $E = \{al, et, hs\}$. L'événement qui correspond à l'action "allumer l'interrupteur" est noté al, l'événement qui correspond à l'action "éteindre l'interrupteur" est noté et et l'événement qui traduit le fait que l'alimentation est hors service est noté hs.
- La fonction de transition T est définie par $T : X \times E \rightarrow X$.
- L'état initial est x_0 = (eteint,faux,pas_courant).

Les paramètres du système d'alimentation sont représentés dans les états de l'automate : $\mathcal{P}$ = {interrupteur, hors_service, sortie_alimentation}. Chaque état correspond à une affectation de valeurs aux paramètres du système.

Le modèle d'un système complexe Σ est défini par le triplet suivant :

$$\Sigma = \langle \mathcal{P}, \mathcal{R}, \mathcal{A} \rangle \quad (2.5)$$

dans lequel
- $\mathcal{P} = \{p^k\}$ est l'ensemble des paramètres du modèle,
- $\mathcal{R} = \{r(p^k)\}$ est l'ensemble des rangs de valeur des paramètres,
- $\mathcal{A} = \{ar^k\}$ est l'ensemble des relations entre les paramètres.

Un système complexe Σ est composé d'équipements interconnectés qui communiquent entre eux. Cet ensemble d'équipements apparaît à un utilisateur comme une entité unique et cohérente. Dû à sa complexité, il est cependant pratiquement impossible de modéliser un tel système de manière monolithique, c'est-à-dire de manière globale. Une approche possible est de décomposer le système complexe Σ en un ensemble de composants.

DÉFINITION 18 (Composant). *Un composant est une entité élémentaire qui peut être remplacée en ligne par un opérateur de maintenance.*

Un équipement est constitué d'un ou plusieurs composants qui communiquent entre eux. Il s'agit donc d'un sous-système de l'entité globale complexe Σ. Notons $Comps = \{C^1, \ldots, C^N\}$, l'ensemble des N composants communiquants du système Σ. Un composant C^i peut être modélisé à partir d'un sous-ensemble des paramètres $\mathcal{P}$ du système et d'un sous-ensemble des relations $\mathcal{A}$ entre ces paramètres. En réutilisant les notations introduites précédemment, un composant $C^i \in Comps$ est modélisé par le triplet suivant :

$$C^i = \langle \mathcal{P}^i, \mathcal{R}^i, \mathcal{A}^i \rangle, \tag{2.6}$$

dans lequel $\mathcal{P}^i = \{p^{i,k}\} \subseteq \mathcal{P}$, $\mathcal{R}^i = \{r^{i,k}\} \subseteq \mathcal{R}$, et $\mathcal{A}^i = \{ar^{i,k}\} \subseteq \mathcal{A}$.

La définition de composant que l'on vient d'introduire et la modélisation générique à partir de paramètres et de relations nous permet de représenter un système complexe Σ par un ensemble donné de N composants tel que

$$\Sigma = \langle \mathcal{P}, \mathcal{R}, \mathcal{A} \rangle = \langle \bigcup_{i=1}^{N} \mathcal{P}^i, \bigcup_{i=1}^{N} \mathcal{R}^i, \bigcup_{i=1}^{N} \mathcal{A}^i \rangle. \tag{2.7}$$

Les composants du système communiquent à travers les paramètres partagés (voir la section 2.2.2) dans le but de réaliser un ensemble d'applications, de fonctions spécifiques. Le niveau de description du système doit être raffiné pour prendre en compte cette connaissance sur les interactions entre composants et sur les fonctions implémentées par le système.

Une approche multi-modèles est proposée dans [CHI 93] pour représenter un système physique. Les divers modèles utilisés se fondent sur différents niveaux d'agrégation de la connaissance. Quatre niveaux de connaissance sont identifiés.

- **La connaissance structurelle** porte sur la topologie du système. Elle décrit l'ensemble des composants du système, leurs connexions ainsi que leurs interactions au moyen de ports.
- **La connaissance comportementale** décrit le fonctionnement des composants ainsi que la manière dont ils interagissent entre eux par des relations (équations) établies entre des quantités physiques. Les quantités physiques sont des entités permettant de capturer la nature, l'état et le comportement d'un système. Il en existe trois types : les constantes, les paramètres et les variables d'état.
- **La connaissance fonctionnelle** décrit le comportement individuel des composants en terme de rôle et de processus qui contribuent à la réalisation de la fonction but (ou objectif) du système. Un composant est associé à un ou plusieurs rôles fonctionnels.
- **La connaissance téléologique** décrit l'ensemble des fonctions que le système doit réaliser en terme de but, d'objectif.

La notion de paramètre que l'on vient de définir dans cette thèse regroupe toutes les quantités physiques considérées dans [CHI 93] (les constantes et les variables sont des paramètres). Cette approche multi-modèles permet de représenter l'ensemble des connaissances fondamentales relatives à la structure d'un système complexe et aux fonctions qu'il réalise. Le travail de [CHI 93] montre qu'il est intéressant de réaliser une étude fonctionnelle au niveau des différents composants ainsi qu'une étude fonctionnelle au niveau du système global qui prendra en compte les objectifs du système.

Afin de définir de manière précise les concepts liés aux problèmes de diagnostic et de pronostic, il est nécessaire de raffiner le niveau de description d'un système en introduisant un modèle structurel et un modèle fonctionnel fondés sur le comportement des composants du système complexe. Ces deux modèles reposent sur les notions de paramètres et de relations entre paramètres que l'on vient de définir.

2.2.2 Modélisation structurelle

Un modèle structurel permet de décrire l'ensemble des composants d'un système ainsi que les connexions et les interactions qui sont possibles entre ces divers composants. Les interactions entre composants sont généralement modélisées par des *ports* qui permettent l'échange de données. Les flux de données doivent être considérés au moyen de paramètres au niveau du modèle. Bien évidemment, dans un système réel, un flux de données correspond à un flux physique (une tension, une intensité, une pression, ...) ou logique (communication, ...). Un port peut être modélisé par un *paramètre d'entrée* s'il correspond à un flux d'informations en entrée ou par un *paramètre de sortie* s'il correspond à un flux d'informations en sortie.

2.2.2.1 Niveau composant

Pour un composant C^i, il est possible de distinguer trois différents types de paramètres : les paramètres privés, les paramètres d'entrée et les paramètres de sortie.

Paramètre privé

Un paramètre $p^{i,k}$ est un paramètre privé du composant C^i que l'on note $pp^{i,k}$, s'il appartient uniquement à C^i. L'ensemble des paramètres privés du composant C^i est représenté par $\mathcal{PP}^i$. Formellement, le paramètre $pp^{i,k}$ appartient à $\mathcal{PP}^i$ si et seulement si

$$\nexists C^j, j \neq i \mid pp^{i,k} \in \mathcal{PP}^j. \tag{2.8}$$

Un paramètre privé $pp^{i,k}$ est un paramètre interne, spécifique à un composant qui peut être modifié par les mécanismes mis en œuvre par le composant. Les paramètres privés

sont nécessaires pour la généralité du modèle. Par exemple, sans paramètres privés, la description d'un système dynamique à l'aide d'un vecteur d'état serait impossible.

Paramètre d'entrée

Un paramètre $p^{i,k}$ est un paramètre d'entrée du composant C^i que l'on note $ip^{i,k}$, si les mécanismes mis en œuvre par C^i ne peuvent pas modifier sa valeur. Un paramètre d'entrée d'un composant peut être partagé et appartenir également à un autre composant. L'ensemble des paramètres d'entrée associés au composant C^i est représenté par $\mathcal{IP}^i$. Formellement, le paramètre $ip^{i,k}$ appartient à $\mathcal{IP}^i$ si et seulement si

$$\exists C^j, j \neq i \mid ip^{i,k} \in \mathcal{P}^j \wedge \nexists ar^{i,k} \in \mathcal{A}^i \mid ip^{i,k} = ar^{i,k}(p^{i,1}, \ldots, p^{i,m}). \tag{2.9}$$

Intuitivement, un paramètre $ip^{i,k}$ est un paramètre d'entrée du composant C^i si sa valeur n'est pas fixée par une relation $ar^{i,k}$ du composant C^i.

Paramètre de sortie

Un paramètre $p^{i,k}$ est un paramètre de sortie du composant C^i que l'on note $op^{i,k}$, si sa valeur résulte des mécanismes mis en œuvre par C^i. L'ensemble des paramètres de sortie associés au composant C^i est représenté par $\mathcal{OP}^i$. Formellement, le paramètre $op^{i,k}$ appartient à $\mathcal{OP}^i$ si et seulement si

$$\exists C^j, j \neq i \mid op^{i,k} \in \mathcal{P}^j \wedge \exists ar^{i,k} \in \mathcal{A}^i \mid op^{i,k} = ar^{i,k}(p^{i,1} \ldots p^{i,m}). \tag{2.10}$$

Intuitivement, un paramètre $op^{i,k}$ est un paramètre de sortie du composant C^i si sa valeur est fixée par une relation $ar^{i,k}$ du composant C^i.

L'ensemble des paramètres $\mathcal{P}^i$ d'un composant C^i se décompose donc naturellement en trois sous-ensembles disjoints deux à deux : $\mathcal{P}^i = \mathcal{IP}^i \cup \mathcal{OP}^i \cup \mathcal{PP}^i$. La figure 2.2 représente un composant C^1 qui a trois paramètres d'entrée $ip^{1,1}$, $ip^{1,2}$, $ip^{1,3}$, deux paramètres de sortie $op^{1,1}$, $op^{1,2}$ et deux paramètres privés $pp^{1,1}$ et $pp^{1,2}$. Deux relations $ar^{1,1}$ et $ar^{1,2}$ qui ne sont pas explicitées ici relient les paramètres de C^1.

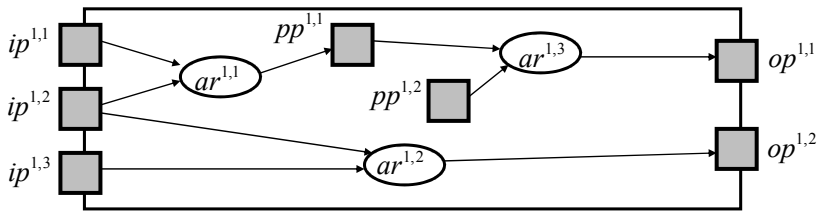

FIG. 2.2 – Modélisation d'un composant avec des paramètres

Pour illustrer les notions de paramètres et de relations sur un phénomène physique réel, prenons l'exemple simple d'un composant Re possédant une résistance r parcourue

par une intensité i et délivrant une tension u. Ces trois paramètres sont reliés par une relation algébrique $ar^{Re,1}$, la loi d'ohm :

$$\begin{aligned} &u = ar^{Re,1}(i, r), \\ &u(t) = r.i(t). \end{aligned} \tag{2.11}$$

La résistance r est un paramètre spécifique, interne au composant Re. Si le composant est conçu dans le but de fournir une tension alors u est considérée comme un paramètre de sortie de Re et i est forcément un paramètre d'entrée. Un tout autre composant pourrait avoir été conçu dans l'objectif de fournir un courant et dans ce cas i serait un paramètre de sortie et u, un paramètre d'entrée.

Considérons maintenant l'exemple d'un bac B de section s dans lequel est déversé un débit q_e. Un capteur permet de mesurer la hauteur d'eau h dans B. L'eau du bac s'écoule par un tuyau avec un débit de sortie q_s. La relation $ar^{B,1}$ entre ces paramètres est une équation différentielle :

$$\begin{aligned} &H = ar^{B,1}(q_e, q_s, s), \\ &\dot{H} = (q_e - q_s)/s. \end{aligned} \tag{2.12}$$

La section s est un paramètre privé du bac B. La hauteur d'eau h dans B est un paramètre de sortie, sa valeur est imposée par les mécanismes de B. Les débits d'eau q_e et q_s sont des paramètres d'entrée du bac B.

Dans le modèle du système à événements discrets de l'alimentation illustrée sur la figure 2.1, on peut repérer trois paramètres : interrupteur, hors_service, sortie_alimentation. Une relation logique ar^{Alim} permet de relier ces paramètres :

$$\text{sortie\_alimentation} = ar^{Alim}(\text{interrupteur, hors\_service}),$$

$$\begin{aligned} (\text{sortie\_alimentation}(t_i) = \text{courant}) \equiv ((&\text{hors\_service}(t_i) = \text{faux }) \\ &\wedge\ (\text{interrupteur}(t_i) = \text{ allume})). \end{aligned} \tag{2.13}$$

L'interrupteur est un paramètre d'entrée du système d'alimentation, le paramètre hors_service est un paramètre interne et le paramètre de sortie est bien évidemment sortie_alimentation.

2.2.2.2 Niveau système

Au niveau du système, le modèle structurel représente la manière dont sont connectés les composants du système. La figure 2.3 représente le modèle structurel d'un système complexe constitué de quatre composants : $Comps = \{C^1, C^2, C^3, C^4\}$.

Pour plus de lisibilité, les paramètres privés de chaque composant ne sont pas représentés sur ce modèle global. Cette représentation structurelle du système met bien en évidence les interactions entre composants qui sont modélisées par des paramètres

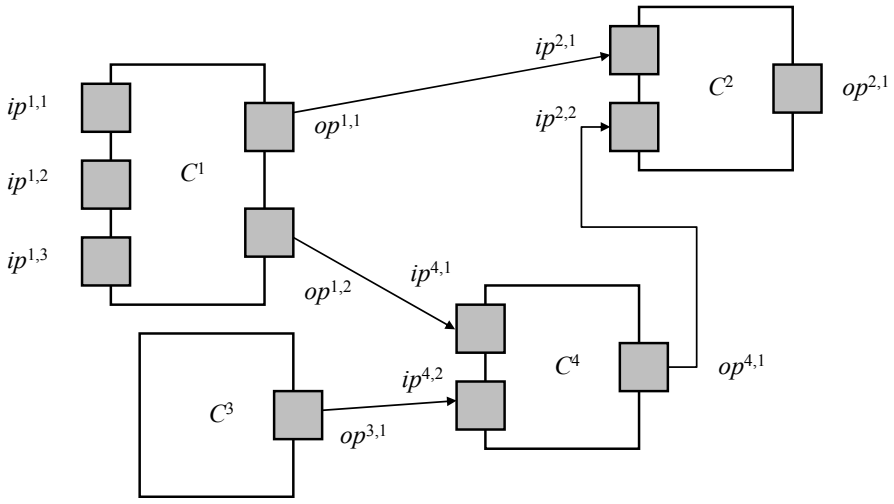

FIG. 2.3 – Modélisation structurelle d'un système complexe

partagés (paramètres d'entrée, paramètres de sortie). La notation $op^{1,1} \rightarrow ip^{2,1}$ signifie que $op^{1,1}$ est structurellement connecté à $ip^{2,1}$. Cette notation repose bien évidemment sur les objectifs de conception des différents composants en indiquant entre autre le sens du flux d'information. Les communications entre composants sont nécessaires afin d'assurer la fonction d'usage du système complexe. Ainsi, lorsque $op^{1,1} \rightarrow ip^{2,1}$, le paramètre de sortie $op^{1,1}$ du composant C^1 impose sa valeur au paramètre d'entrée $ip^{2,1}$ du composant C^2 : $ip^{2,1} = op^{1,1}$.

Lorsque le système est considéré dans sa globalité, il apparaît comme une boîte grise ayant des paramètres d'entrée et des paramètres de sortie. Les paramètres d'entrée et de sortie partagés par les composants deviennent des paramètres privés du système. L'ensemble des paramètres $\mathcal{P}$ du système Σ se décompose également en trois sous-ensembles disjoints deux à deux : $\mathcal{P} = \mathcal{IP} \cup \mathcal{OP} \cup \mathcal{PP}$. L'ensemble des paramètres d'entrée du système est noté $\mathcal{IP}$ et il est formellement défini par

$$\mathcal{IP} = \{\bigcup_{i=1}^{N} \mathcal{IP}^i\} \setminus \{ip^{j,l} | (\exists op^{i,k} \wedge op^{i,k} \rightarrow ip^{j,l})\}. \tag{2.14}$$

L'ensemble des paramètres de sortie du système est noté $\mathcal{OP}$ et il est formellement défini par

$$\mathcal{OP} = \{\bigcup_{i=1}^{N} \mathcal{OP}^i\} \setminus \{op^{i,k} | (\exists ip^{j,l} \wedge op^{i,k} \rightarrow ip^{j,l})\}. \tag{2.15}$$

L'ensemble des paramètres privés du système noté $\mathcal{PP}$ contient l'ensemble des paramètres privés des composants du système ainsi que l'ensemble des paramètres partagés

par les composants :

$$\mathcal{PP} = \{\bigcup_{i=1}^{N} \mathcal{PP}^i\} \cup \{op^{i,k} | (\exists ip^{j,l} \wedge op^{i,k} \rightarrow ip^{j,l})\}. \tag{2.16}$$

2.2.2.3 Exemple : modèle structurel d'un système de bac

On considère maintenant l'ensemble des composants qui permettent de réguler le niveau d'eau dans un bac. Le système étudié est représenté sur la figure 2.4. Un bac cylindrique B de section s_B est connecté à un tube d'évacuation T de section s_T. La hauteur d'eau h dans B est mesurée par un capteur. Le débit de sortie d'eau dans le tube T est noté q_s. Il est commandé par deux électro-vannes Va_1 et Va_2 qui sont alimentées par deux tensions v_{Va_1} et v_{Va_2}. La variable v_1 (resp. v_2) représente le statut de la vanne Va_1 (resp. Va_2). Lorsque que la vanne Va_1 (resp. Va_2) est ouverte, $v_1 = 1$ (resp. $v_2 = 1$) et lorsqu'elle est fermée $v_1 = 0$ (resp. $v_2 = 0$). L'eau qui s'écoule de T est récupérée dans un grand réservoir. Une pompe P déverse un débit q_e dans B lorsqu'elle est alimentée par une tension v_P.

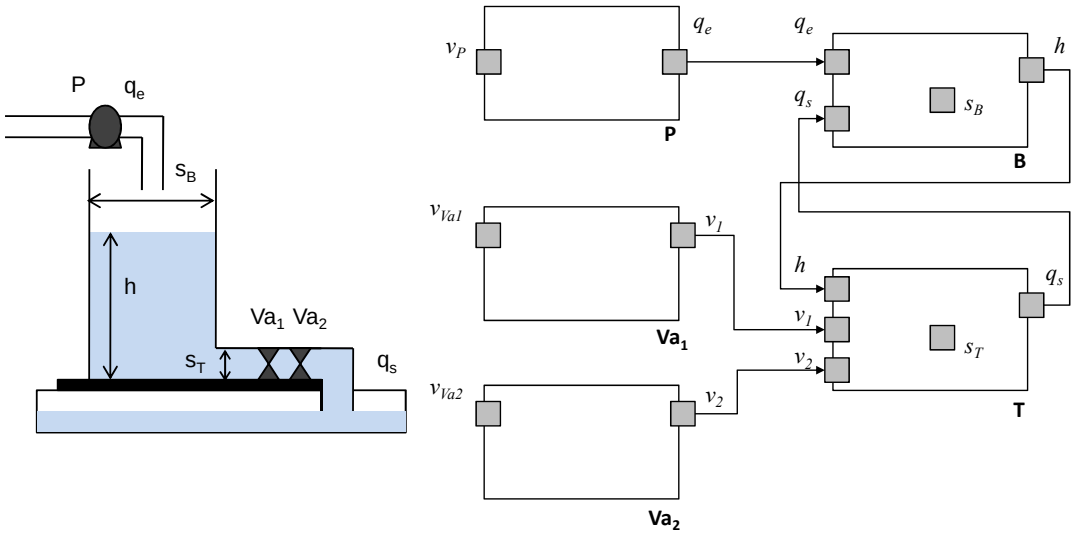

FIG. 2.4 – Modèle structurel du système de bac

Le système de régulation de niveau d'eau, noté Σ, est un ensemble de cinq composants qui interagissent : $Comps = \{P, B, T, Va_1, Va_2\}$. Chaque composant du système est modélisé par un ensemble de paramètres :

- $P : (\mathcal{IP}^P = \{v_P\}, \mathcal{OP}^P = \{q_e\})$,
- $B : (\mathcal{IP}^B = \{q_e, q_s\}, \mathcal{OP}^B = \{h\}, \mathcal{PP}^B = \{s_B\})$,
- $T : (\mathcal{IP}^T = \{h, v_1, v_2\}, \mathcal{OP}^T = \{q_s\}, \mathcal{PP}^B = \{s_T\})$,
- $Va_{i=\{1,2\}} : (\mathcal{IP}^{Va_i} = \{v_{Va_i}\}, \mathcal{OP}^{Va_i} = \{v_i\})$.

Le modèle structurel du système de régulation de niveau d'eau est illustré sur la figure 2.4. Lorsque deux composants interagissent, ils partagent au moins un paramètre sortie/entrée. Par exemple, les composants B et T partagent le paramètre h.

Le modèle structurel d'un système complexe décrit donc l'ensemble des composants du système et leurs interactions au moyen de paramètres d'entrée, de paramètres de sortie et de paramètres privés spécifiques aux composants.

2.2.3 Modélisation comportementale et fonctionnelle

Un système est généralement conçu dans l'objectif de fournir un ensemble de fonctions à son environnement. Ces fonctions sont appelées des fonctions objectifs du système. Un système complexe peut être vu comme un super-composant et défini comme un ensemble de sous-systèmes qui implémentent un ensemble de *fonctions élémentaires* contribuant à la réalisation des fonctions objectifs du système. Ces fonctions élémentaires dérivent du comportement des composants du système. Elles peuvent être modélisées à partir d'un ensemble de relations entre des paramètres.

2.2.3.1 Niveau composant

Un composant C^i fournit un ensemble de fonctions de base, que l'on appelle des fonctions élémentaires et que l'on note $\mathcal{FU}^i$. L'ensemble des fonctions élémentaires implémentées par les divers composants du système est représenté par $\mathcal{FU}_e = \bigcup_{i=1}^{N} \mathcal{FU}^i$. Ces fonctions sont associées aux composants qui les réalisent par une application Fe :

$$\begin{cases} Comps & \xrightarrow{F_e} & P(\mathcal{FU}_e) \\ C^i & \xrightarrow{F_e} & F_e(C^i) = \{Fu^{i,1}, ..., Fu^{i,m}\}, \end{cases} \tag{2.17}$$

où $P(\mathcal{FU}_e)$ représente l'ensemble des parties de $\mathcal{FU}_e$.

Les fonctions élémentaires reposent sur le comportement des composants du système et sont modélisées par des *conditions fonctionnelles* (ou contraintes) sur les paramètres des composants.

Définition 19 (Condition fonctionnelle). *Une condition fonctionnelle associée à une fonction élémentaire $Fu^{i,j} \in \mathcal{FU}^i$ est une propriété définie par une relation entre un paramètre de sortie et des paramètres d'entrée ou des paramètres privés du composant C^i.*

Une condition fonctionnelle associée à une fonction élémentaire $Fu^{i,j}$ peut s'écrire de la manière suivante :

$$Fu^{i,j} \equiv (op^{i,j} = ar^{i,j}(ip^{i,1}, \ldots, ip^{i,l_i}, pp^{i,1}, \ldots, pp^{i,r_i})) \tag{2.18}$$

où $op^{i,j} \in \mathcal{OP}^i$, $ip^{i,j} \in \mathcal{IP}^i$ et $pp^{i,j} \in \mathcal{PP}^i$.

Une fonction élémentaire est *disponible* sur un composant si la condition fonctionnelle qui lui est associée est satisfaite. Les relations entre les paramètres décrivent le modèle de bon fonctionnement (ou comportement normal) du composant tant que les fonctions restent disponibles sur le composant. Cette connaissance sur le bon fonctionnement des composants provient de la phase de conception du système (voir le paragraphe 1.3.2.3).

Si l'on reprend l'exemple du composant C^1 illustré sur la figure 2.2, les trois relations $ar^{1,1}$, $ar^{1,2}$ et $ar^{1,3}$ modélisent les conditions de deux fonctions élémentaires $Fu^{1,1}, Fu^{1,2} \in \mathcal{FU}^1$ mises en œuvre par le composant C^1 :

$$\begin{aligned} Fu^{1,1} &\equiv (op^{1,1} = ar^{1,3}(pp^{1,2}, ar^{1,1}(ip^{1,1}, ip^{1,2}))), \\ Fu^{1,2} &\equiv (op^{1,2} = ar^{1,2}(ip^{1,2}, ip^{1,3})). \end{aligned}$$

Une fonction élémentaire implémentée par un composant utilise nécessairement les ressources internes de ce composant. Toute fonction élémentaire repose donc au moins sur un paramètre privé (une ressource privée) d'un composant. Cependant il se peut que la connaissance sur ce paramètre privé ne soit pas disponible et dans ce cas, le paramètre privé n'est pas représenté dans le modèle du composant. C'est le cas pour la fonction $Fu^{1,2}$ de l'exemple sur la figure 2.2.

Les fonctions élémentaires mises en œuvre par les différents composants se composent dans le but de réaliser l'ensemble des fonctions objectif du système complexe. Cette composition des fonctions élémentaires est décrite à l'aide d'un modèle fonctionnel au niveau du système global [RIB 09].

2.2.3.2 Niveau système

Une fonction intermédiaire résulte de la composition (de l'association) de fonctions élémentaires. Les fonctions intermédiaires sont combinées dans le but de réaliser les fonctions objectif du système, c'est-à-dire ce pourquoi le système a été conçu. L'ensemble des fonctions objectif du système est noté $\mathcal{FU}_s$. L'ensemble de toutes les fonctions implémentées à partir des composants du système est représenté par $\mathcal{FU}$ et contient donc les fonctions élémentaires, les fonctions intermédiaires obtenues par les compositions et les fonctions objectif du système :

$$\mathcal{FU}_e \subseteq \mathcal{FU} \text{ et } \mathcal{FU}_s \subseteq \mathcal{FU}. \tag{2.19}$$

L'application F_s associe un ensemble de fonctions objectif $\mathcal{FU}_s$ à un système Σ :

$$\begin{cases} P(Comps) & \xrightarrow{F_s} & P(\mathcal{FU}) \\ \Sigma & \xrightarrow{F_s} & F_s(\Sigma) = \{Fu_s^1, ..., Fu_s^p\}, \end{cases} \tag{2.20}$$

où $P(Comps)$ représente l'ensemble des parties de $Comps$.

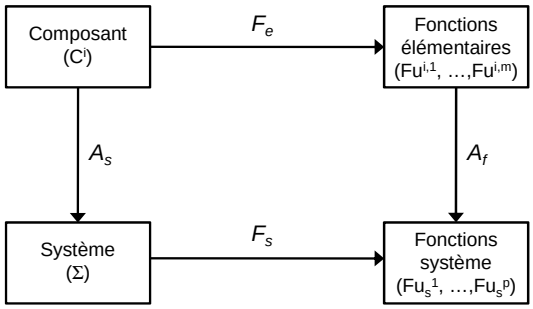

FIG. 2.5 – Système, composants et fonctions

La figure 2.5 illustre les applications F_e et F_s qui viennent d'être définies. L'application $\mathcal{A}_s$ est définie par le modèle structurel des composants du système. Un modèle fonctionnel permet de décrire l'agrégation fonctionnelle $\mathcal{A}_f$ des fonctions élémentaires des composants de $\mathcal{FU}_e$ qui contribuent à la réalisation des fonctions objectif de $\mathcal{FU}_s$ en introduisant deux applications : $Pred$ et n/m. L'application $Pred$ représente les dépendances fonctionnelles en utilisant des relations de précédence :

$$\begin{cases} \mathcal{FU} & \stackrel{Pred}{\longrightarrow} & P(\mathcal{FU}) \\ Fu^i & \stackrel{Pred}{\longrightarrow} & Pred(Fu^j) = \{Fu^k \ldots, Fu^l\} \end{cases} \tag{2.21}$$

La notation $Pred(Fu^j) = \{Fu^k, \ldots, Fu^l\}$ signifie que la réalisation de la fonction Fu^j repose sur la disponibilité d'au moins une fonction de l'ensemble $\{Fu^k, \ldots, Fu^l\}$. Les fonction de $Pred(Fu^j)$ sont en dépendance directe avec Fu^j (dépendance par transitivité). Si $Fu^k \in Pred(Fu^i)$, la fonction Fu^k est alors appelée un *prédécesseur* de Fu^j. Dans le cas où $Pred(Fu^j) = \{Fu^k\}$, la fonction Fu^j n'est réalisée que si la fonction Fu^k est disponible sur le composant. L'introduction de cette application $Pred$ permet de redéfinir une fonction élémentaire et une fonction objectif du système. Une fonction de $\mathcal{FU}$ est une fonction élémentaire $Fu^{i,j} \in \mathcal{FU}^i$ si elle n'a pas de prédécesseur dans la représentation fonctionnelle du système :

$$Pred(Fu^{i,k}) = \emptyset.$$

Une fonction élémentaire ne dépend pas de la réalisation d'une autre fonction. Une fonction de $\mathcal{FU}$ est une fonction objectif $Fu_s^i \in \mathcal{FU}_s$ si elle n'est le prédécesseur d'aucune autre fonction de $\mathcal{FU}$:

$$\forall Fu^j \in \mathcal{FU}, \quad Fu_s^i \notin Pred(Fu^j).$$

L'application $Pred$ peut être graphiquement représentée sous la forme d'un arbre, également appelé un arbre de fonctions dans [RAU 04].

La définition de l'application $Pred$ n'est cependant pas suffisamment précise dans le cas où une fonction Fu^j a plusieurs prédécesseurs, c'est-à-dire plusieurs fonctions

dans $Pred(Fu^j)$. La réalisation de la fonction Fu^j peut reposer sur la disponibilité d'une ou plusieurs fonctions de $Pred(Fu^j)$. Par exemple si $Pred(Fu^1) = \{Fu^2, Fu^3\}$, la réalisation de Fu^1 peut reposer sur la disponibilité d'une des deux fonctions Fu^2 ou Fu^3 ou bien sur la disponibilité de la totalité des fonctions dans $Pred(Fu^j)$, c'est-à-dire Fu^2 et Fu^3. Une application n/m permet de préciser le nombre de fonctions qui doivent être disponibles dans $Pred(Fu^j)$ pour la réalisation de Fu^j. Elle signifie qu'au moins n fonctions parmi m fonctions dans $Pred(Fu^j)$ sont nécessaires à la réalisation de Fu^j. Cette application permet d'exprimer les redondances fonctionnelles dans le système. L'application n/m est formellement définie de la manière suivante :

$$\begin{cases} P(\mathcal{FU}) \xrightarrow{n/m} P\left[P(\mathcal{FU})\right] \\ n/m[(Pred(Fu^j)] = \left\{X \subseteq Pred(Fu^j) \;\; tel \;\; que \;\; ||X|| = n \;\; avec \;\; 1 \leq n \leq m\right\} \end{cases} \tag{2.22}$$

Dans la notation $n/m[Pred(Fu^j)] = \{X_1 \; X_2 \; \ldots \; X_k\}$, chaque X_i est un ensemble de n fonctions dont l'interaction est suffisante pour réaliser la fonction Fu^j et un seul X_i est nécessaire pour que Fu^j soit réalisée. L'application n/m est une extension des opérateurs logiques classiques ET et OU.

- $1/m[Pred(Fu^j)] = \{\{Fu_1\} \ldots \{Fu_m\}\}$ signifie que toute fonction de $Pred(Fu^j)$ est suffisante, c'est l'équivalent de l'opérateur OU logique.
- $m/m[Pred(Fu^j)] = \{\{Fu_1, \ldots, Fu_m\}\}$ signifie qu'il n'existe aucun sous-ensemble de n fonctions ($n < m$) suffisant dans $Pred(Fu^j)$ et donc que toute fonction de $Pred(Fu^j)$ est nécessaire, c'est l'équivalent de l'opérateur ET logique.

On suppose que les ensembles $Comps$, $\mathcal{FU}$ et les applications F_e, F_s, $Pred$ font partie d'une connaissance disponible fournie par les concepteurs du système.

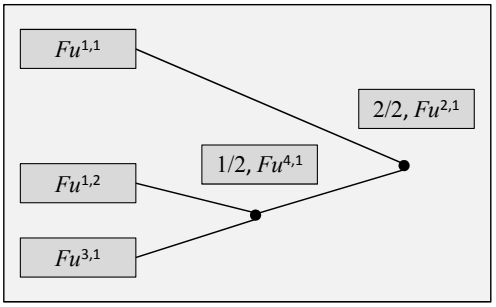

FIG. 2.6 – Modèle fonctionnel d'un système

La figure 2.6 représente un exemple de modèle fonctionnel pour le système à quatre composants illustré sur la figure 2.3. Les composants C^1 et C^2 supportent l'implémentation de trois fonctions élémentaires $Fu^{1,1}$, $Fu^{1,2}$ et $Fu^{3,1}$ qui se combinent ensuite pour réaliser la fonction intermédiaire $Fu^{4,1}$ et la fonction objectif $Fu^{2,1}$. Par abus de notation, on pourra écrire :

$$Fu^{2,1} = 2/2(Fu^{1,1}, 1/2(Fu^{1,2}, Fu^{3,1})). \tag{2.23}$$

Ces fonctions utilisent nécessairement des ressources qui sont disponibles sur les composants C_4 et C_2. Sans expliciter les relations algébriques qui décrivent le comportement des composants du système, il est possible de modéliser les fonctions élémentaires du système par

$$\begin{array}{l} Fu^{1,1} \equiv (op^{1,1} = ar^{1,1}(\{ip^{1,j}\}, \{pp^{1,k}\})), \\ Fu^{1,2} \equiv (op^{1,2} = ar^{1,2}(\{ip^{1,j}\}, \{pp^{1,k}\})), \\ Fu^{3,1} \equiv (op^{3,1} = ar^{3,1}(\{pp^{1,k}\})) \end{array} \tag{2.24}$$

où $\{ip^{1,j}\}$ représente un ensemble de paramètres d'entrée du composant C^1, $\{pp^{1,k}\}$ représente un ensemble de paramètres privés de C^1 et $\{pp^{1,k}\}$, un ensemble de paramètres privés de C^3. La fonction intermédiaire $Fu^{4,1}$ est modélisée par l'expression suivante :

$$1/2[Pred(Fu^{4,1})] = \{\{Fu^{1,2}\}\{Fu^{3,1}\}\}. \tag{2.25}$$

Cette expression décrit une redondance fonctionnelle. La fonction $Fu^{4,1}$ est réalisée si au moins une des deux fonctions élémentaires est disponible soit sur C^1 soit sur C^3. Le composant C^3 est donc redondant. Par contre, pour réaliser la fonction objectif $Fu^{2,1}$ les deux prédécesseurs doivent être disponibles :

$$2/2[Pred(Fu^{2,1})] = \{\{Fu^{1,1}, Fu^{4,1}\}\}. \tag{2.26}$$

2.2.3.3 Exemple : modèle fonctionnel d'un système de bac

Reprenons l'exemple du système Σ de bac constitué de cinq composants : $Comps = \{P, B, T, Va_1, Va_2\}$. La vanne Va_2 est un composant redondant identique au composant Va_1. La fonction objectif du système Fu^1_s est de maintenir une hauteur d'eau h comprise entre 0.4m et 0.6m dans un bac B. Afin de réaliser Fu^1_s, les composants du système doivent implémenter un ensemble de fonctions élémentaires. Les fonctions élémentaires sont disponibles sur les composants si les conditions qui leur sont associées sont vérifiées. Ces conditions reposent sur des relations entre les paramètres des composants.

Pompe La condition de la fonction élémentaire Fu^P mise en œuvre par la pompe P est modélisée par les équations suivantes :

$$Fu^P \equiv (q_e = ar^P(v_p)) \quad \text{avec } ar^P \ : \ q_e(t) = \begin{cases} v_p & si \ \ 0 < v_p < v_{max}, \\ 0 & si \ \ v_p \leq 0, \\ q_{max} & si \ \ v_p > v_{max}. \end{cases} \tag{2.27}$$

Ces équations décrivent le comportement de la pompe P. La pompe P délivre un débit q_e qui est supposé proportionnel à sa tension d'alimentation v_p. Le débit est limité et ne peut pas dépasser une certaine valeur q_{max}.

Bac La condition de la fonction élémentaire Fu^B mise en œuvre par le bac d'eau B est modélisée par l'équation de conversion suivante :

$$Fu^B \equiv (h = ar^B(q_e, q_s, s_B)) \quad \text{avec } ar^B \ : \ \dot{h} = (q_e - q_s)/s_B. \tag{2.28}$$

Vannes La condition de la fonction élémentaire Fu^{Va_i} mise en œuvre par la vanne Va_i est modélisée par l'équation suivante :

$$Fu^{Va_i} \equiv (v_i = ar^{Va_i}(v_{Va_i})) \quad \text{avec } ar^{Va_i} \ : \ v_i = \begin{cases} 0 & si \ \ v_{Va_i} \leq 0, \\ 1 & si \ \ v_{Va_i} > 1, \end{cases} \tag{2.29}$$

où v_{Va_i} est la tension qui alimente l'électro-vanne Va_i. Les deux composants Va_1 et Va_2 réalisent la même fonction élémentaire. Ce sont des composants redondants.

Tube La condition de la fonction élémentaire Fu^T mise en œuvre par le tube d'évacuation T est modélisée par l'équation de Torricelli :

$$Fu^T \equiv (q_s = ar^T(h, s_T, v_1, v_2)) \quad \text{avec } ar^T \ : \ q_s = \max(v_1, v_2) s_T \sqrt{2g \ h}. \tag{2.30}$$

Le modèle fonctionnel illustré sur la figure 2.7 décrit les associations des fonctions élémentaires implémentées par les composants pour réaliser la fonction objectif Fu_s^1 du système.

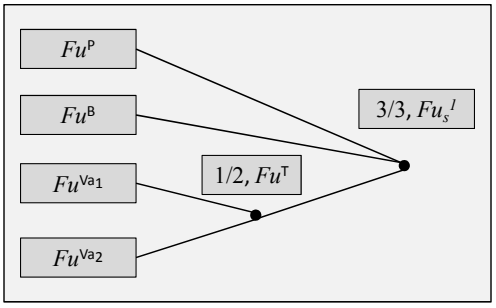

FIG. 2.7 – Modèle fonctionnel d'un système de régulation de niveau d'eau

Le modèle fonctionnel d'un système complexe définit l'ensemble des fonctions élémentaires implémentées par les composants et décrit la manière dont elles s'associent dans le but de réaliser les fonctions objectif du système. Les fonctions élémentaires sont disponibles sur les composants lorsque les conditions fonctionnelles qui leur sont associées sont vérifiées. Ces conditions reposent sur des relations entre les paramètres des composants.

2.3 Modes opérationnels pour un système complexe

Lorsque les conditions fonctionnelles ne sont plus vérifiées, les composants n'implémentent plus leurs fonctions élémentaires, ils ne fonctionnent plus correctement. La réalisation des fonctions objectif du système dépend alors du comportement de l'ensemble des composants interagissants.

La notion de *mode opérationnel* est introduite pour un composant et définie à partir de la connaissance disponible sur les comportements des composants du système complexe. Pour répondre aux problèmes de diagnostic et de pronostic, cette connaissance peut être décrite par des modèles de comportement nominal, des modèles de comportement en présence de fautes (voir le paragraphe 1.3.2.3) ou bien encore des modèles de vieillissement des composants du système (voir le paragraphe 1.4.2.1). L'ensemble des modes opérationnels peut alors être vu comme une partition de l'ensemble des comportements possibles du système.

2.3.1 Définition d'un mode opérationnel

Les modes opérationnels dépendent de l'état fonctionnel des composants du système complexe. Un mode opérationnel pour un composant est défini par le modèle qui représente l'ensemble des fonctions élémentaires disponibles sur le composant à un instant donné. Lorsque le composant C^i est dans un mode m_x^i donné, tous les paramètres du modèle de C^i sont dans leur rang et toutes les relations du modèle sont satisfaites.

Le modèle du composant C^i qui correspond au mode m_x^i est formellement défini par :

$$C_x^i = \langle \mathcal{P}^i, \mathcal{R}_x^i, \mathcal{A}_x^i \rangle \tag{2.31}$$

avec
- $\mathcal{P}^i = \{p^{i,k}\}$, l'ensemble complet des paramètres du modèle supposé fixé (c'est-à-dire qu'ils ne sont pas modifiés selon le mode du composant considéré),
- $\mathcal{R}_x^i = \{r_x^{i,k}\}$, l'ensemble des rangs pour les paramètres $\mathcal{P}^i$ dans le mode M,
- $\mathcal{A}_x^i = \{ar_x^{i,k}\}$, l'ensemble des relations définies dans le mode m_x^i.

Le composant C^i est dans le mode m_x^i si toutes les relations entre les paramètres définies dans le modèle du mode m_x^i sont satisfaites et si tous les paramètres de $\mathcal{P}^i$ appartiennent à leur rang $\mathcal{R}_x^i$ défini pour le mode m_x^i :

$$\forall k, \begin{cases} op^{i,k} = ar_x^{i,k}(ip^{i,1}, \ldots, ip^{i,l_i}, pp^{i,1}, \ldots, pp^{i,r_i}) \\ p^{i,k} \in r_x(p^{i,k}). \end{cases} \tag{2.32}$$

En s'appuyant sur la description structurelle et fonctionnelle des composants du système présentée dans la section précédente, il est possible de repérer trois types de modes opérationnels pour un composant : le mode nominal, les modes de faute et le mode anormal.

2.3.2 Mode nominal

Le mode opérationnel nominal m_n^i pour un composant C^i est défini par le modèle de comportement nominal (normal) du composant $C_n^i = \langle \mathcal{P}^i, \mathcal{R}_n^i, \mathcal{A}_n^i \rangle$, dont la connaissance provient généralement de la phase de conception. Un composant C^i est dans le mode nominal m_n^i si et seulement si tous ses paramètres $\{p^{i,k}\}$ sont dans leur rang nominal $\{r_n(p^{i,k})\}$ défini dans le modèle C_n^i et si toutes les relations $\{ar_n^{i,k}\}$ du modèle sont satisfaites :

$$\forall k, \begin{cases} op^{i,k} = ar_n^{i,k}(ip^{i,1}, \ldots, ip^{i,l_i}, pp^{i,1}, \ldots, pp^{i,r_i}) \\ p^{i,k} \in r_n(p^{i,k}). \end{cases} \tag{2.33}$$

Lorsqu'un composant est dans un mode nominal, il fonctionne correctement et fournit un ensemble de fonctions élémentaires. Les relations $\mathcal{A}_n^i$ définies dans le modèle C_n^i représentent alors les conditions fonctionnelles associées aux fonctions élémentaires implémentées par le composant C^i.

Le mode opérationnel d'un système complexe dépend directement des modes opérationnels de ses composants. Un système Σ est dans son mode nominal m_n^Σ décrit par le modèle Σ_n, lorsque tous les composants du système fonctionnent correctement et sont donc également dans leur mode nominal :

$$\forall i, C_n^i = \langle \mathcal{P}^i, \mathcal{R}_n^i, \mathcal{A}_n^i \rangle \Rightarrow \Sigma_n = \langle \mathcal{P}, \mathcal{R}_n, \mathcal{A}_n \rangle. \tag{2.34}$$

Lorsque le système est dans un mode nominal, toutes les fonctions élémentaires implémentées par les différents composants sont disponibles.

2.3.3 Mode de faute

Dans plusieurs cas, il est possible d'obtenir des modèles de comportement des composants pour un ensemble donné de fautes physiques dites anticipées (voir le paragraphe 1.3.2.3). Cette connaissance sur les fautes possibles pouvant survenir dans le système provient d'une analyse de sécurité des composants (d'une AMDE par exemple, voir le paragraphe 1.3.2.1).

Comme pour la connaissance du comportement nominal, il est possible de représenter pour un composant C^i et une faute anticipée f, un modèle correspondant au mode de faute m_f^i : $C_f^i = \langle \mathcal{P}^i, \mathcal{R}_f^i, \mathcal{A}_f^i \rangle$. Lorsque qu'une faute f est présente dans un composant C^i, les rangs des paramètres de $\mathcal{P}^i$ peuvent être différents de ceux définis pour le mode nominal m_n^i. Les relations entre les paramètres qui sont définies pour le mode nominal peuvent être aussi totalement modifiées.

L'occurrence de la faute f est due à un problème interne au composant, ce qui se traduit par l'existence d'au moins un paramètre privé du modèle du composant qui n'est plus dans son rang défini pour le mode nominal. Un composant C^i est dans un mode

de faute m_f^i si et seulement si :

$$\exists j, pp^{i,j} \notin r_n(pp^{i,j}) \quad \text{et} \quad \forall k, \begin{cases} op^{i,k} = ar_f^{i,k}(ip^{i,1}, \ldots, ip^{i,l_i}, pp^{i,1}, \ldots, pp^{i,r_i}) \\ p^{i,k} \in r_f(p^{i,k}) \end{cases} \tag{2.35}$$

Lorsqu'un composant C^i est dans un mode de faute m_f^i, il ne peut plus fournir la totalité de ses fonctions élémentaires.

Il est possible que dans un mode de faute, certaines conditions de fonctions soient toujours vérifiées, ce qui signifie que certaines fonctions élémentaires sont encore disponibles. Les relations entre les paramètres ne sont donc pas forcément toutes modifiées. Il est également possible pour certains paramètres privés du composant que le rang défini pour le mode nominal soit équivalent au rang défini pour un mode de faute. Dans un mode de faute, une action de maintenance est nécessaire afin de remplacer le composant en faute.

Le système complexe Σ est dans un mode de faute m_f^Σ s'il existe au moins un composant C^i de Σ qui est dans le mode de faute m_f^i. Il existe alors au moins une fonction élémentaire qui n'est plus disponible sur un composant du système. Les modes de fautes des composants sont définis dans le but de faire du diagnostic. Une faute f peut être souvent la cause de plusieurs défaillances, c'est-à-dire de la perte de plusieurs fonctions objectif du système.

2.3.4 Mode anormal

Lorsqu'une fonction élémentaire n'est plus réalisée par un composant, la condition fonctionnelle qui lui est associée n'est plus satisfaite. Cette condition est représentée par une relation définie dans le modèle du mode nominal du composant. Si elle n'est plus satisfaite, c'est qu'au moins un paramètre privé ou un paramètre d'entrée du composant est en dehors de son rang défini dans le modèle du mode nominal. Lorsqu'un paramètre privé est en dehors de son rang nominal, le composant est dans un mode de faute (voir le paragraphe 2.3.3). Lorsqu'un paramètre d'entrée est en dehors de son rang nominal, le composant est dans un mode anormal.

Un composant C^i est dans un mode anormal m_a^i, s'il existe au moins un paramètre d'entrée du composant qui n'est plus dans le rang nominal d'une condition associée à une fonction élémentaire :

$$\exists p^{i,j} \in \mathcal{IP}^i \mid p^{i,j} \notin r_n(p^{i,j}). \tag{2.36}$$

Par conséquent, cette condition n'est plus satisfaite et la fonction élémentaire associée n'est plus réalisée par le composant C^i. Lorsqu'un paramètre d'entrée $ip^{i,k} \in \mathcal{IP}^i$ d'un composant C^i est en dehors de son rang $r_n^i(ip^{i,k})$ défini par le modèle nominal C_n^i, c'est

qu'il existe au moins un autre composant C^j avec lequel il interagit qui ne peut pas être lui même dans son mode nominal. Le paramètre partagé de sortie $op^{j,l} \rightarrow ip^{i,k}$ est forcément en dehors de son rang nominal.

Dans certains cas, il est possible d'obtenir des information supplémentaires sur le comportement d'un composant C^i qui permettent d'établir un modèle C^i_a de ce mode anormal $m^i_a : C^i_a = \langle \mathcal{P}^i, \mathcal{R}^i_a, \mathcal{A}^i_a \rangle$.

Lorsqu'un composant C^i est dans un mode anormal m^i_a, un paramètre d'entrée $ip^{i,k}$ est en dehors de son rang nominal mais il est aussi possible qu'un paramètre privé $pp^{i,k}$ du composant soit également en dehors de son rang nominal. Cependant, la connaissance disponible sur le composant C^i ne nous permet pas de déterminer s'il est dans un mode de faute. Un mode anormal peut donc masquer un mode de faute. Une action de maintenance peut être décidée pour un composant dans un mode anormal mais cette décision n'est pas nécessairement optimale. Les informations disponibles sur le comportement du composant sont insuffisantes pour déterminer si le composant est réellement responsable de la défaillance.

D'après ces définitions, un ensemble de modes opérationnels peut être associé à un composant C^i :

$$\mathcal{M}^i = \{m^i_n, m^i_a, m^i_{f1}, ..., m^i_{fk}\},$$

dans lequel chaque mode m^i_x est caractérisé par un modèle spécifique C^i_x si la connaissance relative au comportement des composants du système est disponible. A un instant donné, un système ne peut bien évidemment être que dans un seul mode.

2.3.5 Exemple : modes opérationnels d'un système de bac

Le comportement des composants est décrit par les équations qui modélisent les conditions fonctionnelles associées aux fonctions élémentaires du système (voir le paragraphe 2.2.3.3, page 55). Ces équations représentent les relations du modèle nominal lorsque les fonctions sont disponibles sur les composants. Selon le rang des valeurs des paramètres privés impliqués dans ces équations, ces relations modélisent les modes opérationnels des composants.

Des paramètres privés sont rajoutés aux modèles des composants du système de manière à modéliser des fautes qui pourraient apparaître dans chacun des composants. Le modèle structurel qui représente l'ensemble des paramètres des composants est représenté sur la figure 2.8.

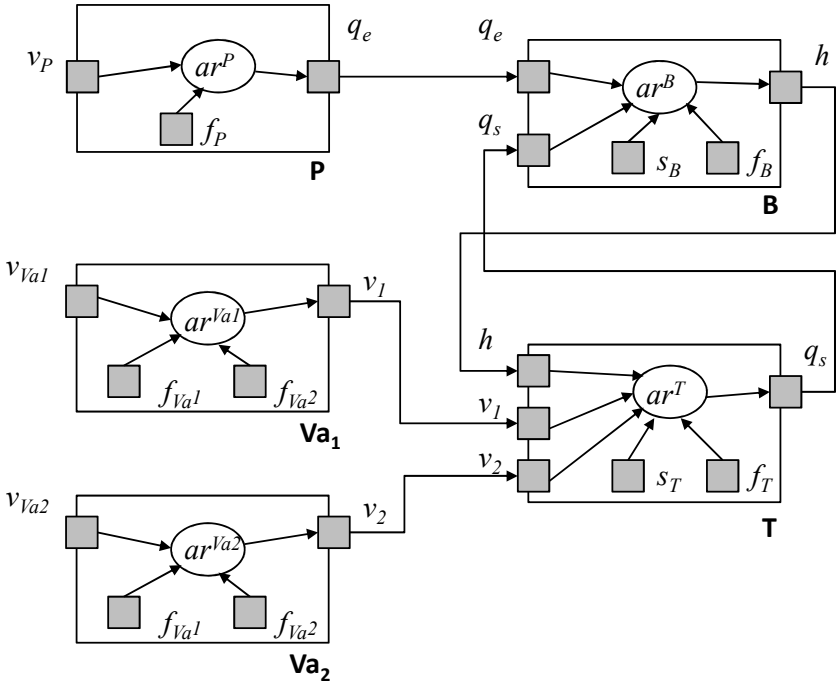

FIG. 2.8 – Modèle structurel d'un système de bac

Pompe Le comportement de la pompe P est modélisé par les équations suivantes :

$$ar^P \ : \ q_e(t) = \begin{cases} v_p(1-f_P) & si \ \ 0 < v_P < q_{max}, \\ 0 & si \ \ v_P \leq 0, \\ q_{max}(1-f_P) & si \ \ v_P > q_{max}. \end{cases} \tag{2.37}$$

Lorsque P est dans le mode nominal m_n^P, la fonction Fu^P est disponible, et le paramètre privé f_P est dans son rang nominal : $r_n^P(f_P) = \{0\}$. On retrouve bien les équations correspondant à la condition fonctionnelle de Fu^P. Lorsque la pompe P est dans un mode de faute m_f^P, elle ne délivre plus aucun débit même si sa tension d'alimentation v_P n'est pas nulle. Dans ce cas, le paramètre privé f_P est dans son rang défini pour le mode de faute m_f^P : $r_f^P(f_P) = \{1\}$.

Bac Le comportement du bac B est modélisé par l'équation suivante :

$$ar^B \ : \ \dot{h} = (q_e - q_s - f_B)/s_B. \tag{2.38}$$

Le mode de faute m_f^B décrit le comportement de B lorsqu'il y a une fuite d'eau dans le bac. Cette fuite est modélisée par le paramètre privé f_B et son rang défini pour le modèle de faute m_f^B est $r_f^B(f_B) = [\varepsilon, q_s]$. Lorsque B est dans son mode nominal, il n'y a pas de fuite dans le bac et $r_n^B(f_B) = \{0\}$.

Vannes Le comportement des vannes Va_i est modélisé par l'équation suivante :

$$ar^{Va_i} \ : \ v_i = \begin{cases} 0 + f_{Va^1} & si \ \ v_{Va_i} \leq 0, \\ 1 - f_{Va^2} & si \ \ v_{Va_i} > 1. \end{cases} , \tag{2.39}$$

où v_i représente le statut de la vanne : ouverte ou fermée. Lorsque les vannes Va_i sont dans le mode nominal $m_n^{Va_i} : r_n^{Va_i}(v_i) = \{0, 1\}$, $r_n^{Va_i}(f_{Va^1}) = r_n^{Va_i}(f_{Va^2}) = \{0\}$. Le paramètre privé f_{Va^1} permet de modéliser le mode de faute $m_{f1}^{Va_i}$ dans lequel la vanne Va_i est restée bloquée en position ouverte, son rang défini dans le modèle du mode de faute $m_{f1}^{Va_i}$ est donc $r_{f1}^{Va_i}(f_{Va^1}) = \{1\}$. Le paramètre privé f_{Va^2} permet de représenter le mode de faute $m_{f2}^{Va_i}$ dans lequel la vanne Va_i est restée bloquée en position fermée, dans ce cas $r_{f2}^{Va}(f_{Va^2}) = \{1\}$.

Tube Le comportement du tube T est modélisé par l'équation suivante :

$$ar^T \ : \ q_s = \max(v_1, v_2) s_T \sqrt{2gh} - f_T. \tag{2.40}$$

Lorsque le tube est dans son mode nominal $m_n^T : r_n^T(f_T) = \{0\}$. Le paramètre privé f_T permet de modéliser un mode de faute pour le tube T en représentant la présence d'une fuite de liquide dans le tube T. Dans ce mode de faute m_f^T, le rang du paramètre de fuite est différent de zéro, $r_f^T(f_T) = [\varepsilon, q_s]$, avec $\varepsilon > 0$.

2.4 Conclusion

Nous avons défini un cadre de modélisation générique pour représenter de manière homogène la connaissance disponible pour chaque composant d'un système complexe. Un modèle structurel et un modèle fonctionnel ont été introduits pour représenter l'ensemble des composants du système, leurs interactions et les fonctions qu'ils mettent en œuvre. Ces deux modèles reposent sur un ensemble de paramètres caractéristiques des composants ainsi qu'un ensemble de relations entre ces paramètres.

A partir de cette modélisation générique pour un système complexe, il est possible de définir des modes de fonctionnement du système que l'on appelle des modes opérationnels. Selon la disponibilité des fonctions du système, le système peut être dans un mode nominal, un mode anormal ou un mode de faute.

3 Caractérisation d'une architecture générique de supervision

Résumé : Une architecture générique de supervision pour la maintenance est proposée. Elle intègre un module de diagnostic et un module de pronostic afin de surveiller et d'analyser l'état du système complexe pour aider à la prise de décisions d'actions de maintenance. Afin de faciliter la communication entre les modules de l'architecture, le formalisme unificateur s'appuyant sur les notions de paramètres et de modes opérationnels introduites dans le chapitre précédent est utilisé pour établir un couplage des problèmes de diagnostic et de pronostic. Il permet de caractériser un couplage des problèmes de diagnostic et de pronostic.

3.1 Introduction

Les nouvelles technologies embarquées permettent de développer et de mettre en place une architecture de supervision capable de surveiller les composants d'un système complexe et de détecter en ligne les éventuels problèmes ou les pannes pouvant survenir sur les composants du système. L'architecture de supervision doit intégrer des capacités de diagnostic et de pronostic dans le but d'améliorer la maintenance préventive. Un système complexe résulte d'un assemblage de composants hétérogènes. Dans le chapitre précédent, nous avons proposé un moyen de représenter de manière homogène la connaissance sur chaque composant. Cette modélisation générique d'un système complexe constitue la base de connaissance commune aux fonctions de diagnostic et de pronostic intégrées dans l'architecture de supervision.

Ce chapitre présente une architecture de supervision générique qui couple un module de diagnostic et un module de pronostic dans le but d'analyser de manière précise et complète l'état de santé du système complexe. Le formalisme qui repose sur un ensemble de modes opérationnels pour le système (un ensemble de paramètres, de rangs et de relations entre ces paramètres), permet de caractériser un couplage des problème de diagnostic et de pronostic.

3.2 Architecture de supervision pour la maintenance

Un système complexe est un ensemble de composants interconnectés qui communiquent entre eux dont la description est précisément donnée dans le chapitre 2. Nous présentons une architecture de supervision générique qui tient compte de l'hétérogénéité des composants en fournissant une même représentation de diagnostic et de pronostic pour chaque composant (ou ensemble de composants) du système complexe.

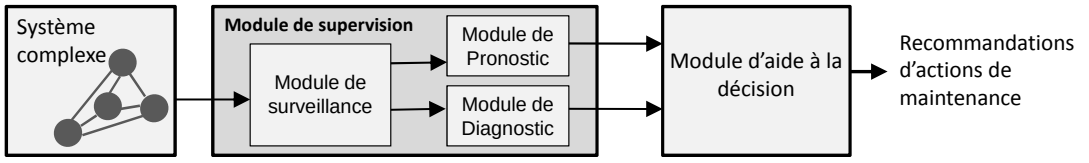

FIG. 3.1 – Architecture de supervision et de maintenance

L'architecture de supervision que nous proposons est décrite par un diagramme de type SADT (*Structured Analysis and Design Technic*) illustré sur la figure 3.1. Cette représentation graphique standardisée fournit une vue fonctionnelle des différents modules de l'architecture en indiquant par des flèches les données qui sont échangées. L'architecture est principalement composée de deux modules. Un macro-module de supervision a pour but de surveiller et d'analyser l'état du système tout au long de son fonctionnement. Il fournit des informations de diagnostic et de pronostic à un module d'aide à la décision dont la tâche est de déterminer des actions de maintenance appropriées [WIL 04].

Le macro-module de supervision se décompose en trois sous-modules. L'objectif et le contenu de chacun des sous-modules est décrit ci-dessous. On détermine la connaissance requise pour chaque sous-module de l'architecture ainsi que la forme du résultat qu'il fournit.

3.2.1 Module de surveillance

Le module de surveillance encapsule une fonction de surveillance dont l'objectif est d'élaborer et de mettre à disposition des informations structurées sur la situation du système (état de santé) que l'on observe au moyen de capteurs.

La fonction de surveillance met en œuvre des mécanismes d'observation, de détection et de filtrage pour générer des indicateurs pertinents à partir des informations enregistrées par les capteurs (voir le paragraphe 1.3.1). Ces informations issues des capteurs sont appelées des observations en ligne du système. Dans notre formalisme, une observation est une mesure d'un paramètre. Cependant, on notera indifféremment le paramètre et sa valeur par $pp^{i,k}$. Dans notre cas, les observations OBS correspondent donc à un ensemble de paramètres du système dont les valeurs peuvent être mesurées :

$OBS \subseteq \mathcal{P}$. Les observations propres à un composant C^i sont appelées des observations locales et notées $OBS^i : OBS^i \subseteq OBS$ tel que $OBS^i = \{p^{i,j}, \ldots, p^{i,k}\}$.

Il fournit les observations en ligne nécessaires aux modules de diagnostic et de pronostic. Il contient donc l'ensemble des capteurs disponibles sur les composants du système à surveiller ainsi que l'ensemble des protocoles de communication entre ces capteurs, les indicateurs générés et les autres modules de l'architecture de supervision.

3.2.2 Module de diagnostic

Pour de grands systèmes complexes, il est très difficile de raisonner de manière globale afin d'obtenir un diagnostic de panne sur la totalité du système. On utilise alors un ensemble de modules de diagnostic que l'on appelle des modules de diagnostic local. Un module de diagnostic local réalise un diagnostic de faute pour un composant (voir la définition 18, page 45) ou pour un ensemble de composants du système, c'est-à-dire un sous-système.

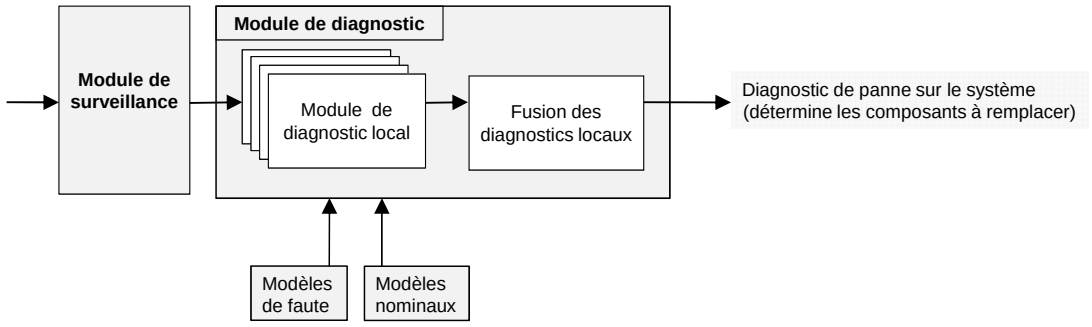

FIG. 3.2 – Module de diagnostic

Un module de diagnostic local pour un sous-système γ récupère les observations en ligne disponibles à partir des capteurs positionnés sur les composants du sous-système γ. Ce sont des observations locales du sous-système γ que l'on note OBS^γ. Une connaissance sur le comportement et les différents modes de fonctionnement m_x^i des composants C^i de γ est également disponible (modes opérationnels, voir le paragraphe 2.3, page 56). Un module de diagnostic local repose sur des modèles de comportement en mode nominal m_n^i des composants et des modèles des comportements en présence de fautes $m_{f_k}^i$ comme le montre la figure 3.2. Les modèles de comportement nominal proviennent généralement d'une spécification du système. Les modèles de faute sont obtenus à partir d'une analyse des modes de défaillance et de leurs effets (AMDE, page 14) [VIL 88] [ZWI 95].

Une méthode de diagnostic classique à base de modèles peut être alors utilisée (voir le paragraphe 1.3.2.3, page 19). On rappelle l'objectif du diagnostic classique à base

de modèles qui consiste à affecter des modes à des composants à partir d'observations délivrées par le module de surveillance et des modèles de comportement des composants du système [REI 87].

Un résultat de diagnostic local Δ^γ pour un sous-système γ est un ensemble de candidats H_i qui expliquent les observations locales OBS^γ, c'est-à-dire les observations qui proviennent uniquement du sous-système γ, en incriminant un ou plusieurs composants C^i de γ susceptibles d'être en faute :

$$\Delta^\gamma = H_1 \vee H_2 \vee \ldots \vee H_n. \tag{3.1}$$

Chaque candidat de diagnostic local H_i est une hypothèse cohérente avec les observations locales OBS^γ. La section 3.4 caractérise formellement le problème de diagnostic et définit de manière plus précise un résultat de diagnostic à partir du formalisme introduit dans le chapitre 2.

On considère que tout composant du système est couvert au moins par un module de diagnostic local. On dispose alors d'un ensemble de diagnostics locaux. Un diagnostic local est construit à partir d'observations locales, il est indépendant des autres observations du système.

Certains candidats proposés par les diagnostics locaux peuvent ne pas être compatibles avec l'ensemble des observations OBS du système. Pour établir le diagnostic global du système complexe Σ, les résultats des diagnostics locaux doivent être fusionnés à l'aide d'une stratégie de cohérence globale. Une stratégie de cohérence globale consiste à éliminer les candidats incompatibles avec les observations du système [PEN 02]. Le module de fusion vérifie si les candidats proposés par les différents diagnostics locaux sont globalement cohérents (c'est-à-dire s'ils sont compatibles avec les modes déterminés pour l'ensemble des composants, voir les définitions 23 et 24) et fournit un diagnostic en ligne sur le système global en déterminant les composants en faute qu'il faut remplacer.

3.2.3 Module de pronostic

De même que pour le diagnostic, il existe plusieurs modules de pronostic local qui réalisent en ligne un pronostic de défaillance pour un composant ou un ensemble de composants, un sous-système. L'objectif de ces modules de pronostic est de déterminer la durée de vie résiduelle (RUL) des composants du système.

Un module de pronostic local pour un sous-système γ utilise les observations en ligne locales OBS^γ délivrées par le module de surveillance et une connaissance sur la manière dont vieillissent les composants du sous-système γ. Un composant qui vieillit voit ses paramètres internes dévier de leur spécification nominale. Ces déviations de paramètres ont donc des conséquences sur le comportement des composants et les fonctions qu'ils implémentent. Elle peuvent provoquer des fautes dans le composant si les paramètres

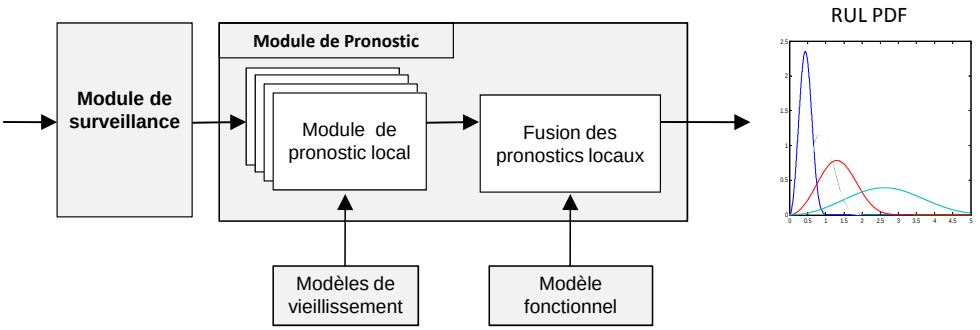

FIG. 3.3 – Module de pronostic

évoluent en dehors de leur rang nominal. Un module de pronostic local pour un sous-système γ repose donc sur un ensemble de lois de vieillissement, que l'on suppose fournies par les concepteurs des composants. Une loi de vieillissement (également appelé loi de dégradation) décrit la manière dont vieillit un paramètre privé $pp^{i,k}$ d'un composant C^i. Une loi de vieillissement peut dépendre des facteurs de stress qui sont mesurés, il s'agit alors d'un modèle d'estimation ou d'un modèle physique. Dans les cas où les facteurs de stress ne sont pas mesurés, des modèles de vieillissement statistiques sont utilisés [BRO 00] [ROE 05] (voir la section 1.4.2, page 30). Quels que soient les modèles de vieillissement disponibles et le type de composant considéré, un module de pronostic local pour un sous-système γ détermine le vieillissement des paramètres privés $pp^{i,k}$ des composants C^i de γ en terme de probabilité de faute, comme dans les études de fiabilité décrites dans la section 1.4.2.1.

La figure 3.3 représente l'ensemble des connaissances utilisées par les modules de pronostic local. Un module de fusion récupère les probabilités de faute des paramètres privés des composants fournies par les différents modules de pronostic locaux et les compose à l'aide du modèle fonctionnel du système complexe Σ (voir le paragraphe 2.2.3.2, page 52). Il fournit un pronostic du système global en obtenant des probabilités de défaillance pour chaque fonction mise en œuvre par le système.

Le problème de pronostic est formellement caractérisé dans la section 3.5. Il s'appuie également sur le formalisme défini dans le chapitre 2. La méthode utilisée pour représenter un résultat de pronostic local est précisément décrite dans le chapitre 4, ainsi que la manière dont sont fusionnés les résultats de pronostic local.

3.2.4 Module d'aide à la décision de maintenance

Le module d'aide à la décision fournit des recommandations de maintenance pour la totalité du système Σ. Ces recommandations de maintenance concernent le remplacement ou le retrait d'un ou plusieurs composants du système auxquels le diagnostic a affecté un mode de faute ou pour lesquels le pronostic prévoit l'apparition d'un mode de

faute. Une décision d'action de maintenance tient compte des candidats de diagnostic déterminés par le module de diagnostic et des probabilités de défaillance du système Σ fournies par le module de pronostic.

Une décision d'action de maintenance repose également sur une mission que le système doit accomplir et sur des critères de risque et de coût comme le montre la figure 3.4. La future mission d'un système est un objectif (ou plusieurs objectifs) que le système Σ doit atteindre avant la prochaine phase de maintenance programmée. Un critère de risque est associé à chaque faute détectée par la fonction de diagnostic ou chaque défaillance prévue par la fonction de pronostic selon l'impact qu'elle peut avoir sur l'accomplissement de la mission du système. Un critère économique est également associé à chacune de ces fautes détectées ou défaillances prévues selon le coût de la réparation qu'elles engendrent dans le cas où le système tombe en panne avant la prochaine phase de maintenance [RIB 08].

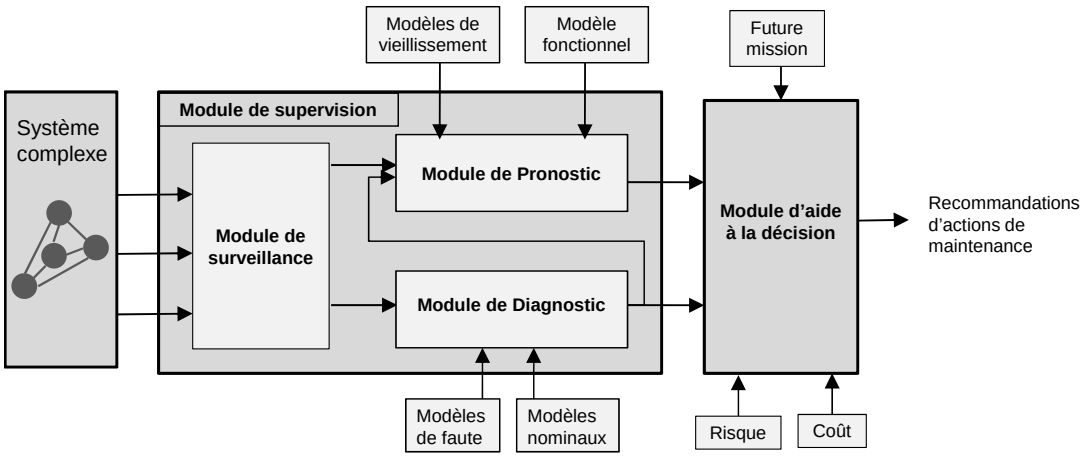

FIG. 3.4 – Architecture de supervision et connaissances requises

L'architecture générique de supervision et les différentes connaissances requises pour les modules de diagnostic et de pronostic sont rappelées sur la figure 3.4. La modélisation structurelle et fonctionnelle du système complexe ainsi que les modes opérationnels définis pour ses composants (mode nominal, modes de faute) constituent une connaissance commune aux modules de diagnostic et de pronostic.

3.3 Du problème de diagnostic au problème de pronostic

Un lien entre les modules de diagnostic et de pronostic peut être repéré sur la figure 3.4. En effet, le module de diagnostic peut fournir des informations importantes sur l'état de fonctionnement (état de santé) du système qui seront utilisées par le module de pronostic afin de mettre à jour, d'adapter les probabilités de défaillance du

système. Pour expliciter ce lien diagnostic-pronostic, il est nécessaire de trouver une représentation générique de ces deux problèmes en s'appuyant sur le formalisme introduit dans le chapitre 2. Cette représentation générique permettra de caractériser un couplage diagnostic-pronostic.

3.3.1 Séquence de modes du système

Le comportement d'un système peut être représenté par une succession de modes opérationnels (ou modes de fonctionnement, section 2.3, page 56) à partir de t_0 qui correspond à la date de mise en service du système et jusqu'à t_p, la date à laquelle il tombe en panne, c'est-à-dire qu'il ne peut plus assurer la totalité de ses fonctions objectifs.

Un mode de faute pour un système complexe est un mode dans lequel au moins un paramètre privé d'un de ses composants est en dehors de son rang défini par le modèle du mode nominal Σ_n. Lorsque le système est dans un mode de faute, une ou plusieurs fonctions élémentaires ne sont plus disponibles sur les composants. Un mode de faute devient un mode de panne pour le système lorsque les fonctions élémentaires perdues ne permettent plus de réaliser l'ensemble des fonctions objectifs du système. Le système global est alors dit défaillant (voir la définition 2) et aucune évolution vers un autre mode opérationnel n'est possible sans une action de maintenance.

Par conséquent, durant sa vie opérationnelle, c'est-à-dire entre deux phases de maintenance corrective, un système Σ suit une trajectoire de modes

$$(mr^{\Sigma}_{n,0},\ mr^{\Sigma}_{f_1,1},\ \ldots,\ mr^{\Sigma}_{f_p,p}). \tag{3.2}$$

La notation $mr^{\Sigma}_{x,j}$ est introduite pour différencier les modes réels des modes $m^{\Sigma}_{x,j}$ déterminés par la fonction de diagnostic pour le système Σ. On représente donc par $mr^{\Sigma}_{n,0}$, le fait que le système Σ est en réalité dans son mode normal (nominal) à l'instant t_0 et $mr^{\Sigma}_{f_j,j}$, le fait que le système Σ est réellement dans le mode de faute $m^{\Sigma}_{f_j}$ à l'instant t_j. Notons t_{j+1}, la date à laquelle le système quitte le mode $mr^{\Sigma}_{f_j,j}$ pour le mode $mr^{\Sigma}_{f_{j+1},j+1}$. La séquence de modes d'un système du début à la fin de sa vie est illustrée sur la figure 3.5.

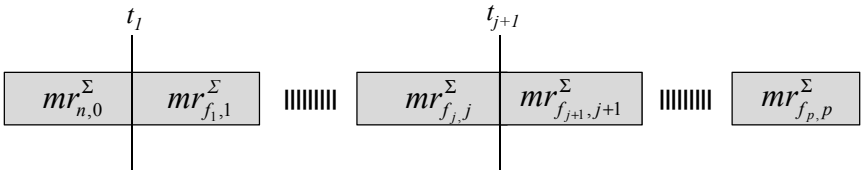

FIG. 3.5 – Séquence de modes d'un système Σ

Au début de son utilisation, le système Σ est dans son mode nominal $mr^{\Sigma}_{n,0}$: tous les paramètres du système sont dans leur rang défini par le modèle nominal Σ_n. Le

mode $mr^{\Sigma}_{f_1,1}$ est un mode de faute simple dans lequel un seul paramètre privé est en dehors de son rang nominal. Les modes qui suivent sont des modes de fautes multiples dans lesquels plusieurs paramètres privés des composants sont en dehors de leur rang nominal. Le dernier mode de faute $mr^{\Sigma}_{f_p,p}$ représente un mode de panne du système dans lequel une fonction objectif n'est plus assurée. Un mode de panne est donc associé à la perte d'une fonction objectif.

La plupart des changements de modes sont dûs à l'occurrence de fautes sur les composants du système mais certains peuvent être provoqués par des actions de maintenance. Ce dernier cas sera étudié dans le prochain paragraphe.

Cette notion de séquence de modes pour un système Σ permet d'introduire l'automate des modes d'un système à partir duquel le problème de pronostic peut être défini comme une extension du problème de diagnostic.

3.3.2 Automate des modes d'un système

Un automate des modes pour un système complexe s'appuie sur les modèles disponibles des composants du système pour représenter l'ensemble des séquences de modes qu'il pourrait suivre dès sa mise en service à t_0. Cet automate des modes est illustré sur la figure 3.6. Le but n'est pas de construire l'automate des modes d'un système. On sait simplement qu'il en existe un pour tout système, et que dans la réalité, le système suit effectivement une séquence de modes de cet automate.

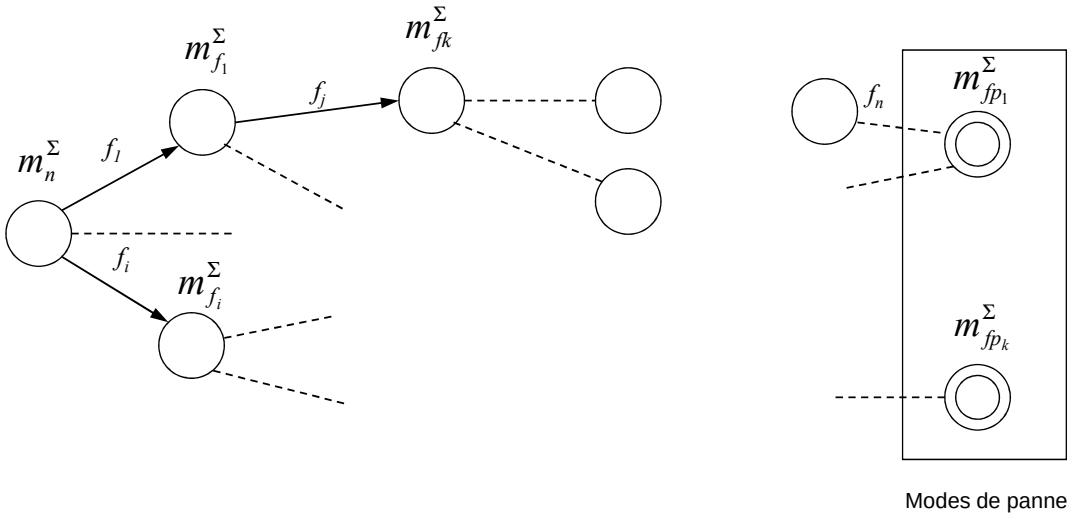

FIG. 3.6 – Automate des modes du système

L'automate décrit les différentes possibilités d'évolution des modes des composants d'un système complexe Σ tout au long de sa vie opérationnelle, c'est-à-dire à partir de sa mise en service à t_0, quand il est supposé être dans son mode nominal $m^{\Sigma}_{n,0}$, jusqu'à sa

défaillance à t_p, lorsqu'il est dans un mode de panne $m^{\Sigma}_{f_p,p}$. L'occurrence d'une faute f_i sur un des composants du système Σ fait évoluer le mode du système qui est représenté par les états de l'automate des modes. Une faute f_i est considérée comme un événement non observable.

Définition 20 (Automate des modes). *L'automate des modes d'un système complexe Σ est défini par le quintuplet (X, E, T, x_0, X_f) dans lequel*

- *X est un ensemble fini d'états correspondant aux modes du système Σ tout au long de sa vie opérationnelle. Le mode du système provenant directement des modes de ses composants, un état $x_i \in X$ représente alors une affectation de mode aux N composants du système : $x_i = m^{\Sigma}_x$ tel que $m^{\Sigma}_x = \langle m^1_y, \dots m^N_z \rangle$.*
- *E est un ensemble d'événements de faute $\{f_1, \dots, f_k\}$ pouvant survenir sur les composants du système Σ.*
- *T est la fonction de transition définie par $T : X \times E \to X$.*
- *x_0 est l'état initial de l'automate qui correspond au mode nominal $m^{\Sigma}_{n,0}$ dans lequel se trouve le système au début de son utilisation à t_0.*
- *X_f est un ensemble fini d'états finaux (ou terminaux) qui correspondent aux modes de panne du système Σ.*

Lorsque le système quitte son mode nominal m^{Σ}_n, il est dans un mode de faute simple (par exemple $m^{\Sigma}_{f_1}$ ou $m^{\Sigma}_{f_i}$). Ce qui signifie qu'un seul paramètre privé est en dehors de son rang nominal et que donc un seul composant du système est dans un mode de faute. Les modes de faute simples sont suivis des modes de fautes multiples (par exemple $m^{\Sigma}_{f_k}$) dans lesquels un ou plusieurs composants du système peuvent être dans un mode de faute.

Un système complexe Σ doit fournir un ensemble de fonctions objectifs $\mathcal{FU}_s$ à son environnement. Lorsqu'au moins une de ses fonctions n'est plus réalisée, le système est dans un mode de panne. Un mode de panne $m^{\Sigma}_{fp_k}$ est donc associé à la perte d'au moins une fonction objectif Fu_{s_j}. Il est représenté dans l'automate des modes par un état final $x \in X_f$. Un état final est un état définitif à partir duquel aucune évolution vers un autre mode du système n'est possible. Une action de maintenance corrective doit alors être décidée.

La plupart des changements de modes sont dûs à l'occurrence de fautes sur les composants du système. Des actions de maintenance telles que la réparation ou le remplacement d'un composant du système peuvent également modifier son mode opérationnel. Dans l'automate des modes d'un système, une action de maintenance (intervention) fait revenir le système dans un état dans lequel il est déjà passé. Un automate des modes étendu est défini et prend en compte les actions de maintenance possibles qui permettent de faire évoluer le mode opérationnel du système. Un exemple d'automate des modes étendu pour un système complexe Σ est illustré sur la figure 3.7.

DÉFINITION 21 (Automate des modes étendu). *L'automate des modes étendu d'un système complexe Σ est défini par le quintuplet $(X^m, E^m, T^m, x_0^m, X_f^m)$ dans lequel*

- *X^m est un ensemble fini d'états représentant les modes du système Σ. Un état $x_i^m \in X^m$ est une affectation de mode aux N composants du système : $x_i^m = m_x^\Sigma$ tel que $m_x^\Sigma = \langle m_y^1, \dots m_z^N \rangle$.*
- *E^m est un ensemble d'événements de faute et de réparation : $E^m = \{f_1, \dots, f_k\} \cup \{Int(C^1), \dots, Int(C^N)\}$, où $Int(C^N)$ est l'événement associé à l'action de maintenance qui consiste à réparer le composant C^N.*
- *T^m est la fonction de transition définie par $T : X \times E \to X$.*
- *x_0^m est l'état initial de l'automate qui correspond au mode nominal $m_{n,0}^\Sigma$ dans lequel se trouve le système au début de son utilisation à t_0.*
- *X_f^m est un ensemble fini d'états finaux (ou terminaux) qui correspondent aux modes de panne du système Σ.*

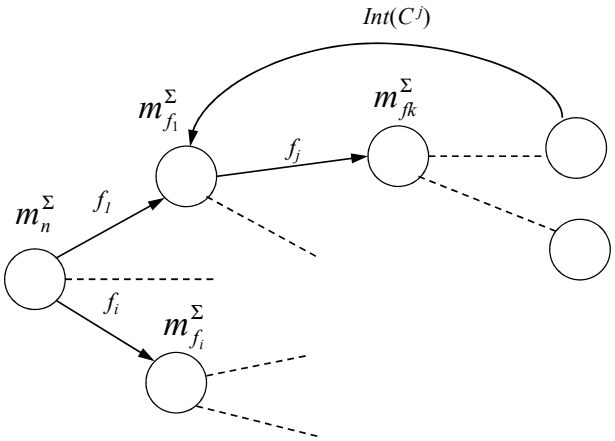

FIG. 3.7 – Automate des modes étendu du système Σ (avec interventions)

L'événement $Int(C^j)$ représente l'intervention de maintenance qui consiste à remplacer le composant C^j et permet de rétablir le système dans un mode $m_{f_1}^\Sigma$. Ces actions de maintenance sont considérées dans l'automate des modes comme des événements contrôlables et observables. Un rapprochement peut être fait avec la commande supervisée, l'objectif étant d'éviter d'arriver dans un état de panne [RAM 87].

3.3.3 Extension du diagnostic pour le pronostic

L'automate des modes permet d'introduire les problèmes de diagnostic et de pronostic. Il permet également de définir le problème du pronostic comme une extension du problème de diagnostic comme le montre la figure 3.8.

Le diagnostic d'un système consiste à déterminer la séquence de modes passés du système et le pronostic consiste à prédire la séquence de modes futurs du système.

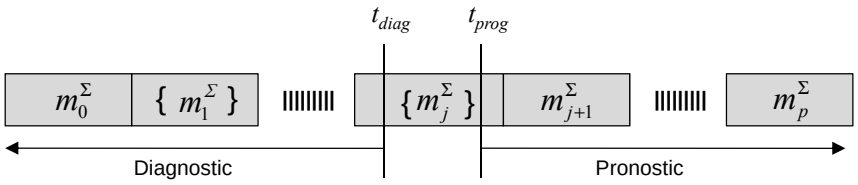

FIG. 3.8 – Diagnostic et pronostic d'un système Σ

La notation m_j^Σ représente le mode du système Σ au temps t_j. Dans le cas idéal où le système est totalement diagnosticable (voir le chapitre 5.2.1), le diagnostic à un instant t_{diag} détermine exactement la séquence de modes qui a eu lieu : $\forall t_j \in [t_0, t_{diag}]$, $m_j^\Sigma = mr_j^\Sigma$, où mr_j^Σ représente le mode dans lequel est en réalité le système Σ à l'instant t_j. Il n'y a alors aucune ambiguïté sur le mode du système depuis sa mise en service à t_0 jusqu'à t_{diag}. Ce cas est cependant très rare en pratique, le diagnostic détermine alors l'ensemble des séquences possibles de modes passés qui sont cohérentes avec les modèles du système et l'ensemble des observations. Le diagnostic détermine les séquences de l'automate des modes du système de t_0 à t_{diag} qui sont cohérentes avec l'ensemble des observations en ligne. Le système est supposé être dans son mode nominal à t_0, mais dès l'apparition d'une faute sur un composant, il peut y avoir plusieurs hypothèses possibles $\{m_1^\Sigma\}$ concernant le mode du système déterminé par la fonction de diagnostic.

Il ne sera jamais possible de prédire l'unique séquence de modes futurs du système car la séquence pronostiquée n'a pas encore eu lieu. Le pronostic à un instant t_{prog} consiste à estimer la séquence de modes futurs $(m_{j+1}^\Sigma, \ldots, m_p^\Sigma)$ qui aura la plus grande probabilité de se produire selon les modèles de vieillissement disponibles pour les composants du système (voir la section 3.5). Une fonction de pronostic présentée dans le chapitre 4 estime le prochain mode de faute en fonction des modèles de vieillissement des composants.

L'ensemble des séquences déterminées par le diagnostic et prédites par le pronostic à partir des observations disponibles peut être représenté comme une sous-partie de l'automate des modes.

3.4 Caractérisation du problème de diagnostic

L'objectif du diagnostic à base de modèles est de déterminer le mode opérationnel de chaque composant du système complexe qui est cohérent avec l'ensemble complet

des observations (valeurs des capteurs, indicateurs de défaillances ...) fournies par la fonction de surveillance et le(s) modèle(s) du système (modèle de comportement nominal, modèles de fautes))[HAM 92]. Des diagnostics locaux sont d'abord réalisés au niveau des composants. Ils sont ensuite fusionnés en tenant compte des interactions et des dépendances fonctionnelles entre les composants dans le but de fournir un diagnostic global pour le système. La caractérisation du diagnostic se fait un deux temps : dans un premier temps, une caractérisation atemporelle du diagnostic est proposée, puis une caractérisation temporelle du diagnostic est présentée.

3.4.1 Diagnostic local au niveau composant

La fonction de diagnostic local doit déterminer le ou les modes possibles d'un composant qui expliquent les observations locales enregistrées et qui sont cohérents avec les différents modèles du composant. Un composant C^i dans un mode x est modélisé par un ensemble de paramètres $\mathcal{P}^i$, les rangs $\mathcal{R}^i_x$ de ces paramètres dans le mode x et un ensemble de relations $\mathcal{A}^i_x$ entre ces paramètres (voir la section 2.3, page 56) :

$$C^i_x = \langle \mathcal{P}^i, \mathcal{R}^i_x, \mathcal{A}^i_x \rangle. \tag{3.3}$$

On rappelle que l'ensemble des paramètres $\mathcal{P}^i$ est fixé quel que soit le mode opérationnel x du composant. Par contre les relations et les rangs des paramètres peuvent évoluer d'un mode à un autre. Les rangs des paramètres et les relations définissent précisément le mode de fonctionnement du composant (voir la section 2.3 : mode nominal, mode anormal ou modes de faute, page 56).

Une fonction de diagnostic local pour un composant C^i retourne un ensemble de candidats de diagnostic pour le composant C^i. Un candidat de diagnostic local pour un composant C^i est un mode opérationnel m^i_x possible qui est cohérent avec les observations locales OBS^i et les modèles des modes opérationnels du composant C^i, c'est-à-dire que tous les paramètres de $\mathcal{P}^i$ appartiennent à leur rang défini pour le mode m^i_x. Un résultat de diagnostic local Δ^i pour un composant C^i est l'ensemble des candidats de diagnostic local possibles pour le composant C^i. Le résultat de diagnostic local se définit formellement de la manière suivante. On rappelle que l'ensemble des modes possibles pour un composant C^i est noté $\mathcal{M}^i$.

Définition 22 (Résultat de diagnostic local). *Un diagnostic local Δ^i pour un composant C^i est l'ensemble des modes possibles pour ce composant C^i qui sont cohérents avec les observations locales OBS^i et les modèles des modes du composant C^i :*

$$\Delta^i = \{m^i_x \in \mathcal{M}^i \mid \forall p^{i,k} \in OBS^i, p^{i,k} \in r^i_x(p^{i,k}).$$

La notion $p^{i,k} \in OBS^i$ signifie que l'ensemble des mesures du paramètre $p^{i,k}$ fait partie des observations disponibles du composants C^i. Le résultat de diagnostic local Δ^i est dit ambigu s'il contient plusieurs candidats de diagnostic local possibles pour le composant C^i : $|\Delta^i| \geq 1$.

3.4.2 Diagnostic global au niveau système

Il s'agit maintenant de fusionner les différents résultats de diagnostic local afin d'obtenir un diagnostic global qui détermine le mode de fonctionnement du système.

Le mode du système Σ est un N-uplet qui rassemble les modes des N composants du système :

$$m_x^\Sigma = \langle m_{x1}^1, m_{x2}^2, ..., m_{xN}^N \rangle, \tag{3.4}$$

avec $m_x^\Sigma \in \mathcal{M}^\Sigma = \mathcal{M}^1 \times \mathcal{M}^2 \times \ldots \times \mathcal{M}^N$ où $\mathcal{M}^\Sigma$ représente l'ensemble des modes possibles pour le système Σ.

Les diagnostics locaux Δ^i retournent un ensemble de candidats de diagnostic, c'est-à-dire un ensemble de modes possibles pour chaque composant C^i qui est cohérent avec les observations locales et les modèles des composants. En fusionnant les candidats des diagnostics locaux, on obtient plusieurs N-uplets qui sont des hypothèses de diagnostic pour le système complexe Σ. En supposant que l'ensemble des modèles du système est correct et complet, le mode du système Σ correspond bien entendu à l'une de ces hypothèses :

$$m_x^\Sigma \in (\Delta^1 \times \ldots \times \Delta^N). \tag{3.5}$$

Tous les N-uplets de $\mathcal{M}^\Sigma$ obtenus par la fusion des diagnostics locaux ne sont pas forcément globalement cohérents avec l'ensemble des observations OBS et la définition du mode du système qui tient compte des interactions entre les composants qui sont modélisées par les paramètres partagés (entrée/sortie).

Pour tester la cohérence globale d'une hypothèse de diagnostic, il faut s'assurer de la *compatibilité des modes* des composants du système. La compatibilité des modes est dans un premier temps établie hors ligne (*compatibilité statique*) à partir des modèles des composants. On rappelle que la notation $op^{i,k} \rightarrow ip^{j,l}$ signifie que le paramètre de sortie $op^{i,k}$ du composant C^i est connecté au paramètre d'entrée $ip^{j,l}$ de C^j et qu'il lui impose sa valeur.

Définition 23 (Compatibilité statique des modes des composants). *Deux modes de composant, m_x^i pour C^i et m_y^j pour C^j, sont statiquement compatibles si et seulement si toutes les interactions entre eux sont statiquement compatibles :*

$$\begin{aligned}(si\ \exists k,l \mid op^{i,k} \rightarrow ip^{j,l}\ \ alors\ \ \forall k,\forall l,\ r_x(op^{i,k}) \subseteq r_y(ip^{j,l}) \\ \wedge (si\ \exists k,l \mid op^{j,k} \rightarrow ip^{i,l}\ \ alors\ \ \forall k,\forall l,\ r_y(op^{j,k}) \subseteq r_x(ip^{i,l}).\end{aligned} \tag{3.6}$$

Lorsque les intersections des rangs des paramètres d'entrée et de sortie des deux composants C^i et C^j ne sont pas vides et que des observations en ligne OBS du système sont disponibles, il est possible d'évaluer en ligne la *compatibilité dynamiquement* des modes des composants.

DÉFINITION 24 (Compatibilité dynamique des modes des composants). *Deux modes de composant, m_x^i pour C^i et m_y^j pour C^j, sont dynamiquement compatibles si et seulement si toutes les interactions entre eux sont dynamiquement compatibles :*

$$\begin{aligned}&(si\ \exists k,l \mid (op^{i,k} \rightarrow ip^{j,l} \wedge op^{i,k} \in OBS)\ \ alors\ \ \forall k, \forall l,\ op^{i,k} \in r_x(op^{i,k}) \cap r_y(ip^{j,l}))\\ \wedge &(si\ \exists k,l \mid (op^{j,k} \rightarrow ip^{i,l} \wedge op^{j,k} \in OBS)\ \ alors\ \ \forall k, \forall l,\ op^{j,k} \in r_y(op^{j,k}) \cap r_x(ip^{i,l})).\end{aligned} \tag{3.7}$$

Les hypothèses de diagnostic sur les modes des composants qui ne sont pas compatibles statiquement et dynamiquement sont éliminées de manière à rétablir la cohérence globale du diagnostic. Le mode $m_x^\Sigma = \langle m_{x1}^1, m_{x2}^2, ..., m_{xN}^N \rangle$ est un mode globalement cohérent pour le système Σ, si et seulement si $\forall (C^i, C^j) \in Comps^2$, m_{xi}^i et m_{xj}^j sont compatibles. Le sous-ensemble des modes du système qui est obtenu hors ligne en éliminant les modes des composants qui sont statiquement incompatibles est représenté par $\mathcal{MC}^\Sigma$. Lorsque les observations du système le permettent, il est possible d'évaluer en ligne la compatibilité dynamique des modes contenus dans l'ensemble $\mathcal{MC}^\Sigma$.

Une hypothèse de diagnostic portant sur des modes de composants qui sont compatibles entre eux (statiquement ou dynamiquement dans le meilleur des cas) est un candidat de diagnostic global. Au niveau du système complexe global, seuls les modes de faute des composants sont diagnostiqués. En effet les paramètres d'entrée et de sortie partagés par les composants sont considérés comme des paramètres internes. Le mode anormal d'un composant peut être très souvent expliqué par un mode de faute d'un composant avec lequel il interagit. Un candidat de diagnostic global pour le système Σ est donc un mode normal ou un mode de faute $m_x^\Sigma \in \mathcal{MC}^\Sigma$ qui est globalement cohérent avec les observations du système OBS et les modèles des composants. Le diagnostic global Δ^Σ est l'ensemble des candidats de diagnostic possibles pour le système Σ.

DÉFINITION 25 (Résultat de diagnostic global). *Un diagnostic global Δ^Σ pour un système Σ est l'ensemble des modes du système Σ cohérents avec l'ensemble des observations disponibles et les modèles des composants :*

$$\Delta^\Sigma = \{m_1^\Sigma, \ldots, m_x^\Sigma, \ldots, m_z^\Sigma\} \tag{3.8}$$

tel que

$$\begin{cases} m_x^\Sigma \in (\Delta^1 \times \ldots \times \Delta^N) \cap \mathcal{MC}^\Sigma, \\ m_x^\Sigma = \langle m_{x1}^1, \ldots, m_{xN}^N \rangle, \\ \forall m_{xi}^i,\ \forall p^{i,k} \in OBS,\ p^{i,k} \in r_{xi}(p^{i,k}). \end{cases} \tag{3.9}$$

Idéalement, le résultat de diagnostic ne contient qu'un seul candidat, c'est-à-dire qu'une seule hypothèse pour le mode du système. Il est cependant très difficile d'obtenir un résultat de diagnostic certain (non ambigu). Il s'agit d'un problème de *diagnosticabilité* du système dû à des capacités de surveillance insuffisantes (voir le paragraphe 5.2.1).

Lorsque l'ensemble complet des observations depuis la mise en service du système est disponible, il est possible de fournir un diagnostic temporel du système. En plus de déterminer le mode courant du système, le diagnostic doit déterminer la séquence de modes passés du système, c'est-à-dire l'ensemble des modes dans lesquels le système a été depuis le début de son utilisation. Dans la suite du chapitre, la notation $m^{\Sigma}_{x,j}$ correspondant au mode x du système Σ à la date t_j est simplifiée par m^{Σ}_{j}.

Un candidat de diagnostic global pour un système Σ à une date $t_{diag} \in [t_j, t_{j+1}[$ est une séquence de modes possible pour le système $(m^{\Sigma}_0,\ \ldots,\ m^{\Sigma}_j)$ qui est globalement cohérente avec l'ensemble des observations OBS_{diag} disponibles depuis la date t_0 jusqu'à t_{diag}. Les trajectoires déterminées par la fonction de diagnostic n'ont pas forcément la même longueur, c'est-à-dire le même nombre de modes entre t_0 et t_{diag}, car il est possible que certains modes ne soient pas détectables à partir des observations disponibles. Le mode m^{Σ}_j pour $t_j < t_{diag}$ représente alors le dernier mode qui est diagnostiqué. Le diagnostic global $\Delta^{\Sigma}_{t_{diag}}$ au temps $t_{diag} \in [t_j, t_{j+1}[$ retourné par la fonction de diagnostic est l'ensemble de candidats de diagnostic possibles à l'instant t_{diag}. Le mode du composant C^i à la date t_j est noté m^i_j et $r_j(pp^{i,k})$ représente le rang du paramètre $pp^{i,k}$ associé au mode m^i_j. Le diagnostic local obtenu pour le composant C^i à la date t_j se note Δ^i_j.

Définition 26 (Diagnostic temporel d'un système). *Le diagnostic $\Delta^{\Sigma}_{t_{diag}}$ pour un système complexe Σ à l'instant $t_{diag} \in [t_j, t_{j+1}]$ est l'ensemble des séquences de modes possibles du système à partir de sa mise en service à t_0 jusqu'à la date t_{diag} qui sont cohérentes avec les observations $OBS_{t_{diag}}$ disponibles de t_0 à t_{diag} et les modèles des composants :*

$$\Delta^{\Sigma}_{t_{diag}} = \{(m^{\Sigma}_0,\ \ldots,\ m^{\Sigma}_j)\} \tag{3.10}$$

tel que

$$\begin{cases} m^{\Sigma}_j \in (\Delta^1_j \times \ldots \times \Delta^N_j) \cap \mathcal{MC}^{\Sigma}, \\ m^{\Sigma}_j = \langle m^1_j, \ldots, m^N_j \rangle \\ \forall m^{\Sigma}_j,\ \forall p^{i,k} \in OBS_{t_{diag}},\ p^{i,k} \in r_j(p^{i,k}). \end{cases} \tag{3.11}$$

La détermination des modes successifs $(m^{\Sigma}_0, \ldots, m^{\Sigma}_j)$ du système depuis le début de son utilisation nécessite une méthode récursive dont les étapes sont décrites ci-dessous. À sa mise en service, le système est supposé être dans son mode nominal m^{Σ}_n (voir le paragraphe 2.3.2, page 58).

1. Initialiser le diagnostic avec le mode dans lequel se trouve le système Σ dès le début de son utilisation : $\Delta^{\Sigma}_j = \langle m^{\Sigma}_j \rangle$ avec $m^{\Sigma}_j = m^{\Sigma}_{n,0}$.
2. Si une incohérence est détectée entre le mode du système m^{Σ}_j et l'ensemble des observations $OBS_{t_{j+1}}$ disponibles à la date t_{j+1}, rétablir la cohérence en déterminant le nouveau mode du système $m^{\Sigma}_{j+1} \in \mathcal{MC}^{\Sigma}$:

   si $\exists p^{i,k} \in OBS_{t_{j+1}} \mid p^{i,k} \notin r_j(p^{i,k}),$ trouver $m^{\Sigma}_{j+1} \mid \forall p^{i,k} \in OBS_{t_{j+1}},\ p^{i,k} \in r_{j+1}(p^{i,k})$.

3. Calculer le diagnostic Δ^{Σ}_{j+1} en concaténant et en raffinant Δ^{Σ}_{j} avec $\langle m^{\Sigma}_{j+1} \rangle$: $\Delta^{\Sigma}_{j+1} \leftarrow \langle \Delta^{\Sigma}_{j}.m^{\Sigma}_{j+1} \rangle$.
4. $j \leftarrow j+1$; retour au pas 2.

Cette méthode génère une séquence de modes du système Σ pour un horizon temporel $[t_0, t_{diag}]$ qui est globalement cohérente avec l'ensemble complet des observations $OBS_{t_{diag}}$ disponibles à cette date t_{diag}.

La fonction de diagnostic Δ^{Σ} a pour but de déterminer le mode opérationnel m^{Σ}_{x} du système complexe Σ qui est cohérent avec les observations OBS et les modèles des composants du système. Si l'ensemble des observations depuis la mise en service du système est disponible, la fonction de diagnostic peut déterminer la séquence passée de modes du système, c'est-à-dire l'ensemble des modes dans lesquels a été le système depuis le début de son utilisation.

3.4.3 Exemple : diagnostic d'un système de bac

Cette caractérisation du problème de diagnostic peut être illustrée sur l'exemple simple du système de régulation de niveau d'eau dont la modélisation structurelle et fonctionnelle a été détaillée dans les paragraphes 2.2.2.3 et 2.2.3.3. Le système est constitué de cinq composants : $Comps = \{P, B, T, Va_1, Va_2\}$. On considère le cas où il y a une fuite d'eau dans le bac B.

La fonction de diagnostic doit attribuer un mode opérationnel à chaque composant du système de bac d'eau afin de rétablir la cohérence globale avec les observations disponibles du système. Un ensemble de modes opérationnels est connu pour chaque composant du système (voir le paragraphe 2.3.5). Les observations sont enregistrées à l'aide de capteurs positionnés sur les différents composants. On peut alors mesurer la hauteur d'eau h dans le bac, les valeurs des débits q_e et q_s et visualiser les tensions d'alimentation v_p de la pompe et v_v des vannes, ainsi que les positions des deux vannes v_1 et v_2 : $OBS = \{h, q_e, q_s, v_p, v_v, v_1, v_2\}$.

Un diagnostic local est réalisé au niveau de chaque composant et repose sur les modèles de mode nominal et les observations locales des composants du système. Les diagnostics locaux sont ensuite fusionnés afin d'établir un diagnostic global du système de bac d'eau.

Diagnostic local de la pompe P Les observations locales de la pompe P rassemblent l'ensemble des mesures des paramètres v_p et q_e. Les observations locales $OBS^P_j = \{v_p, q_e\}$ et le modèle du mode nominal de la pompe m^P_n sont cohérents. La relation ar^P_n définie à partir de ar^P pour $f_P \in r_n(f_P)$ qui modélise la disponibilité de la fonction élémentaire Fu^P est satisfaite. Aucune autre relation ar^P n'est cohérente

avec OBS^P et comme on est sous l'hypothèse d'un modèle complet alors le diagnostic local $\Delta^P_{t_{diag}}$ de la pompe P à un instant t_{diag} est dit certain : $\Delta^P_{t_{diag}} = \{m^P_n\}$.

Diagnostic local des vannes Va_i Les observations locales $OBS^{Va_i} = \{v_v, v_i\}$ sont cohérentes avec le modèle de bon fonctionnement de la vanne Va_i. La fonction élémentaire Fu^{Va_i} de la vanne Va_i est donc bien réalisée. Aucune autre relation ar^{Va_i} n'est cohérente avec $OBS^{Va_i}_j$, le diagnostic local à un instant t_j de la vanne Va_i est donc également certain : $\Delta^{Va_i}_{t_{diag}} = \{m^{Va_i}_n\}$.

Diagnostic local du bac B Les observations locales $OBS^B_j = \{q_e, q_s, h\}$ indiquent que la relation ar^B_n définie dans le modèle du mode nominal m^B_n du bac B lorsque $f_B \in r_n(f_B)$ n'est pas satisfaite. La fonction élémentaire Fu^B n'est donc pas réalisée, le composant B n'est pas donc son mode nominal, il est soit dans un mode anormal soit dans un mode de faute. Les observations ne peuvent pas nous aider à identifier précisément le mode du composant B. Le diagnostic local à un instant t_j du bac B est ambigu et contient deux candidats de diagnostic local : $\Delta^B_{t_{diag}} = \{m^B_a, m^B_f\}$.

Diagnostic local du tube T Les observations locales $OBS^T_j = \{h, q_s\}$ (qui sont également des observations locales du bac B) montrent que la relation ar^T_n définie dans le modèle du mode nominal m^T_n du tube T n'est également pas satisfaite. La fonction élémentaire Fu^T n'est pas correctement réalisée, le composant T n'est pas donc son mode nominal. Il est soit dans un mode anormal soit dans un mode de faute. Le diagnostic local à un instant t_j du bac T contient deux candidats de diagnostic local, le composant T est soit dans un mode anormal, soit dans un mode de faute : $\Delta^T_{t_{diag}} = \{m^T_a, m^T_f\}$.

Diagnostic global du système de bac Le mode du système à une date t_j est obtenu en rassemblant les résultats des diagnostics locaux :

$$m^\Sigma \in (\{m^P_n\} \times \{m^B_a, m^B_f\} \times \{m^T_a, m^T_f\} \times \{m^{Va_1}_n\} \times \{m^{Va_2}_n\}. \tag{3.12}$$

Il faut ensuite vérifier la cohérence globale de chaque candidat de diagnostic. Les modes du candidat de diagnostic $\langle m^P_n, m^B_f, m^T_a, m^{Va_1}_n, m^{Va_2}_n\rangle$ se sont pas dynamiquement compatibles deux à deux. Le bac et le tube ne peuvent pas être simultanément dans un mode anormal, cette hypothèse doit donc être éliminée. Le diagnostic global du système à l'instant t_{diag} contient alors trois candidats de diagnostic qui sont globalement cohérents avec l'ensemble des observations :

$$\begin{aligned}\Delta^\Sigma = \{&\langle m^P_n, m^B_f, m^T_a, m^{Va_1}_n, m^{Va_2}_n\rangle,\\ &\langle m^P_n, m^B_a, m^T_f, m^{Va_1}_n, m^{Va_2}_n\rangle, \langle m^P_n, m^B_f, m^T_f, m^{Va_1}_n, m^{Va_2}_n\rangle\}.\end{aligned} \tag{3.13}$$

3.5 Caractérisation du problème de pronostic

La section 3.3 a montré que le pronostic peut être défini comme une extension du problème de diagnostic. Le pronostic consiste à prédire la future séquence de modes pour le système. Cette séquence doit être cohérente avec l'ensemble des séquences de modes passés du système déterminées par la fonction de diagnostic. À partir de chaque séquence de modes déterminée par le diagnostic, il faut donc étudier les évolutions possibles du système vers de nouveaux modes de faute. Il s'agit ensuite de prédire l'occurrence du prochain événement de faute qui va modifier le mode opérationnel du système. Tout comme pour le diagnostic, un pronostic local est réalisé au niveau de chaque composant.

3.5.1 Pronostic local au niveau composant

La fonction de pronostic local doit déterminer, pour un composant C^i la séquence future de modes de faute la plus probable. L'occurrence d'une faute sur le composant C^i modifie son mode opérationnel. Une faute correspond à au moins un paramètre privé du composant dont la valeur sort de son rang nominal, c'est-à-dire qu'elle dévie de sa spécification. Une connaissance sur le vieillissement des paramètres spécifiques des composants est généralement connue et provient d'une analyse de sécurité des composants du système (voir le paragraphe 3.2.3, page 66).

Cette connaissance représentée par une loi de vieillissement (ou modèle de vieillissement) va permettre de déterminer la date d'occurrence de la prochaine faute, c'est-à-dire la date à laquelle un paramètre privé $pp^{i,k}$ d'un composant C^i va sortir de son rang nominal $r_n(pp^{i,k})$.

Une loi de vieillissement pour un paramètre privé peut être plus ou moins précise. Dans certains cas, elle peut ne correspondre qu'à une valeur moyenne donnant la date d'apparition de la faute, comme le MTTF (*Mean Time To Failure*) établie par une analyse de fiabilité par exemple. Dans d'autres cas, elle permettra d'estimer le vieillissement d'un paramètre privé selon la manière dont le composant auquel il appartient est sollicité (stressé) par une loi physique. Ces modèles de vieillissement décrivent alors l'évolution de la valeur d'un paramètre privé $pp^{i,k}$ d'un composant C^i dans plusieurs conditions opérationnelles (telles que l'humidité, la pression, les vibrations, ...). Ces conditions opérationnelles peuvent être représentées par des paramètres d'entrée du composant C^i et permettent de modéliser la manière dont le composant C^i est sollicité.

Un modèle de vieillissement $ag^{i,k}$ est associé à un paramètre privé $pp^{i,k}$ d'un composant C^i. Il peut être représenté soit par une connaissance de bas niveau telle qu'une valeur moyenne donnant la date de la faute, soit par une relation (notée alors $ag_x^{i,k}$) entre des paramètres représentant les conditions opérationnelles du mode m_x^i dans lequel se trouve le composant C^i (plus de détails sur les modèles de vieillissement sont

donnés dans le chapitre 4). Une application lv associe à un paramètre privé $pp^{i,k}$ d'un composant C^i supposé être dans un mode opérationnel m_x^i, une loi de vieillissement $ag_x^{i,k}$:

$$\begin{cases} \mathcal{PP}^i \times \mathcal{M}^i \xrightarrow{lv} \mathcal{AG}^i \\ (pp^{i,k}, m_x^i) \xrightarrow{lv} lv(pp^{i,k}, m_x^i) = ag_x^{i,k} \end{cases} \tag{3.14}$$

Le mode opérationnel m_x^i d'un composant C^i définit le rang de ses paramètres privés $r_x(pp^{i,k})$ et permet de sélectionner pour chaque paramètre privé $pp^{i,k}$ le modèle de vieillissement approprié $ag_x^{i,k}$. Une loi représentant une valeur moyenne de la date de la faute ne dépend évidemment pas du mode opérationnel du composant. Néanmoins, quel que soit le modèle disponible, le vieillissement d'un paramètre privé $pp^{i,k}$ peut s'exprimer en terme de probabilité de faute (voir le chapitre 4).

Un candidat de pronostic local à une date $t_{prog} \in [t_j, t_{j+1}[$ pour un composant C^i correspond à la séquence de modes de faute $(m_{j+1}^i, \ldots, m_{pp}^i)$ qui a la plus grande probabilité de se produire dans le futur compte tenu des modèles de vieillissement de $\mathcal{AG}^i$ à partir d'un candidat de diagnostic local qui détermine le mode courant m_j^i du composant C^i. Le dernier mode prédit m_{pp}^i pour le composant C^i correspond au mode de faute dans lequel il se trouve lorsque le système est dans un mode m_p^Σ et ne peut plus réaliser la totalité de ses fonctions objectifs.

Un résultat de pronostic local se définit formellement de la manière suivante. Notons $cont(m_j^i)$, l'ensemble des séquences de modes possibles à partir du mode m_j^i et $Pr_{\mathcal{AG}_j^i}(m_{j+1}^i, \ldots, m_{pp}^i)$, la probabilité qu'à la séquence $(m_{j+1}^i, \ldots, m_{pp}^i)$ de se produire dans le futur compte tenu des modèles de vieillissement $\mathcal{AG}_j^i$ des paramètres du composant C^i sélectionnés pour le mode m_j^i.

Définition 27 (Résultat de pronostic local). *Un résultat de pronostic local $\Pi_{t_{prog}}^i$ pour un composant C^i à l'instant $t_{prog} \in [t_j, t_{j+1}[$ est la séquence future de modes de fautes possible pour ce composant C^i qui a la plus grande probabilité de se produire selon les modèles de vieillissement de $\mathcal{AG}_j^i$:*

$$\Pi_{t_{prog}}^i = (m_{j+1}^i, \ldots, m_{pp}^i), \tag{3.15}$$

tel que

$$\begin{cases} (m_{j+1}^i, \ldots, m_{pp}^i) \in cont(m_j^i) \wedge m_j^i \in \Delta_j^i, \\ \max\{Pr_{\mathcal{AG}_j^i}(m_{j+1}^i, \ldots, m_{pp}^i)\}. \end{cases} \tag{3.16}$$

Un résultat de pronostic local Π_j^i pour un composant C^i à l'instant $t_{prog} \in [t_j, t_{j+1}[$ se détermine alors de la manière suivante.

1. Il faut tout d'abord déterminer le mode opérationnel m_j^i du composant C^i à partir d'un candidat de diagnostic local. Les candidats de diagnostic considérés sont ceux qui n'ont pas été éliminés par la cohérence globale.

2. Pour chaque paramètre privé $pp^{i,k} \in \mathcal{PP}^i$ du composant C^i, on sélectionne la loi de vieillissement appropriée $ag_x^{i,k} = lv(pp^{i,k}, m_j^i)$ et on calcule t_{j+1}, la date de la prochaine faute qui correspond au prochain changement de mode.

La date t_{prog} doit être très proche de t_{diag} pour que le raisonnement de pronostic soit appliqué sous l'hypothèse que le mode courant du composant n'a pas changé. Lorsque le diagnostic local est ambigu et qu'il contient plusieurs candidats de diagnostic local $\Delta_j^i = \{m_j^i\}$, la date t_{j+1} du prochain changement de mode est déterminée à partir de chaque candidat de diagnostic local. Le pronostic local Π_j^i correspond au prochain mode m_{j+1}^i qui est cohérent avec le résultat de diagnostic local Δ_j^i et qui a la plus grande probabilité de se produire selon les modèles de vieillissement de $\mathcal{AG}^i$. Le chapitre 4 définit une fonction de pronostic générique qui permet d'évaluer le mode futur le plus probable pour un composant.

3.5.2 Pronostic global au niveau système

Le pronostic global d'un système Σ consiste à déterminer la séquence de modes de fautes futurs $(m_{j+1}^\Sigma, \ldots, m_p^\Sigma)$ que va suivre le système Σ jusqu'à ce qu'il soit en panne et ne puisse plus assurer la totalité de ses fonctions objectifs.

Le pronostic global du système est obtenu en fusionnant les pronostics locaux calculés au niveau des composants du système. Cette fusion est un peu différente de celle présentée pour le diagnostic car elle ne peut pas tenir compte de la compatibilité dynamique des modes des composants, les observations futures des paramètres ne peuvent évidemment pas être prédites. Les pronostics locaux sont composés en respectant les dépendances fonctionnelles décrites dans le modèle fonctionnel du système (voir le paragraphe 2.2.3.2, page 52). On ne s'attardera pas ici sur les règles de composition des pronostics locaux qui sont précisément décrites dans le chapitre 4.

Un candidat de pronostic global pour un système complexe Σ à une date $t_{prog} \in [t_j, t_{j+1}[$ avec $t_{prog} \geq t_{diag}$ est une séquence de modes futurs $(m_{j+1}^\Sigma, \ldots, m_p^\Sigma)$ possible qui est cohérente avec un candidat de diagnostic global déterminé à t_{diag}. Le pronostic global $\Pi_{t_{prog}}^\Sigma$ détermine le candidat de pronostic global $(m_{j+1}^\Sigma, \ldots, m_{fp}^\Sigma)$ qui a la plus grande probabilité de se produire selon les modèles de vieillissement à partir d'une séquence de modes $(m_0^\Sigma, \ldots, m_j^\Sigma)$ déterminée par la fonction de diagnostic $\Delta_{t_{diag}}^\Sigma$ à la date $t_{diag} \leq t_{prog}$. Notons $cont(m_j^\Sigma)$, l'ensemble des séquences de modes possibles à partir du mode du système m_j^Σ qui sont représentées dans l'automate des modes et $Pr_{\mathcal{AG}_j}(m_{j+1}^i)$, la probabilité qu'a le mode m_{j+1}^i de se produire selon l'ensemble des modèles de vieillissement $\mathcal{AG}_j$ des composants du système sélectionnés pour le mode m_j^i.

Définition 28 (Résultat de pronostic global). *Le pronostic Π_j^Σ pour un système Σ effectué à l'instant $t_{prog} \in [t_{diag}, t_{j+1}]$ consiste à prédire jusqu'à t_p, la date de défaillance du système, la séquence de modes de faute futurs du système la plus probable selon les*

modèles de vieillissement des composants, qui est cohérente avec une séquence de modes passés déterminée par le diagnostic Δ_j^Σ *:*

$$\Pi_j^\Sigma = (m_{j+1}^\Sigma, \ldots, \ m_p^\Sigma), \tag{3.17}$$

tel que

$$\begin{cases} (m_{j+1}^\Sigma, \ldots, m_p^\Sigma) \in cont(m_j^\Sigma) \wedge m_j^\Sigma \in \Delta_j^\Sigma, \\ \max\{Pr_{\mathcal{AG}_j}(m_{j+1}^\Sigma, \ldots, m_p^\Sigma)\}. \end{cases} \tag{3.18}$$

Une fonction de pronostic est définie dans le chapitre 4 et permet d'évaluer la séquence future de modes de faute la plus probable pour le système à partir de chaque séquence de mode déterminée par la fonction de diagnostic. Les évolutions de modes possibles pour le système sont représentées dans l'automate des modes. L'occurrence d'un événement de faute qui modifie le mode opérationnel du système est prédite avec une certaine probabilité à l'aide des modèles de vieillissement des composants.

La prédiction de la séquence future de modes du système $(m_{j+1}^\Sigma, \ldots, \ m_p^\Sigma)$ nécessite une méthode itérative qui couple le problème de diagnostic et de pronostic. Cette méthode doit s'arrêter à la date t_p à laquelle le système n'assure plus la totalité de ses fonctions objectifs. Les différentes étapes de cette méthode sont énumérées ci-dessous.

1. Déterminer le mode du composant $m_j^\Sigma = \langle m_j^1, m_j^2, \ldots, m_j^N \rangle$ à partir d'un candidat de diagnostic local mais globalement cohérent.
2. Sélectionner les lois de vieillissement $\{ag_j^{i,k}\}$ des paramètres privés $\{pp^{i,k}\}$ des composants du système et calculer la date t_{j+1} qui correspond au prochain changement de mode vers m_{j+1}^Σ.
3. Initialiser le pronostic $\Pi_j^\Sigma \leftarrow \langle m_{j+1}^\Sigma \rangle$.
4. À partir du dernier mode contenu dans Π_j^Σ, déterminer les lois de vieillissement $ag_{j+1}^{i,k}$ des paramètres privés des composants du systèmes et calculer la date t_{j+2} du prochain changement de mode vers m_{j+2}^Σ.
5. Calculer le pronostic Π_{j+1}^Σ en concaténant Π_j^Σ et $\langle m_{j+2}^\Sigma \rangle$: $\Pi_{j+1}^\Sigma \leftarrow \langle \Pi_j^\Sigma . m_{j+2}^\Sigma \rangle$.
6. $j \leftarrow j+1$; retour au pas 4 si $\forall Fu_s^i \in \mathcal{FU}_s$, Fu_s^i est réalisée.

Cette méthode génère une séquence de modes du système pour un horizon temporel $[t_j, t_p]$ avec t_p, la date à laquelle le système est dans un mode de panne et doit être réparé. Dans le cas où il y a plusieurs candidats de diagnostic possibles, $\Delta_j^\Sigma = \{m_j^\Sigma\}$, la méthode doit être appliquée à partir de chacun de ces candidats avant d'évaluer la trajectoire la plus probable à partir des modèles de vieillissement des composants.

La fonction de pronostic Π^Σ d'un système Σ consiste à prédire la séquence de modes futurs possible pour le système Σ qui est cohérente avec un candidat de diagnostic déterminé par la fonction Δ^Σ et qui a la plus grande probabilité de se produire selon l'ensemble des modèles de vieillissement des composants du système.

3.5.2.1 Prédiction de modes de fautes et RUL

Dans la littérature, le pronostic se définit comme étant la prédiction de la durée de vie résiduelle (RUL) d'un système ou d'un composant (voir la section 1.4, page 28). Dans le cadre que nous avons défini, le pronostic est l'estimation de la séquence de modes futurs du système. Cette séquence correspond à la séquence la plus probable considérant le fait que le système est supposé être dans le mode m_j^Σ à la date $t_{prog} < t_{j+1}$. Il s'agit alors de calculer la date t_{j+1} du prochain changement de mode du système.

Si le changement de mode du système à t_{j+1} de m_j^Σ vers m_{j+1}^Σ est dû à une déviation d'un paramètre privé du composant C^i, alors pour $t_{prog} \in [t_j, t_{j+1}]$, le RUL du composant C^i qui sera prochainement en faute est donné par

$$RUL(C^i) = t_{j+1} - t_{prog}. \tag{3.19}$$

En réitérant ces prédictions de changements de mode, on obtient le RUL du système :

$$RUL(\Sigma) = t_p - t_{prog}, \tag{3.20}$$

où t_p est la date à laquelle le système passe dans un mode de panne m_p^Σ. Les composants pouvant être redondants dans le système, il est possible que le RUL du système complexe global soit plus grand que le RUL des composants.

3.5.3 Exemple : pronostic d'un système de bac

La notion de modèle de vieillissement va être illustrée sur l'exemple du système de bac dont le problème de diagnostic a été caractérisé dans le paragraphe 3.4.3.

Lorsque le système de bac Σ est dans son mode nominal m^Σ à t_j, le futur mode m_{j+1}^Σ correspondant à la déviation d'un paramètre privé des composants peut être pronostiqué à l'aide des lois de vieillissement des paramètres privés des composants. La loi de vieillissement associée à un paramètre privé $pp^{i,k}$ dans un mode m_j^i est représentée par une relation $ag_j^{i,k}$ entre des paramètres du composant qui décrit la manière dont évolue la valeur de ce paramètre $pp^{i,k}$.

Mode nominal Dans le mode nominal du système, des paramètres mécaniques tels que f_P, représentant le paramètre privé mécanique de la pompe P, peut être affecté, détérioré par la tension ou/et le temps d'utilisation. La tension v_P est un facteur de stress pour le composant P. Un modèle de vieillissement pouvant être associé au paramètre f_P est de la forme : $f_P = ag_n^P(v_P, t)$. Les vannes Va_1 et Va_2 sont également des éléments mécaniques qui peuvent être détériorées par trois facteurs de stress : l'eau polluée, modélisée par $\rho_{liquide}$, qui encrasse (ou bouche) la vanne, le nombre de commutations N_{sw} effectuées par la vanne, et un débit q_s trop important forçant le passage

d'eau au niveau de la vanne. Les modèles de vieillissement des paramètres privés f_{v^1} et f_{v^2} sont de la forme : $f_{v^i} = ag_n^{Va_i}(\rho_{liquide}, N_{sw}, q_s)$. On ne considérera pas le vieillissement du paramètre privé s_B, correspondant à la section du bac car $s_B >> s_T$, le tube sera encrassé beaucoup plus rapidement que le bac.

Cet ensemble de lois de vieillissement associé au mode nominal m_n^Σ est utilisé par la fonction de pronostic pour calculer la date de la prochaine occurrence de faute (le prochain changement de mode du système dû à l'évolution d'un paramètre privé en dehors de son rang nominal).

Mode de faute Lorsque le bac B est détecté dans un mode de faute m_f^B, c'est-à-dire qu'il y a une fuite dans B, un autre ensemble de lois de vieillissement doit être utilisé pour les paramètres privés des composants car les conditions de stress des composants sont modifiées. L'ensemble des lois doit considérer des phénomènes physiques. Par exemple la présence de la fuite dans B fait qu'un débit q_s anormalement bas circule dans T, l'eau stagne à cet endroit plus qu'elle ne le devrait et le taux d'encrassement du tube est plus élevé que dans le mode nominal. Cela affecte directement le paramètre privé s_T.

Si le diagnostic détermine en ligne que le composant B est dans le mode de faute m_f^B, les modèles de vieillissement associés à ce mode sont utilisés pour chaque paramètre privé des composants afin de calculer un pronostic qui tient compte des conditions de stress réelles du système.

3.6 Conclusion

L'architecture de supervision que nous avons proposée intègre un module de diagnostic et un module de pronostic pour aider à la prise de décisions de maintenance. Le formalisme que nous avons présenté dans le chapitre 2, qui repose sur un ensemble de modes opérationnels pour les composants (des paramètres, des rangs et des relations entre les paramètres), sert de connaissance commune aux modules de diagnostic et de pronostic.

Une représentation générique des problèmes de diagnostic et de pronostic qui s'appuie sur ce formalisme unificateur permet de faciliter la communication entre les modules de l'architecture mais également de définir un couplage diagnostic-pronostic. Le cadre que nous avons posé dans ce chapitre permet de définir formellement le problème du pronostic à partir d'un résultat de diagnostic.

Le diagnostic consiste à déterminer l'ensemble des séquences de modes passés qui sont cohérentes avec les observations et les modèles des composants. Le pronostic consiste à prédire la succession de modes de faute futurs qui est cohérente avec le résultat de diagnostic et la plus probable en tenant compte des modèles de vieillissement des paramètres privés des composants.

4 Définition d'une fonction générique de pronostic

Résumé : Une fonction générique de pronostic est définie à l'aide d'un modèle de Weibull dans le but d'évaluer la séquence future de modes du système qui a la plus grande probabilité de se produire selon les modèles de vieillissement des composants. Pour déterminer le prochain mode du système, la fonction de pronostic fournit une probabilité de faute pour chaque paramètre privé des composants. Les composants étant parfois redondants, il est nécessaire d'obtenir un pronostic pour chaque fonction mise en œuvre par le système afin de déterminer le temps restant avant la prochaine défaillance. La distribution de probabilité de défaillance des fonctions repose sur les modèles de vieillissement des composants. La fonction de pronostic définie est adaptative et permet de prendre en compte les facteurs de stress qui influencent le vieillissement des composants. Le RUL du système complexe est obtenu en combinant ces pronostics de fonctions.

4.1 Introduction

Le problème de pronostic est caractérisé dans la section 3.5 (page 80). Il s'agit de prédire la séquence future de modes du système qui a la plus grande probabilité de se produire selon les modèles de vieillissement des composants. Le pronostic d'un système consiste essentiellement à estimer à un instant $t_{prog} \geq t_j$, la date $t_{j+1} \geq t_{prog}$ à laquelle le système changera de mode de fonctionnement. Une fonction de pronostic générique est définie pour évaluer la probabilité associée à chaque futur mode possible pour le système. Cette probabilité est obtenue à partir d'un ensemble de modèles de vieillissement disponibles dans le mode opérationnel courant m_j^{Σ} du système Σ. On suppose donc qu'un ensemble de modèles de vieillissement $\{ag^{i,k}\}$ (qui sont aussi appelés des modèles de durée de vie dans [WIL 04]) est disponible pour chaque paramètre privé $pp^{i,k} \in \mathcal{PP}$ du système.

Les modèles de vieillissement $\{ag^{i,k}\}$ d'un paramètre $pp^{i,k}$ représentent la connaissance disponible sur l'usure du composant C^i. Ils peuvent n'indiquer qu'une valeur moyenne de la prochaine faute comme un MTTF (c'est-à-dire la date moyenne à laquelle le paramètre $pp^{i,k}$ sera en dehors du rang défini par le modèle nominal du composant C^i, voir le paragraphe 2.3.3, page 58) ou décrire l'évolution de la valeur de $pp^{i,k}$ en fonction des conditions opérationnelles associées au mode du composant m_x^i. Ces condi-

tions opérationnelles représentent les facteurs de stress qui influencent le vieillissement du composant dans le mode m_x^i, comme par exemple les conditions environnementales telles que la température, l'humidité, les vibrations, ... [RIB 09]. Dans certains cas, la loi de vieillissement ne change pas quelles que soient les conditions opérationnelles, elle est alors indépendante du mode opérationnel du composant.

Pour déterminer la date du prochain changement de mode d'un composant du système, il faut estimer le temps restant avant que les paramètres privés $\{p^{i,k}\}$ soient en faute. Pour un paramètre $pp^{i,k}$, ce temps est noté $rtf(pp^{i,k})$ (*rtf* pour *Remaining Time to Fault*). Une probabilité de faute est établie pour chaque paramètre $pp^{i,k} \in \mathcal{PP}$ à partir des modèles de vieillissement sélectionnés dans le mode opérationnel courant du composant m_x^i. Notons $f_{pp^{i,k}}$, la fonction de densité de probabilité (pdf pour *probability density function*) qui représente la probabilité de faute du paramètre $pp^{i,k}$ dans le mode m_x^Σ. L'estimation du $rtf(pp^{i,k})$ consiste à déterminer la date t_{max} à partir de laquelle la probabilité de faute du paramètre $pp^{i,k}$ aura atteint un seuil P_{max} :

$$rtf(pp^{i,k}) = t_{max} \text{ tel que } \int_0^{t_{max}} f_{pp^{i,k}}(t)dt = P_{max}. \tag{4.1}$$

La valeur P_{max} correspond à un seuil de probabilité de faute considérée comme non acceptable pour le système d'aide à la décision en tenant compte du risque de la faute du paramètre $pp^{i,k}$ et du coût de la réparation qu'elle engendre. La date du prochain changement de mode du système Σ est ensuite calculée par

$$t_{j+1} = t_{prog} + \min(rtf(pp^{i,k}), pp^{i,k} \in \mathcal{PP}). \tag{4.2}$$

Le prochain mode du système m_{j+1}^Σ correspond au mode dans lequel le paramètre privé $pp^{i,k}$ est en dehors de son rang défini dans le modèle du mode m_j^Σ.

Ce chapitre définit une fonction générique de pronostic qui fournit la probabilité de faute d'un paramètre privé à l'aide d'un modèle de Weibull. La fonction de pronostic établit ensuite une probabilité de défaillance associée à chacune des fonctions mises en œuvre par le système complexe. La durée de vie résiduelle du système est obtenue en combinant ces pronostics de fonctions.

4.2 Représentation d'un pronostic fondé sur la fiabilité

Les paramètres privés des composants du système représentent généralement les attributs physiques des composants et sont donc totalement hétérogènes. La difficulté est d'établir une représentation du pronostic commune à chaque type de paramètre privé. Cette représentation doit être aussi flexible que possible afin de représenter les fonctions de densité de probabilité de faute de chaque paramètre privé du système. Le modèle probabiliste de Weibull est très souvent utilisé dans le domaine de la fiabilité et pour l'analyse des données de vie de systèmes [HUM 02; KIR 04; FER 08]. C'est pour

cette raison que nous l'utilisons, la plupart des représentations de la loi de vieillissement pouvant être approximées par un modèle de Weibull en théorie.

4.2.1 Modèle de Weibull

Le modèle de Weibull est très souvent utilisé dans le domaine de la fiabilité pour représenter une fonction de densité de probabilité [VAC 06]. Ce modèle probabiliste est très flexible et dépend de caractéristiques lui permettant de reproduire le comportement d'autres lois de probabilité telles que la loi exponentielle ou la loi normale. La fonction de densité de probabilité de Weibull est la suivante :

$$W(t,\beta,\eta,\theta) = \frac{\beta}{\eta}\Big(\frac{t-\theta}{\eta}\Big)^{(\beta-1)} e^{-(\frac{t-\theta}{\eta})^{\beta}} \tag{4.3}$$

où $t \geq 0$, $\beta \geq 0$, $\eta \geq 0$ et $\theta \geq 0$. La caractéristique β modifie la forme de la distribution, η définit l'échelle et θ détermine sa localisation sur l'axe temporel.

Le modèle de Weibull est utilisé pour représenter la densité de probabilité de faute d'un paramètre privé d'un composant. Pour un paramètre privé $pp^{i,k} \in \mathcal{PP}^i$ et un seuil de probabilité P_{max} fixé a priori, le temps restant avant que le paramètre $pp^{i,k}$ ne devienne fautif est évalué par l'expression :

$$rtf(pp^{i,k}) = t_{max} \;\text{ tel que }\; \int_0^{t_{max}} W(t,\beta_{pp^{i,k}},\eta_{pp^{i,k}},\theta_{pp^{i,k}})dt = P_{max}. \tag{4.4}$$

Les caractéristiques $\beta_{pp^{i,k}}$, $\eta_{pp^{i,k}}$ et $\theta_{pp^{i,k}}$ définissent complètement la distribution de probabilité de faute et modélisent par conséquent la manière dont vieillit le paramètre privé $pp^{i,k}$. La description des caractéristiques $\beta_{pp^{i,k}}, \eta_{pp^{i,k}}$ et $\theta_{pp^{i,k}}$ repose sur les modèles de vieillissement qui sont associés au paramètre $pp^{i,k}$. Dans le cas où les modèles de vieillissement tiennent compte des conditions opérationnelles des composants, la connaissance disponible pour chaque paramètre privé $pp^{i,k}$ du système est caractérisée par des relations qui définissent les caractéristiques $\beta_{pp^{i,k}}$, $\eta_{pp^{i,k}}$ et $\theta_{pp^{i,k}}$ du modèle de Weibull :

$$\begin{cases} \beta_{pp^{i,k}} = ar_{\beta}(ip^{i,1},\ldots,ip^{i,n},pp^{i,1},\ldots,pp^{i,m}), \\ \theta_{pp^{i,k}} = ar_{\theta}(ip^{i,1},\ldots,ip^{i,n},pp^{i,1},\ldots,pp^{i,m}), \\ \eta_{pp^{i,k}} = ar_{\eta}(ip^{i,1},\ldots,ip^{i,n},pp^{i,1},\ldots,pp^{i,m}). \end{cases} \tag{4.5}$$

Les valeurs de $\beta_{pp^{i,k}}$, $\eta_{pp^{i,k}}$ et $\theta_{pp^{i,k}}$ dépendent des conditions opérationnelles du composant C^i qui représentent les divers facteurs de stress interne ou externe agissant sur le composant (comme par exemple la température, la pression, l'humidité) qui peuvent être représentés par des paramètres d'entrée ou des paramètres privés du composant.

4.2.2 Caractéristiques du modèle

Les valeurs des caractéristiques $\beta_{pp^{i,k}}$, $\eta_{pp^{i,k}}$, et $\theta_{pp^{i,k}}$ définissent complètement la pdf de Weibull. Une modification des caractéristiques peut influer sur la forme et la localisation de la courbe.

Une variation de la caractéristique $\theta_{pp^{i,k}}$ déplace la fonction de Weibull le long de l'axe temporel. Cette caractéristique définit la durée vie minimale d'une entité, par conséquent ici d'un paramètre $pp^{i,k}$. Dans la plupart des cas, on suppose que $\theta_{pp^{i,k}}$ est nul. L'échelle de temps de la courbe de Weibull commence à $t = 0$, ce qui signifie que la probabilité de faute d'un paramètre $pp^{i,k}$ à $t = 0 + \epsilon$ est non nulle. Les paramètres peuvent devenir fautifs dès $t = 0 + \epsilon$. Cette hypothèse n'est pas restrictive, au contraire, si $\theta_{pp^{i,k}} = c$, avec c représentant une constante positive, la probabilité de faute établie par le modèle de Weibull serait nulle jusqu'à $t = c$.

La caractéristique $\eta_{pp^{i,k}}$ est appelée le paramètre d'échelle de la fonction de probabilité de Weibull. Elle définit la durée de vie caractéristique d'une entité et correspond à la durée de vie moyenne pour un échantillon de population étudié. Une modification de $\eta_{pp^{i,k}}$ entraîne une modification de l'abscisse et de la forme de la pdf de Weibull qui devient plus ou moins large comme le montre la figure 4.1. Comme pour toute densité de probabilité, l'aire sous la courbe de Weibull est toujours égale à 1. La valeur maximale de la courbe décroît forcément lorsque la caractéristique $\eta_{pp^{i,k}}$ augmente, la courbe est donc plus large.

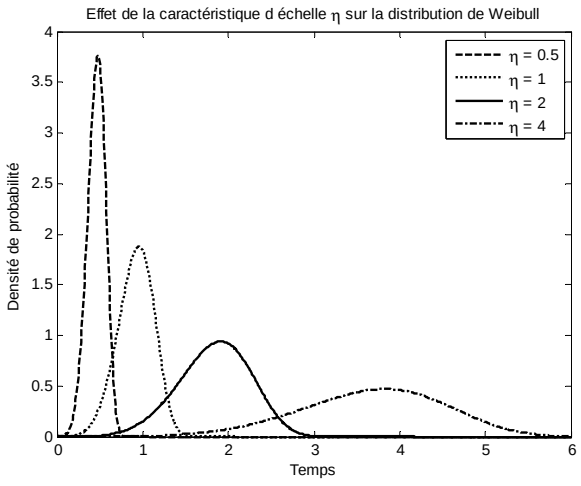

FIG. 4.1 – Modification de la caractéristique η de la pdf de Weibull

La caractéristique de forme $\beta_{pp^{i,k}}$ change la nature de la pdf de Weibull et permet à la courbe de modéliser les différentes phases de vie d'un composant représentées sur la courbe baignoire de la fiabilité illustrée sur la figure 1.11, page 33 [HUM 02][WIL 04].

La courbe baignoire est une courbe idéalisée du taux de défaillances qui illustre les

trois phases de la vie d'un composant : la période de jeunesse, la période de vie utile et la période de vieillissement (d'usure rapide). Cette courbe baignoire représente l'évolution du taux de défaillance dont l'expression dérivant du modèle de Weibull est la suivante :

$$H(t, \beta_{pp^{i,k}}, \eta_{pp^{i,k}}, \theta_{pp^{i,k}}) = \frac{\beta_{pp^{i,k}}}{\eta_{pp^{i,k}}} \Big(\frac{t - \theta_{pp^{i,k}}}{\eta_{pp^{i,k}}} \Big)^{(\beta_{pp^{i,k}} - 1)}. \tag{4.6}$$

La valeur de la caractéristique $\beta_{pp^{i,k}}$ est déterminée par la région de vie du composant que l'on considère. Durant la période de jeunesse du composant, le taux de défaillance H décroît. La fonction de Weibull modélise ces défaillances précoces par $\beta_{pp^{i,k}} < 1$. Tout au long de la vie utile d'un composant, le taux de défaillance H est constant : $H(t) = \frac{1}{\eta_{pp^{i,k}}}$. La fonction de Weibull modélise cette période de vie utile avec $\beta_{pp^{i,k}} = 1$, elle est alors similaire à la loi exponentielle. Lorsque $\beta_{pp^{i,k}} > 1$, la fonction de Weibull modélise l'usure rapide qui correspond à la dégradation physique des entités en fin de vie. Durant cette période, le taux de défaillance H augmente et la fonction de Weibull prend la forme approximative d'une gaussienne. Par exemple pour la valeur précise $\beta_{pp^{i,k}} = 3.4$, la pdf de Weibull reproduit la comportement de la distribution normale. L'impact d'une modification de la caractéristique $\beta_{pp^{i,k}}$ sur la pdf de Weibull est représenté sur la figure 4.2.

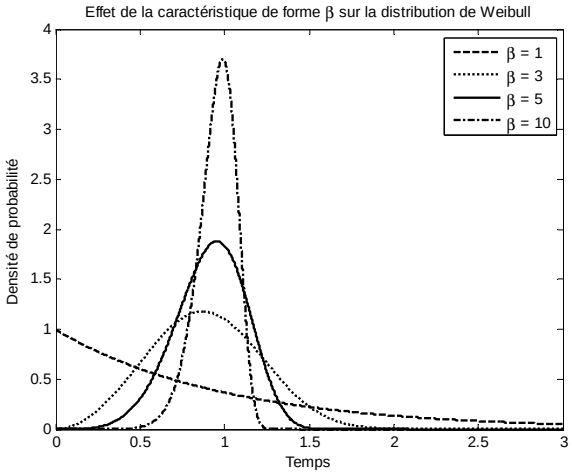

FIG. 4.2 – Modification de la caractéristique β de la pdf de Weibull

Tout comme pour les modes de faute définis dans le paragraphe 2.3.3 (page 58), si la connaissance sur les composants est disponible, il est donc aussi possible de distinguer différents modes de dégradation tels que la période de jeunesse, la période de vie utile et l'usure rapide. Ces modes de dégradation sont définis par un ensemble de relations qui permettent de déterminer les valeurs des caractéristiques $\beta_{pp^{i,k}}, \theta_{pp^{i,k}}$ et $\eta_{pp^{i,k}}$.

La caractéristique $\theta_{pp^{i,k}}$ est supposée nulle pour les raisons expliquées précédemment. Si on considère qu'une étape de déverminage des composants a été effectuée, seule la période de vie utile du composant et la période d'usure rapide sont à prendre en compte.

Lorsque le modèle de vieillissement d'un paramètre $pp^{i,k}$ contient assez d'informations pour estimer son vieillissement en fonction des conditions opérationnelles du composant, on peut ne s'intéresser qu'à la durée de vie utile du composant. L'objectif étant de prévoir la faute avant que le composant ne se dégrade trop rapidement et doive être immédiatement remplacé. Dans ce cas, $\beta_{pp^{i,k}} = 1$.

Lorsque le modèle de vieillissement d'un paramètre $pp^{i,k}$ ne donne qu'un MTTF, c'est-à-dire le temps moyen avant la prochaine faute, cette information peut être également représentée à l'aide d'un modèle de Weibull par $\beta_{pp^{i,k}} \gg 1$ et $\eta_{pp^{i,k}} = \text{MTTF}$.

Les caractéristiques $\beta_{pp^{i,k}}$ et $\eta_{pp^{i,k}}$ dépendent de la connaissance représentée par le modèle de vieillissement. Dans le cas d'un modèle physique qui permet de prendre en compte les conditions opérationnelles du composant, les facteurs de stress ne peuvent agir que sur $\eta_{pp^{i,k}}$, les deux autres caractéristiques étant fixées. Les relations définissant les caractéristiques d'un modèle de Weibull pour un paramètre privé $pp^{i,k} \in \mathcal{PP}^i$ sont les suivantes :

1. Cas d'un modèle physique (prise en compte des facteurs de stress) :

$$\begin{cases} \beta_{pp^{i,k}} = 1, \\ \theta_{pp^{i,k}} = 0, \\ \eta_{pp^{i,k}} = ar_\eta(ip^{i,1}, \ldots, ip^{i,n}, pp^{i,1}, \ldots, pp^{i,m}). \end{cases} \tag{4.7}$$

2. Cas d'un modèle donnant un MTTF :

$$\begin{cases} \beta_{pp^{i,k}} > 1, \\ \theta_{pp^{i,k}} = 0, \\ \eta_{pp^{i,k}} = \text{MTTF}. \end{cases} \tag{4.8}$$

Le modèle probabiliste de Weibull est très utilisé dans le domaine de la fiabilité. Il permet ici de représenter de manière générique les fonctions de densité de probabilité de faute de chaque paramètre privé du système. La probabilité de faute d'un paramètre privé $pp^{i,k}$ à un instant t est alors donnée par l'intégrale suivante :

$$\int_0^t W(t, \beta_{pp^{i,k}}, \eta_{pp^{i,k}}, \theta_{pp^{i,k}})dt = \int_0^t \frac{\beta_{pp^{i,k}}}{\eta_{pp^{i,k}}} e^{-\left(\frac{t}{\eta_{pp^{i,k}}}\right)^{\beta_{pp^{i,k}}}} dt. \tag{4.9}$$

4.3 Caractérisation d'une fonction générique de pronostic

Pour optimiser la maintenance, le pronostic doit prédire le RUL du système complexe à partir duquel les actions de maintenance pourront être programmées (voir la section 1.4.1, page 28). Le RUL du système correspond au temps restant avant que le système

ne puisse plus réaliser la totalité de ses fonctions objectif de $\mathcal{FU}_s$. Cette section montre comment le RUL du système peut être estimé à partir des probabilités de faute associées aux paramètres privés des composants. Les fonctions objectif du système sont obtenues par une composition des fonctions élémentaires qui est décrite par le modèle fonctionnel du système (voir la section 2.2.3.2, page 52). Dans les systèmes complexes, les fonctions élémentaires sont souvent mises en œuvre par des composants redondants. C'est la raison pour laquelle le pronostic doit tenir compte de l'aspect fonctionnel du système [VOI 09] [DRA 07]. Pour représenter la disponibilité dans le futur de chaque fonction objectif du système, il faut obtenir un pronostic pour chaque fonction élémentaire $F \in \mathcal{FU}^i$ mise en œuvre par les composants du système.

4.3.1 Représentation générique d'un pronostic fonctionnel

Le but d'un pronostic fonctionnel est d'évaluer la disponibilité dans le futur d'une fonction implémentée par un ou plusieurs composants du système.

Définition 29 (Pronostic fonctionnel). *Le pronostic d'une fonction $F \in \mathcal{FU}$ consiste à estimer le temps qu'il reste avant que la fonction F ne soit plus disponible dans le système. Ce temps est noté $ettf(F)$ (pour* estimated time to failure*).*

La réalisation d'une fonction de base F repose sur un ensemble de paramètres privés que l'on note $PP(F)$. Pour qu'une fonction F soit disponible, il est nécessaire que l'ensemble des paramètres privés de $PP(F)$ ne soient pas en faute. L'*ettf* d'une fonction F peut alors être évalué à partir des *rtf* des paramètres privés de $PP(F)$:

$$ettf(F) = min(rtf(pp^{i,k}), pp^{i,k} \in PP(F)). \tag{4.10}$$

Le vieillissement d'une fonction élémentaire F dépend naturellement du vieillissement des paramètres privés du composant qui permettent de mettre en œuvre cette fonction F. La probabilité de défaillance associée à la disponibilité d'une fonction F peut être également représentée de manière générique par un modèle de Weibull :

$$\int_0^t W(t, \beta_F, \eta_F, \theta_F)dt. \tag{4.11}$$

Les caractéristiques β_F, η_F et θ_F du modèle de Weibull sont directement déterminées à partir des caractéristiques $\{\beta_{pp^{i,k}}, \eta_{pp^{i,k}}, \theta_{pp^{i,k}}\}$ associées aux modèles de Weibull des paramètres $\{pp^{i,k}\} \in PP(F)$, où $PP(F)$ représente l'ensemble des paramètres privés sur lesquels reposent la réalisation de la fonction F :

$$\begin{cases} \theta_F = 0, \\ \beta_F \geq 1 \text{ (selon les modèles de vieillissement considérés pour } \{pp^{i,k}\} \in PP(F)), \\ \eta_F = \min\{\eta_{pp^{i,k}} \mid pp^{i,k} \in PP(F)\}. \end{cases} \tag{4.12}$$

Les paramètres privés sont supposés non redondés dans le composant C^i mettant en œuvre la fonction F.

4.3.2 Fonction de pronostic adaptative

Dans l'architecture que l'on a présentée dans la section 3.2 (page 64), un module de pronostic local pour un sous-système γ a pour objectif de déterminer les *ettf* de chaque fonction élémentaire mise en œuvre par les composants de γ. Pour cela, une fonction de densité de probabilité de Weibull est attribuée à chaque fonction élémentaire F afin de représenter la densité de probabilité de défaillance à partir de laquelle le *ettf* est évalué. Le *ettf* d'une fonction élémentaire F est tout d'abord initialisé au $MTTF$ du composant qui la met en œuvre. Ce $MTTF$ est supposé être fourni par les concepteurs du système.

L'évolution de la probabilité de défaillance dépend des modèles de vieillissement des paramètres privés de $PP(F)$ qui peuvent tenir compte des facteurs de stress externes et internes ayant une influence sur les composants du système. Un facteur de stress représente une sollicitation anormale des composants. Lorsque les modèles de vieillissement des paramètres $\{pp^{i,k}\} \in PP(F)$ prennent en compte ce genre d'information (cas d'un MTTF, voir 92.), il est possible de définir une fonction de pronostic adaptative.

Les caractéristiques β_F et θ_F sont alors fixées par les caractéristiques des paramètres : $\beta_F = \beta_{pp^{i,k}} = 1$ et $\theta_F = \theta_{pp}^{i,k} = 0$ tel que $pp^{i,k} \in PP(F)$. Seule la caractéristique d'échelle d'un paramètre $\eta_{pp^{i,k}}$ peut être modifiée par ces facteurs de stress et permet de déterminer la caractéristique η_F associée à la distribution de Weibull de la fonction F. La section suivante expliquera comment η_F peut être obtenu à partir des $\{pp^{,ik}\} \in pp(F)$. Une variation de $\eta_{pp^{i,k}}$ modifie la pdf de Weibull $W(t, \beta_{pp^{i,k}}, \eta_{pp^{i,k}}, \theta_{pp^{i,k}})$ à partir de laquelle est évalué le rtf des paramètres $\{pp^{i,k}\} \in PP(F)$ puis le *ettf* de la fonction F.

Lorsqu'un composant C^i est fréquemment et fortement sollicité, la caractéristique $\eta_{pp^{i,k}}$ du modèle de Weibull de ses paramètres diminue. La courbe de Weibull est plus pentue et par conséquent, le *rtf* du paramètre privé $pp^{i,k}$ est plus court. Ce cas est illustré sur la figure 4.3.

Lorqu'au contraire, le composant C^i est moins sollicité que ce qui était attendu, la caractéristique $\eta_{pp^{i,k}}$ de la distribution de Weibull associé aux paramètres privés de $\mathcal{PP}^i$ augmente. La probabilité de faute d'un paramètre privé $pp^{i,k}$ représentée par la courbe de Weibull évolue beaucoup plus progressivement et par conséquent le *rtf* du paramètre $pp^{i,k}$ est augmenté. Ce cas est illustré sur la figure 4.4. Il faut donc trouver un moyen de quantifier le stress qui agit sur le système à partir duquel la valeur des caractéristiques $\eta_{pp^{i,k}}$ des paramètres privés des composants pourra être déterminée.

En effet, la caractéristique $\eta_{pp^{i,k}}$ du modèle de Weibull d'un paramètre privé $pp^{i,k}$ peut être obtenue à partir d'une fonction f des facteurs de stress du composant :

$$\eta_F = f(\mathcal{CE}, \Delta^i, \mathcal{IT}, \mathcal{CM}). \tag{4.13}$$

Cette fonction f dépend des modèles de vieillissement $\mathcal{AG}^i$ des paramètres privés du composant C^i. Elle peut être représentée par une relation $ar_{\eta^{pp^{i,k}}}$ entre des paramètres

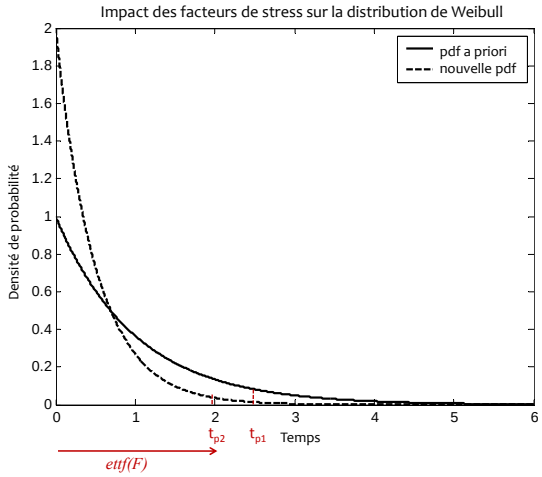

FIG. 4.3 – Impact d'un stress anormalement élevé sur la pdf de Weibull

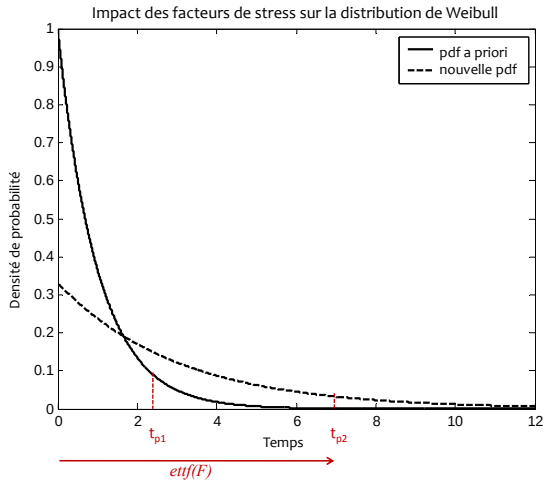

FIG. 4.4 – Impact d'un stress anormalement faible sur la pdf de Weibull

qui correspondent aux facteurs de stress du composant (voir l'équation 4.5). Le vieillissement d'un composant est influencé par des facteurs de stress qui peuvent être répartis en deux groupes : les *facteurs de stress externes* (résultant de phénomènes externes au composant C^i) et les *facteurs de stress internes* (résultant de phénomènes internes au composant C^i).

- Les conditions environnementales anormales $\mathcal{CE}$ d'un système représentent un facteur de stress externe qui peut dégrader les composants du système en modifiant leur sollicitation. Ce facteur de stress peut être modélisé à partir d'une relation entre des paramètres d'entrée du système (voir le paragraphe 2.2.2.2, page 48) qui correspondent à des données physiques telles que la température, l'humidité, les

vibrations, ... :

$$\mathcal{CE} = ar_{\mathcal{CE}}(ip^{i,k}, \ldots, ip^{j,l}) \mid \{ip^{i,k}, \ldots, ip^{j,l}\} \subseteq \mathcal{IP}. \tag{4.14}$$

- L'occurrence d'une faute sur un composant peut générer un stress interne sur l'ensemble du composant et altérer le vieillissement des autres paramètres privés du composant s'ils sont impliqués dans une même relation. Une faute associée à un paramètre privé $pp^{i,j}$ du composant C^i peut être la cause d'une faute du paramètre $pp^{i,k}$. Le résultat de diagnostic Δ^i du composant C^i est donc considéré comme un facteur de stress interne dès lors qu'il détecte l'occurrence d'une faute dans le composant.

- L'ensemble $\mathcal{IT}$ modélise l'effet que peuvent avoir les composants interagissant sur le composant C^i considéré. En effet, si un composant C^j est anormalement utilisé et que celui-ci interagit avec le composant C^i, le composant C^j peut avoir un impact sur le vieillissement du composant C^i et par conséquent modifier les probabilités de défaillance des fonctions implémentées par ce composant C^i. Les interactions entre un composant C^i et le reste du système sont considérées comme un facteur de stress externe dès lors qu'un composant interagissant est anormalement sollicité. Typiquement une AMDEC étendue peut être utilisée pour quantifier ces interactions entre les composants du système. Dans le cadre de notre modélisation générique, les interactions entre les composants du système sont modélisées par des paramètres partagés (sortie/entrée) dans le système :

$$\mathcal{IT} = ar_{\mathcal{IT}}(op^{i,k}, \ldots, op^{j,l}) \mid \{op^{i,k}, \ldots, op^{j,l}\} \cap \mathcal{OP} = \emptyset. \tag{4.15}$$

- Il est possible d'estimer le stress futur induit par un ensemble de missions possibles. Une mission possible est un objectif que le système doit atteindre avant la prochaine phase de maintenance. Elle peut donc être représentée par la réalisation d'une ou plusieurs fonctions objectif du système. Les conditions opérationnelles d'un système sont différentes selon la mission considérée et induisent un stress plus ou moins élevé sur les composants. Par exemple l'usure des moteurs d'un avion est accélérée lorsque celui-ci traverse un espace aérien pollué. La pureté de l'air a une influence sur le vieillissement d'un moteur d'avion. Si une mission est d'effectuer un vol Paris-Dakar, les conditions opérationnelles ne seront pas les mêmes que pour une mission correspondant au vol Paris-Tokyo, l'espace aérien au niveau de l'océan indien et du continent asiatique étant beaucoup plus pollué qu'en France et en Afrique. Il faut pouvoir quantifier le stress $S(M_i)$ induit par les conditions opérationnelles des composants associée à une future mission M_i. Les futures missions peuvent être alors réparties dans différentes classes selon le niveau de stress qu'elles vont induire sur le système. L'ensemble des classes de missions est noté $\mathcal{CM}$. À chaque classe de missions va correspondre un niveau de stress et donc une valeur de $\eta_{pp^{i,k}}$.

Connaissant la mission que le système doit réaliser, et les connexions entre les composants, s'il est possible de mesurer les conditions environnementales du système et de fournir un diagnostic le moins ambigu possible sur l'état des composants du système, la caractéristique $\eta_{pp^{i,k}}$ des distributions de probabilité de faute associée à chaque paramètre $pp^{i,k}$ peut être déterminée et mise à jour lorsque de nouvelles informations (observations) sont disponibles en ligne. Les $ettf$ et les caractéristiques $\{\eta_F\}$ associées aux fonctions élémentaires $\{F\}$ de C^i peuvent alors être calculées.

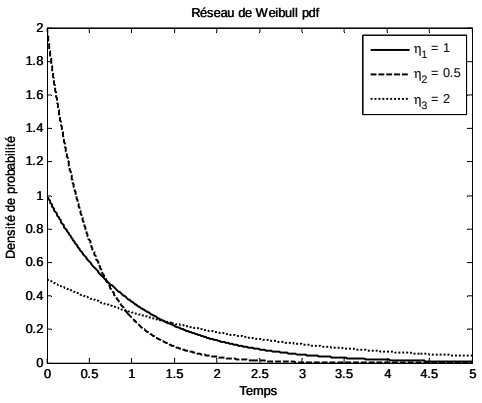

FIG. 4.5 – Densités de probabilité de Weibull associées aux différentes classes de missions

La fonction de pronostic d'un module local de pronostic pour un composant C^i calcule pour chaque paramètre privé de $\mathcal{PP}^i$ un ensemble de fonctions de densité de probabilité de Weibull qui correspond aux différentes valeurs de $\eta_{pp^{i,k}}$ pour les différentes classes de missions comme illustré sur la figure 4.5. Les caractéristiques $\{\eta_F\}$ des distributions de probabilité de défaillance associées aux fonctions élémentaires mises en œuvre par le composant C^i sont directement déterminées à partir des caractéristiques $\{\eta_{pp^{i,k}}\}$ telles que $pp^{i,k} \in PP(F)$.

4.3.3 RUL d'un composant

Si un composant ne fonctionne correctement alors toutes les fonctions élémentaires qu'il met en œuvre sont disponibles. Les fonctions élémentaires d'un composant reposent sur un ensemble de paramètres privés. Ces paramètres privés ne doivent pas être en dehors de leur rang nominal pour que les fonctions élémentaires soient disponibles sur le composant. Lorsqu'un paramètre privé pp est en faute (en dehors de son rang nominal), au moins une fonction élémentaire $Fu^{i,k}$ n'est plus disponible sur le composant C^i. Le composant C^i est défaillant et doit être remplacé.

Le RUL d'un composant est le temps qu'il reste jusqu'à ce que le composant atteigne une probabilité de défaillance P_F non acceptable pour le système d'aide à la décision. Ce seuil de probabilité dépend des seuils P_{max} fixés pour chaque paramètre $pp^{i,k}$ de

$PP(F)$. Le RUL d'un composant C^i est donc évalué à la valeur minimale des *ettf* des fonctions élémentaires qu'il met en œuvre :

$$\begin{aligned} &RUL(C^i) = \min(ettf(Fu^{i,j}) \mid Fu^{i,j} \in \mathcal{FU}^i), \\ &\text{avec} \quad ettf(Fu^{i,j}) = \min(rtf(pp^{i,k}) \mid pp^{i,k} \in PP(Fu^{i,j}). \end{aligned} \tag{4.16}$$

Cette seconde équation n'est vraie que sous l'hypothèse que les paramètres privés de $PP(Fu^{i,j})$ ne sont pas redondants dans le composant C^i. Le $ettf$ d'une fonction élémentaire F peut également être déterminé à partir de la distribution de probabilité de défaillance de Weibull associée à F :

$$ettf(F) = t_F \;\text{ tel que } \int_0^{t_F} \frac{\beta_F}{\eta_F} e^{-\left(\frac{t}{\eta_F}\right)^{\beta_F}} dt, \tag{4.17}$$

sachant que P_F est calculé à partir des seuils de probabilité de faute des paramètres de $PP(F)$ et que η_F provient d'une combinaison des $\{\eta_{pp^{i,k}}\} \in PP(F)$.

4.4 Composition des pronostics de fonctions pour le RUL du système

Les concepts introduits dans la section précédente concernent un composant C^i et ses fonctions élémentaires $\mathcal{FU}^i$. Le challenge consiste maintenant à déterminer le RUL global d'un système complexe Σ [GOE 07]. Le RUL d'un système correspond au temps qu'il reste jusqu'à ce que le système ne puisse plus assurer l'ensemble complet de ses fonctions objectif $\mathcal{FU}_s$ (voir paragraphe 1.4.1, page 28). Cette durée est donc évaluée à partir des $ettf(Fu_s^i)$ de chaque fonction objectif $Fu_s^i \in \mathcal{FU}s$. Il faut donc obtenir un pronostic pour chaque fonction objectif à partir des pronostics réalisés au niveau des fonctions élémentaires des composants.

4.4.1 Composition des pronostics fonctionnels

Le $ettf(Fu_s^i)$ d'une fonction objectif Fu_s^i dépend des *ettf* des fonctions élémentaires dont la disponibilité est nécessaire pour réaliser Fu_s^i. Les *ettf* des fonctions élémentaires sont évalués à partir de leurs fonctions de densités de probabilité de défaillance. Le $ettf(Fu_s^i)$ de la fonction objectif Fu_s^i sera donc évalué à partir d'une composition des espérances (les valeurs moyennes) des distributions de Weibull associées aux fonctions élémentaires requises.

Quel que soit le modèle de vieillissement (connaissance profonde physique ou MTTF) d'un paramètre privé $pp^{i,k}$, la probabilité de faute du paramètre peut toujours être représentée par une pdf de Weibull. Pour les modèles de Weibull considérés, l'espérance mathématique peut être approximée par la valeur de la caractéristique $\eta_{pp^{i,k}}$. La caractéristique η_F d'une fonction F correspond à la valeur minimale des $\{\eta_{pp^{i,k}}\}$ tels que

$pp^{i,k} \in PP(F)$. Comme le $ettf(F)$ d'une fonction élémentaire F peut être déterminé à partir d'une fonction positive croissante et monotone de la caractéristique η_F, le raisonnement qui doit être fait sur le *ettf* peut tout aussi bien être fait sur la caractéristique η_F. Lorsque la caractéristique η_F du modèle de Weibull augmente, le $ettf(F)$ de la fonction F est rallongé et lorsque η_F diminue, le $ettf(F)$ est raccourci.

La composition des pronostics de fonctions repose sur les dépendances fonctionnelles telles qu'elles sont décrites dans le modèle fonctionnel du système à l'aide des applications $Pred$ et n/m (voir la section 2.2.3.2, page 52). La méthode présentée ci-dessous n'est valide que pour une seule classe de missions. Elle doit donc être répétée pour chaque classe de missions.

Dans le cas où une fonction Fu^i est réalisée seulement si toutes les fonctions de $Pred(Fu^i)$ sont disponibles, la caractéristique $\eta(Fu^i)$ de la distribution de Weibull associée à la fonction Fu^i prend la valeur minimale des caractéristiques d'échelle des distributions de probabilité associées aux fonctions de $Pred(Fu^i)$:

$$\eta(Fu^i) = \min_{Fu^j \in Pred(Fu^i)} [\eta(Fu^j)]. \tag{4.18}$$

Dans le cas où la fonction Fu^i est réalisée si au moins une fonction de $Pred(Fu^i)$ est disponible, la caractéristique $\eta(Fu^i)$ prend la valeur maximale des caractéristiques associées aux fonctions de $Pred(Fu^i)$:

$$\eta(Fu^i) = \max_{Fu^j \in Pred(Fu^i)} [\eta(Fu^j)]. \tag{4.19}$$

Par généralisation, la réalisation d'une fonction Fu^i peut s'écrire en utilisant l'application n/m, le η de la fonction Fu^i peut alors s'écrire de la façon suivante :

$$\eta(Fu^i) = \max_{X \in n/m[Pred(Fu^i)]} \left[\min_{Fu^j \in X} \eta(Fu^j) \right],$$

où $\|X\| = n$ et $\|Pred(Fu^i)\| = m$. Il faut considérer l'ensemble de n fonctions dont les caractéristiques η sont les plus grandes puis conserver ensuite la valeur minimale de leurs η. Cette technique permet de calculer la caractéristique η de chaque fonction intermédiaire puis de les composer de manière à obtenir le *ettf* des fonctions objectif du système.

Le $ettf(Fu_s^i)$ de chaque fonction du système Fu_s^i est en fait définit par la valeur minimale des *ettf* des fonctions élémentaires dont la disponibilité est nécessaire pour réaliser F_s^i et qui sont implémentées par des composants non redondants dans le système.

4.4.2 RUL du système

Jusqu'à maintenant, un ensemble de caractéristiques η était associé à chaque fonction objectif du système. Chaque caractéristique η est définie pour un niveau de stress

d'une classe de missions considérée. Afin de calculer le RUL du système, il faut maintenant prendre en compte la future mission que le système doit effectivement accomplir. Une seule caractéristique η correspondant au niveau de stress sélectionné sera associée à chaque fonction objectif du système.

Le RUL d'un système Σ correspond au temps qu'il reste jusqu'à ce que le système ne puisse plus réaliser l'ensemble de ses fonctions objectif avec succès. Le RUL du système Σ est alors évalué à partir de la valeur minimale des *ettf* des fonctions objectif de $\mathcal{FU}_s$ qu'il doit fournir à son environnement :

$$RUL(\Sigma) = \min(ettf(Fu_s^i) \mid \ Fu_s^i \in \mathcal{FU}_s). \tag{4.20}$$

La maintenance peut alors être programmée en tenant compte de la valeur minimale des $ettf(Fu_s^i)$ avec $Fu_s^i \in \mathcal{FU}_s$. L'action de maintenance consiste à remplacer au moins un composant C^i qui met en œuvre une fonction élémentaire $F \in \mathcal{FU}^i$ par un nouveau composant de telle sorte que $ettf(Fu_s^i)$ soit augmenté et par conséquent que le RUL du système soit plus long.

4.5 Conclusion

Nous avons défini une fonction générique de pronostic qui utilise le modèle probabiliste de Weibull pour représenter la probabilité de faute de chaque paramètre privé des composants quels que soient les modèles de vieillissement disponibles. Les composants étant parfois redondants, il est nécessaire de fournir un pronostic pour chaque fonction réalisée par le système afin de pouvoir établir une probabilité de défaillance du système.

La fonction de pronostic fournit un ensemble de fonctions de densité de probabilité de défaillance correspondant à chacune des fonctions élémentaires mises en œuvre par les composants du système. Lorsque les modèles de vieillissement des paramètres le permettent, les fonctions de pronostic peuvent être adaptatives et tenir compte des facteurs de stress qui agissent réellement sur les composants du système. Les pronostics des fonctions sont ensuite composés de manière à obtenir le RUL du système. La composition des pronostics s'appuie sur le modèle fonctionnel du système.

5 Critères de performance pour l'architecture de supervision

Résumé : Ce chapitre présente trois propriétés qui permettent d'évaluer la performance d'un module de supervision : la diagnosticabilité, la pronosticabilité et la précision du diagnostic et du pronostic. Nous proposons d'établir un retour sur conception du système complexe afin d'en améliorer sa diagnosticabilité. Une méthodologie guidée par la propriété de précision est développée pour caractériser automatiquement des recommandations de conception dans le cadre particulier des systèmes à événements discrets distribués.

5.1 Introduction

Le chapitre 3 présente une caractérisation des résultats de diagnostic et de pronostic pour un système complexe. Ces résultats sont établis par les fonctions de diagnostic et de pronostic qui sont intégrées dans un module de supervision. Ils sont assortis de critères de performance afin d'évaluer l'efficacité du module de supervision et d'apprécier les résultats obtenus. Ces critères de performance reposent sur différentes propriétés qui peuvent être vérifiées par les fonctions de diagnostic et de pronostic : la diagnosticabilité, la pronosticabilité et la précision.

Une simple vérification de ces propriétés n'est souvent pas suffisante. Pour assurer un certain niveau de performance du module de supervision, il faut analyser le système complexe dès sa conception. Nous proposons d'établir des retours sur conception pour le système afin de garantir ou d'améliorer son degré de diagnosticabilité. Une méthodologie guidée par la propriété de précision du diagnostic est développée dans le cadre particulier des systèmes à événements discrets distribués dans le but de fournir des recommandations de conception pour la diagnosticabilité.

5.2 Critères de performance

Des critères de performance sont définis pour évaluer l'efficacité des fonctions de diagnostic et de pronostic. Les propriétés de diagnosticabilité, de pronosticabilité et de précision sont caractérisées à partir du formalisme introduit dans les chapitres 2 et 3.

5.2.1 Diagnosticabilité

Une fonction de diagnostic a pour but de diagnostiquer les fautes apparaissant dans le système. La fonction de diagnostic retourne, quelles que soient les observations reçues, le résultat : "chaque composant peut être dans un mode de faute". Ce diagnostic est évidemment correct mais totalement inutile. Les observations ne peuvent pas aider à déterminer quel composant est réellement en faute, le diagnostic est alors totalement ambigu. La fonction de diagnostic n'est pas du tout efficace car elle ne peut pas diagnostiquer les fautes apparaissant sur les composants avec certitude.

Un moyen de mesurer l'efficacité et d'établir les limites d'une fonction de diagnostic est de considérer la propriété de diagnosticabilité. La diagnosticabilité est une mesure de la capacité d'une fonction de diagnostic Δ^{Σ} à diagnostiquer les fautes apparaissant dans un système Σ à partir des observations disponibles OBS du système Σ. La diagnosticabilité est également une propriété logique (vraie ou fausse) qui peut être satisfaite par la fonction Δ^{Σ}. Cette propriété est nécessaire pour déterminer les fautes que l'on peut discriminer à l'aide des observations OBS. Une définition générale de la diagnosticabilité est proposée ci-dessous.

Définition 30 (Notion de diagnosticabilité). *La diagnosticabilité correspond à la capacité d'un système et de ses fonctions de surveillance et de diagnostic à exhiber des observations différentes pour chaque situation de faute anticipée.*

Une situation de faute anticipée est une situation de faute F connue et décrite dans le modèle du système complexe Σ sur lequel s'appuie la fonction de diagnostic Δ^{Σ}. La fonction de diagnostic global Δ^{Σ} à un instant $t_{diag} \in [t_j, t_{j+1}]$ détermine l'ensemble des séquences de modes passées du système qui sont cohérentes avec les observations $OBS_{t_{diag}}$ du système disponibles à t_{diag} et les modèles des composants (voir la définition 26, page 77) :

$$\Delta_j^{\Sigma} = \{s_1(\Delta_j^{\Sigma}), \ldots, s_n(\Delta_j^{\Sigma})\}, \tag{5.1}$$

où $s_k(\Delta_j^{\Sigma})$ est un candidat de diagnostic, c'est-à-dire une séquence de modes $(m_0^{\Sigma}, \ldots, m_j^{\Sigma})$ pour le système qui est cohérente avec les observations $OBS_{t_{diag}}$.

Un mode de faute m_i^{Σ} est diagnosticable si et seulement si pour toute séquence du système $(m_0^{\Sigma}, \ldots, m_p^{\Sigma})$ contenant le mode m_i^{Σ}, il existe un délai d_{max}^i tel que le mode m_i^{Σ} appartient à tous les candidats de diagnostic global proposés par la fonction Δ^{Σ} après avoir observé OBS_{t_j} tel que $t_j > t_i + d_{max}^i$. Notons $\Delta^{\Sigma}(OBS_{t_j})$, le résultat obtenu par la fonction de diagnostic après avoir observé OBS_{t_j}.

Définition 31 (Diagnosticabilité d'un mode). *Soit $(m_0^{\Sigma}, \ldots, m_i^{\Sigma})$ une séquence de modes du système Σ, soit OBS_{t_i} l'ensemble des observations de Σ disponibles et cohérentes avec la séquence $(m_0^{\Sigma}, \ldots, m_i^{\Sigma})$, le mode de faute m_i^{Σ} est diagnosticable si et seulement si pour toute évolution possible $(m_{i+1}^{\Sigma}, \ldots, m_j^{\Sigma})$ du système Σ, il existe un délai fini d_{max}^i tel que si $t_j > t_i + d_{max}^i$ alors $\forall k, m_i^{\Sigma} \in s_k(\Delta^{\Sigma}(OBS_{t_j}))$.*

Cette définition est une adaptation dans le formalisme des modes de la définition classique de la diagnosticabilité pour les SED introduite par [SAM 95]. Un mode est diagnosticable lorsque son occurrence est toujours diagnostiquée sans ambiguïté après un délai d^i_{max} à partir des observations du système.

La définition qui vient d'être donnée pour la diagnosticabilité d'un mode m^Σ_i est statique. C'est-à-dire qu'elle ne prend pas en compte la séquence $(m^\Sigma_0, \ldots, m^\Sigma_i)$ ayant mené le système dans ce mode m^Σ_i. Or il peut être intéressant de connaître l'enchaînement des fautes pour déterminer celle qui est à l'origine de la séquence de modes de faute suivie par le système. La notation $s \subseteq s'$ signifie que la séquence de modes s est une partie de la séquence de modes s', par exemple $(m^\Sigma_2, m^\Sigma_3) \subseteq (m^\Sigma_0, m^\Sigma_1, m^\Sigma_2, m^\Sigma_3)$. Cette notation permet d'introduire la définition de la diagnosticabilité d'une séquence de modes du système. Une séquence de modes de faute $s = (m^\Sigma_l, \ldots, m^\Sigma_i)$ est diagnosticable si et seulement si pour toute séquence du système $(m^\Sigma_0, \ldots, m^\Sigma_i)$ contenant la séquence s, il existe un délai d^i_{max} tel que la séquence s appartient à tous les candidats de diagnostic global proposés par la fonction Δ^Σ après avoir observé OBS_{t_j} tel quel $t_j > t_i + d^i_{max}$.

Définition 32 (Diagnosticabilité d'une séquence de modes). *Soit $(m^\Sigma_0, \ldots, m^\Sigma_i)$ une séquence de modes du système Σ, soit OBS_{t_i} l'ensemble des observations de Σ disponibles et cohérentes avec la séquence $(m^\Sigma_0, \ldots, m^\Sigma_i)$, la séquence de modes de faute $s = (m^\Sigma_l, \ldots, m^\Sigma_i)$ est diagnosticable si et seulement si pour toute évolution possible $(m^\Sigma_{i+1}, \ldots, m^\Sigma_j)$ du système Σ, il existe un délai fini d^i_{max} tel que si $t_j > t_i + d^i_{max}$ alors $\forall k, s \subseteq s_k(\Delta^\Sigma(OBS_{t_j}))$.*

Lorsque la totalité de la séquence $(m^\Sigma_0, \ldots, m^\Sigma_i)$ est diagnosticable, elle est déterminée sans ambiguïté par la fonction de diagnostic et correspond à la séquence de modes réellement suivie par le système $(mr^\Sigma_0, \ldots, mr^\Sigma_i)$.

Lorsque le dernier mode de la séquence n'est pas diagnostiqué avec certitude par la fonction de diagnostic, c'est-à-dire qu'il y a une ambiguïté sur le mode courant du système, une stratégie de tests et de réparations doit être utilisée afin d'identifier les composants qui sont réellement en faute. Cette phase de test et de réparation peut être très longue. Pour optimiser la maintenance, on souhaite minimiser le nombre d'opérations de test et réparer uniquement les composants qui sont réellement en faute. Il faut pour cela réduire le nombre de candidats de diagnostic, ce qui revient à améliorer la diagnosticabilité. Quand un système est diagnosticable, les composants en faute sont directement déterminés à partir des observations. Le choix d'une action de maintenance est totalement déterminé, la maintenance corrective est alors optimale.

5.2.2 Pronosticabilité

Une fonction de pronostic a pour but de pronostiquer les défaillances d'un système, c'est-à-dire de prédire la date de perte de fonctions objectifs d'un système. Pour celà,

elle calcule l'ensemble des séquences futures de modes qui sont cohérentes avec les candidats déterminés par la fonction de diagnostic global et évalue celle qui a la plus grande probabilité de se produire d'après les modèles de vieillissement des paramètres privés des composants. Un modèle de vieillissement d'un paramètre privé peut être représenté par une valeur moyenne de la date de la première défaillance (MTTF) ou par une relation entre des paramètres représentant les conditions opérationnelles du composant.

Définition 33 (Notion de pronosticabilité). *La pronosticabilité correspond à la capacité d'une fonction de pronostic à prédire la date d'une défaillance anticipée du système à partir d'une connaissance sur le vieillissement des paramètres de ses composants.*

La propriété de pronosticabilité repose sur la disponibilité des modèles de vieillissement des paramètres des composants. Une défaillance Def est pronosticable à partir d'un mode quelconque m_i^Σ du système si l'ensemble des modèles de vieillissement nécessaires pour prédire toutes les séquences futures de modes de fautes possibles à partir de m_i^Σ et menant à cette défaillance Def est disponible. La pronosticabilité est une propriété logique : $Pron(Def, m_i^\Sigma) = \{vrai, faux\}$. Une défaillance Def est pronosticable à partir d'un résultat de diagnostic global $\Delta^\Sigma = \{m_i^\Sigma\}$ si l'ensemble des modèles de vieillissement nécessaires pour prédire toutes les séquences futures de modes de fautes possibles à partir des modes $\{m_i^\Sigma\}$ et menant à cette défaillance Def est disponible.

Définition 34 (Pronosticabilité d'une défaillance). *Une défaillance Def est pronosticable à partir d'un résultat de diagnostic $\Delta^\Sigma = \{m_i^\Sigma\}$ ssi $\forall i, \quad Pron(Def, m_i^\Sigma) = vrai$.*

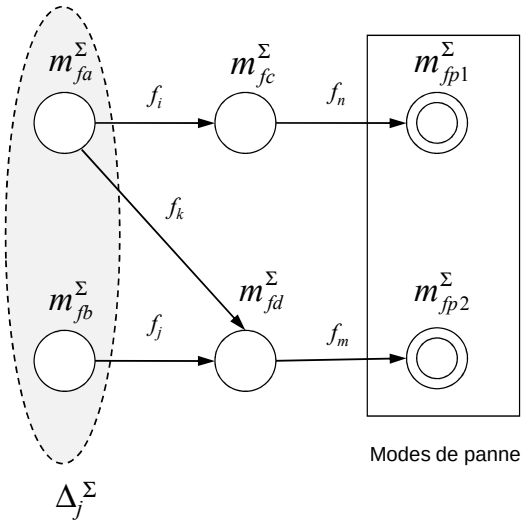

Fig. 5.1 – Exemple pour la pronosticabilité

La propriété de pronosticabilité est exprimée sur l'exemple d'une sous-partie de l'automate des modes d'un système illustré sur la figure 5.1. Les états de l'automate représentent des modes de faute simple ou multiple (c'est-à-dire des modes du système

dans lesquels plusieurs paramètres privés des composants sont en dehors de leur rang nominal). Les états de panne représentent des modes de fautes qui sont par définition associés à des défaillances (c'est-à-dire à la perte d'au moins une fonction objectif du système) : par exemple le mode de faute m^{Σ}_{fp1} est associé à une défaillance Def_1 et le mode de faute m^{Σ}_{fp2} est associé à une défaillance Def_2. Une faute f_j provoquant un changement de mode du système est directement associée à la déviation d'un paramètre privé pp^j. À un instant $t_{diag} \in [t_j, t_{j+1}]$, le résultat de diagnostic fourni par la fonction Δ^{Σ}_j est ambigu : $\Delta^{\Sigma_j} = \{m^{\Sigma}_{fb}, m^{\Sigma}_{fa}\}$. Considérons qu'une connaissance est disponible sur les lois de vieillissement des paramètres pp^i, pp^n, pp^j et pp^m (associés aux fautes f_i, f_n, f_j et f_m pouvant survenir dans le système),

- la défaillance Def_1 est pronosticable à partir du diagnostic Δ^{Σ}_j : les modèles de vieillissement des paramètres pp^i et pp^n sont disponibles et permettent de prédire l'unique séquence future de modes de faute possible $(m^{\Sigma}_{fc}, m^{\Sigma}_{fp1})$ menant à la défaillance Def_1 à partir du résultat fourni par la fonction Δ^{Σ}_j ;
- la défaillance Def_2 n'est pas pronosticable à partir du diagnostic Δ^{Σ}_j : aucune connaissance sur le vieillissement de pp^k n'est disponible, il n'est pas possible de prédire la date à laquelle le système va passer dans le mode de faute $m^{\Sigma}_{f_d}$ qui peut ensuite provoquer la défaillance Def_2. La défaillance Def_2 aura certainement lieu mais il est impossible de prédire quand. Si le diagnostic était $\Delta^{\Sigma}_j = \{m^{\Sigma}_{fb}\}$, la connaissance sur le vieillissement de pp^j et pp^m suffirait à pronostiquer la défaillance Def_2.

Il est possible d'établir un degré de pronosticabilité. Une défaillance est partiellement pronosticable à partir d'un résultat de diagnostic, s'il est possible d'affirmer avec certitude qu'elle aura lieu dans le futur mais que les modèles de vieillissement disponibles se sont pas suffisants pour déterminer sa date d'occurrence. Reprenons l'exemple de la figure 5.1, si la fonction de diagnostic retourne $\Delta^{\Sigma}_j = \{m^{\Sigma}_{fb}\}$ et que seule la connaissance sur le vieillissement de pp^j est disponible, la défaillance Def_2 est dite partiellement pronosticable. Elle aura forcément lieu mais on ne sait pas quand. On peut tout de même dans ce cas-là, déterminer une date minimale d'occurrence pour Def_2 à partir de la probabilité de faute de f_j.

5.2.3 Précision

La diagnosticabilité, telle qu'elle est définie dans la section 5.2.1, est analysée dans le cas où l'algorithme de diagnostic Δ^{Σ} doit diagnostiquer avec certitude chaque faute apparaissant dans le système en considérant l'ensemble complet des observations du système OBS. L'algorithme de diagnostic Δ^{Σ} fournit alors un résultat global. Pour des systèmes complexes qui peuvent être distribués, il est plus naturel et moins complexe de considérer un algorithme de diagnostic décentralisé ou distribué. La section 3.4.1 présente la caractérisation d'un diagnostic au niveau des composants du système. Il est également possible d'établir des diagnostics locaux au niveau de sous-systèmes à

partir d'observations locales à ces sous-systèmes. La performance d'une architecture de supervision peut être évaluée par une étude de précision des algorithmes de diagnostic et de pronostic.

5.2.3.1 Précision du diagnostic

La précision d'un diagnostic aide à déterminer les sous-systèmes qu'il est suffisant de surveiller pour diagnostiquer les fautes dans le système. Soit un sous-système γ de m composants et Δ^γ, l'algorithme de diagnostic sur γ, l'ensemble des observations du sous-système est noté OBS^γ.

Définition 35 (Notion de précision d'un diagnostic). *Le diagnostic Δ^γ d'un sous-système γ est précis pour une faute F si les observations locales OBS^γ sont suffisantes pour établir un diagnostic local qui est globalement cohérent.*

Un résultat de diagnostic local Δ^γ_j pour un sous-système γ à un instant $t_j \in [t_0, t_{diag}]$ est un ensemble de candidats de diagnostic local cohérents avec les observations OBS^γ_j et les modèles des composants de γ (comme défini pour le système global dans le paragraphe 3.4.2, page 75). Le diagnostic du sous-système γ est précis si les candidats de diagnostic déterminé par la fonction Δ^γ sont toujours globalement compatibles (voir la définition 23, page 75).

Définition 36 (Précision forte d'un diagnostic). *Soit un sous-système γ, et une fonction de diagnostic local Δ^Σ sur ce sous-système, le diagnostic local est précis sur ce sous-système si et seulement si les candidats de diagnostic locaux sont toujours globalement compatibles :*

$$\forall j \in [t_0, t_{diag}],\ \forall (m_x^1, \ldots, m_y^m) \in \Delta^\gamma_j,\quad (m_x^1, \ldots, m_y^m) \in \Delta^\Sigma_j. \tag{5.2}$$

La précision d'un diagnostic local est une propriété très forte. A tout instant depuis la mise en service du système, le diagnostic local est forcément globalement cohérent. Il est possible de nuancer cette propriété en introduisant un délai d^j_{max} pour vérifier la condition de cohérence globale des candidats déterminés par l'algorithme de diagnostic local Δ^γ.

Définition 37 (Précision faible d'un diagnostic). *Soit un sous-système γ, et une fonction de diagnostic local Δ^γ sur ce sous-système, le diagnostic local est précis sur ce sous-système si et seulement si les candidats de diagnostic locaux sont globalement cohérents après un temps fini :*

$$\forall j \in [t_0, t_{diag}],\quad \exists d^j_{max} \mid \Delta^\gamma_{j+d^j_{max}} \subseteq \Delta^\Sigma_{j+d^j_{max}}. \tag{5.3}$$

Si le diagnostic local d'un sous-système γ est précis, les observations locales OBS^γ sont suffisantes pour diagnostiquer les fautes apparaissant sur les composants de γ. La

fonction de surveillance n'a alors pas besoin d'informations provenant des autres composants du système. À chaque nouvelle observation reçue, un diagnostic local globalement cohérent est obtenu. Cette propriété améliore l'efficacité de la fonction de diagnostic en minimisant les communications et le nombre de ressources nécessaires pour la fonction de surveillance et la fonction de diagnostic.

5.2.3.2 Précision du pronostic

La précision d'un pronostic détermine l'ensemble des paramètres privés dont il est suffisant de connaître les modèles de vieillissement pour pronostiquer les défaillances du système (évaluer la date des défaillances potentielles). La défaillance Def_F est associée à la perte d'une fonction objectif F du système.

Définition 38 (Notion de précision d'un pronostic). *Le pronostic local Π^γ d'un sous-système γ est précis pour une fonction F si les modèles de vieillissement des paramètres privés de γ sont disponibles et permettent de prédire la séquence future de modes menant à la défaillance Def_F.*

Si un pronostic local Π^Σ est précis pour une fonction F, alors les modèles de vieillissement disponibles pour les paramètres privés de γ sont suffisants pour établir un pronostic de la fonction F, c'est-à-dire prédire la séquence future de modes menant à la défaillance Def_F. L'ensemble des paramètres privés des composants du sous-système γ est noté $\mathcal{PP}^\gamma$ et l'ensemble des paramètres privés sur lesquels reposent la réalisation de la fonction F est noté $PP(F)$.

Définition 39 (Précision d'un pronostic). *Soit un sous-système γ, une fonction F et un algorithme de pronostic local Π^γ sur γ, le pronostic local est précis sur ce sous-système pour la fonction F si et seulement si $PP(F) \subseteq \mathcal{PP}^\gamma$.*

5.3 Amélioration de la diagnosticabilité par un retour sur conception

Lorsque l'objectif est d'optimiser la maintenance, la propriété de diagnosticabilité devient très importante. Des travaux antérieurs présentent des méthodes permettant de vérifier si une faute est diagnosticable ou pas [SAM 95] [DEB 02] [CON 02] [LAM 03]. Mais de nos jours, avec la complexité grandissante des nouveaux systèmes, la simple vérification de cette propriété n'est plus suffisante. Il devient nécessaire de déterminer les causes de la non diagnosticabilité d'une faute et de proposer des solutions à la conception qui permettent de les éliminer. On introduit le problème de la caractérisation automatique de retours sur conception du système afin d'en améliorer son degré de diagnosticabilité.

5.3.1 Conception pour la diagnosticabilité

L'objectif consiste à établir des recommandations de conception pour les composants à surveiller qui, une fois implantés, garantiront un diagnostic compatible avec des actions de maintenance. On propose de fournir de légères modifications dans les spécifications des composants.

5.3.1.1 Types de modifications pour la conception du système

Le premier type d'opération consiste à améliorer l'observabilité d'un sous-système γ en sélectionnant certains types de capteurs et en optimisant leur nombre et leur positionnement sur les composants de γ. Les capteurs que l'on ajoute sont supposés fiables et non bruités. Ce problème est similaire au problème de sélection de capteurs qui est étudié dans [DEB 99] ou [JIA 03]. Le second type d'opération que l'on autorise modifie la structure du sous-système γ et agit donc directement sur son comportement. Les différentes opérations possibles à partir d'une spécification d'un système (des modèles préexistants) sont énumérées ci-dessous. Toutes ces modifications ont une signification physique. L'ensemble des paramètres d'un sous-système γ est noté $\mathcal{P}^\gamma$.

1. Observer un paramètre $p^{i,k}$ (qui peut être un paramètre privé $pp^{i,k}$) d'un composant du sous-système γ : $p^{i,k} \in \mathcal{P}^\gamma \cap OBS^\gamma$. Rendre un paramètre observable signifie ajouter un capteur qui est capable de mesurer la valeur de ce paramètre dans le composant C^i du sous-système γ étudié.
2. Observer un paramètre $p^{j,k}$ d'un autre composant C^j qui interagit avec γ : $p^{j,k} \in \mathcal{P}^j \cap OBS^j$, tel que $p^{j,k} \notin \mathcal{P}^\gamma$ et $C^j \notin \gamma$. L'ensemble des paramètres du composant C^j est noté $\mathcal{P}^j$ et ses observations sont représentées par OBS^j. Pour observer un paramètre sur un autre composant du système, il est suffisant de placer un capteur sur ce composant et de considérer un protocole de communication pour l'algorithme de surveillance.
3. Observer un paramètre $p^{i,k}$ partagé par le sous-système γ et un autre composant C^j : $p^{i,k} \in \mathcal{P}^\gamma \cap \mathcal{P}^j \cap OBS^\gamma \cap OBS^j$, et $C^j \notin \gamma$. Rendre un paramètre partagé observable consiste à placer, par exemple, un capteur sur le bus de communication entre deux composants ou sur un intergiciel.
4. Observer un paramètre $pp^{i,k} \in \mathcal{P}^\gamma$ seulement si des conditions prédéterminées *cond* sont vérifiées en introduisant deux nouveaux paramètres booléens *cond* : $p^{i,k} \in OBS^\gamma$ et $\neg cond$: $p^{i,k} \notin OBS^\gamma$. Pour observer un paramètre dans des conditions données, il faut utiliser un capteur spécifique qui peut être contrôlé en ligne en considérant des informations additives. Ce genre de capteur permet l'acquisition active d'information comme dans [THO 07].
5. Ajouter/Enlever (modifier) une relation $ar^{i,k}$ ou un paramètre $pp^{i,k}$ dans le modèle du sous-système γ. De nouveaux capteurs peuvent être générés comme un capteur

d'alarme après une séquence données d'observations par exemple, ou de nouveaux protocoles peuvent être mis en œuvre (protocole de communication).

Un coût est bien évidemment associé à chaque modification sur le système. Ce coût est supposé connu et listé dans un dictionnaire de modifications (voir le tableau 5.1). Dans ce dictionnaire, l'action d'observer un paramètre est notée *Obs* et l'action de rajouter (resp. de enlever) une relation ou un paramètre dans la structure du modèle est notée *Add* (resp. *Del*).

Opération		Coût
1	$Obs(p^{i,k})$, $p^{i,k} \in \mathcal{P}^{\gamma}$	$c_l(p^{i,k})$
2	$Obs(p^{i,k})$, $\exists C^j \mid p^{i,k} \in \mathcal{P}^j$ et $C^j \notin \gamma$	$c_a(p^{i,k})$
3	$Obs(p^{i,k})$, $\exists C^j \mid p^{i,k} \in \mathcal{P}^{\gamma} \cap \mathcal{P}^j$	$c_i(p^{i,k})$
4	$Obs(p^{i,k}, cond)$, $p^{i,k} \in \mathcal{P}^{\gamma}$ et $p^{i,k} \notin OBS^{\gamma}$	$c_c(p^{i,k})$
5	$Add(ar^{i,k})$, $Del(ar^{i,k})$ et $Add(p^{i,k})$, $Del(p^{i,k})$	$c_r(ar^{i,k})$et $c_p(p^{i,k})$

TAB. 5.1 – Dictionnaire des coûts des opérations possibles pour un sous-système γ

Le coût d'un capteur dépend bien évidemment de son type : un capteur de couple moteur ($\simeq$ 860 €) est bien plus coûteux qu'un capteur infrarouge ($\simeq$ 5,50 €). Certaines modifications ne sont pas réalisables comme par exemple l'observation directe de certains paramètres privés qui modélisent la présence de faute : un coût infini leur sera attribué. Le but est de fournir des recommandations aux concepteurs pour rendre le système diagnosticable en minimisant le coût total C_D des opérations sur le système.

5.3.1.2 Spécification de l'architecture de supervision

Il ne suffit pas de positionner intelligemment des capteurs sur le système pour garantir la diagnosticabilité. Un algorithme de diagnostic doit être déployé afin de récupérer les informations enregistrées par la fonction de surveillance à partir des capteurs sur les différents composants du système. L'accès à ces ressources d'informations induit un coût C_S pour la fonction de surveillance. Ce coût dépend du choix du type d'architecture de supervision. On cherche à rendre le système diagnosticable tout en optimisant le coût des modifications à opérer sur le système (le coût associé au placement de capteur par exemple) mais également le coût associé à la fonction de surveillance (coût pour récupérer des observations, coût algorithmique, ressources de calcul).

Il existe principalement trois différents types d'architecture de supervision. Dans une architecture de diagnostic centralisée [SAM 95], toutes les informations provenant des capteurs, les observations des composants, sont rassemblées au même endroit. L'analyse des données est très complexe car elles sont toutes rassemblées par le module de surveillance sans aucun pré-traitement. Beaucoup de ressources mémoire et de communications sont nécessaires et induisent le coût C_S.

Une architecture centralisée n'est pas vraiment appropriée pour des systèmes complexe de grande taille. Il est préférable d'adopter une architecture de supervision décentralisée ou distribuée. Dans une architecture décentralisée [DEB 02; PEN 05], les résultats de diagnostic sont fournis par des diagnostiqueurs locaux qui ne communiquent pas nécessairement entre eux. Ces résultats sont ensuite fusionnés par un coordinateur afin d'établir un diagnostic global pour le système. Dans une architecture distribuée [FAB 05] ou modulaire [CON 06], il existe une fonction de diagnostic (diagnostiqueur) par composant ou par sous-système qui fournit son propre diagnostic. Pour établir des diagnostics locaux globalement cohérents, les diagnostiqueurs ont besoin d'échanger énormément de données. Ces communications induisent un coût de bande passante que l'on inclut dans le coût C_S de l'algorithme de surveillance.

5.3.1.3 Problème d'optimisation de coût

L'objectif est de fournir des recommandations aux concepteurs pour rendre un système diagnosticable tout en minimisant le coût total des opérations à effectuer sur le système qui est noté C_D, mais également le coût C_S lié à l'algorithme de surveillance qui va garantir ce niveau de diagnosticabilité :

$$C_G = \min \sum_{i=1}^{p} (C_{D_i} + C_{S_i}), \tag{5.4}$$

où p est le nombre de fautes anticipées qui peuvent apparaître dans le système et pour lesquelles la diagnosticabilité doit être garantie.

Le challenge est donc de déterminer un compromis entre les deux coûts (le coût du placement de capteurs C_D et le coût de l'architecture de surveillance associée C_S). La mise en œuvre d'un protocole de communication peut être nécessaire pour que le module de surveillance récupère une observation d'un composant afin d'assurer la diagnosticabilité d'un autre composant. Le coût de la procédure algorithmique C_S doit donc également être considéré en plus du coût directement lié aux capteurs afin de communiquer l'information au module de surveillance.

La seconde difficulté dans le problème d'optimisation de coût provient du fait que tous les paramètres privés de $\mathcal{PP}$ dont les déviations peuvent causer une faute doivent être considérés. Puisque les composants du système communiquent par les paramètres partagés, il est intéressant d'observer des interactions même si ce type d'action peut posséder un coût élevé car son observation pourrait être utile pour diagnostiquer plus d'une faute dans le système, cela permettrait alors de réduire le coût lié à la surveillance.

Une méthodologie est nécessaire pour caractériser des retours sur conception tout en optimisant le couple de coûts (C_D, C_S).

5.3.2 Amélioration de la diagnosticabilité dans les SED distribués

Nous proposons une méthodologie dans le cadre des systèmes à événements discrets (SED) qui établit des recommandations de conception afin de garantir la diagnosticabilité d'un SED distribué. La caractérisation des retours sur conception s'appuie sur un problème d'optimisation de coût dans un cadre distribué dans lequel on doit non seulement considérer le coût de conception du système distribué mais également le coût nécessaire pour déployer un algorithme de diagnostic sur ce système.

Le cadre des SED pour le diagnostic à base de modèles a été développé depuis plusieurs années et il est utilisé pour différents types d'applications (réseaux de communication, processus industriels, intergiciels, ...). Un avantage des SED est que de nombreuses propriétés ont déjà été démontrées dans ce cadre. Avant d'introduire la méthodologie, nous présentons une caractérisation d'un SED et nous spécialisons les définitions de diagnostic et de diagnosticabilité dans le cadre des SED distribués.

5.3.2.1 Caractérisation d'un SED

Commençons par établir un lien entre le formalisme générique présenté dans les chapitres 2 et 3 et le formalisme des SED. À l'aide d'un exemple, on montre qu'à partir de la modélisation générique proposée, il est facile de retrouver le problème classique des SED.

Modèle d'un SED distribué

On se place dans le cadre classique des SED pour le diagnostic à base de modèles comme défini dans [SAM 95]. Un système distribué G est un ensemble de N composants $G = \{G_1, G_2, \ldots, G_N\}$ qui interagissent et qui évoluent avec les occurrences d'événements. Une spécification du système distribué est supposée connue avec une analyse a priori de son comportement en présence de faute. Le système distribué peut être modélisé par un ensemble d'automates dans lequel chaque automate représente le modèle d'un composant, un modèle local. La figure 5.2 illustre l'exemple d'un système à trois composants $G = \{G_1, G_2, G_3\}$. Par la suite, selon le contexte, on notera de manière équivalente le système et son modèle G.

Définition 40 (Modèle local d'un composant). *Un modèle local G_i est un automate $G_i = (X_i, \Sigma_i, T_i, x_{0_i})$ dans lequel*
- *X_i est un ensemble fini d'états ;*
- *Σ_i est l'ensemble des événements apparaissant dans G_i ;*
- *$T_i \subseteq X_i \times \Sigma_i \times X_i$ est l'ensemble des transitions ;*
- *x_{0_i} est l'état initial.*

Différents événements peuvent apparaître sur un composant G_i. L'ensemble des évé-

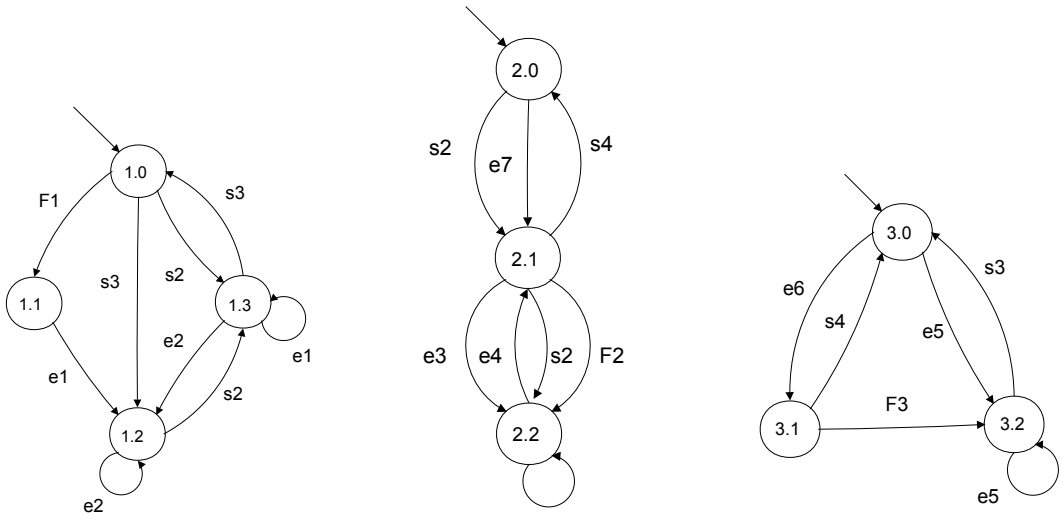

FIG. 5.2 – Exemple SED - Modèles des composants d'un système $G : G_1, G_2, G_3$

nements Σ_i est divisé en deux ensembles disjoints Σl_i et Σc_i. L'ensemble des événements localisés sur un composant (événements locaux ou internes) Σl_i modélisent le comportement interne du composant G_i ($\forall i \neq j, \Sigma_{l_i} \cap \Sigma_{l_j} = \emptyset$). L'ensemble des événements de communication (ou événements d'interaction) Σ_{c_i} modélisent les communications (ou interactions) entre G_i et les autres composants du système. Les fautes pouvant apparaître sur G_i sont modélisées par des événements internes qui doivent être diagnostiqués. Elles appartiennent à l'ensemble que l'on note $\Sigma_{f_i} \subseteq \Sigma_{l_i}$. Un événement e de Σ_i est soit observable ($e \in \Sigma_{o_i}$) soit non observable ($e \in \Sigma_{uo_i}$). Les événements de faute sont supposés non observables sans quoi le problème de diagnostic serait trivial ($\Sigma_{f_i} \subseteq \Sigma_{uo_i}$).

De la modélisation générique aux automates

Un parallèle peut être établi entre les SED de [SAM 95] et le cadre générique défini dans cette thèse. À l'aide d'un exemple, montrons qu'il est facile dans notre formalisme de retrouver le problème classique des SED et de représenter le comportement de chaque composant par un automate.

Considérons l'exemple d'un système d'allumage d'une ampoule à l'aide d'une batterie comme illustré sur la figure 5.3. La batterie est commandée par un interrupteur et fournit, au moyen d'un fil électrique, un courant continu permettant d'allumer une ampoule. Le système est un ensemble de trois composants : $Comps = \{C^1, C^2, C^3\}$ (correspondant respectivement à l'alimentation, au fil électrique et à l'ampoule).

La structure (l'ensemble des paramètres des composants) et les fonctions de base mises en œuvre par les composants du système sont décrites dans un premier temps.

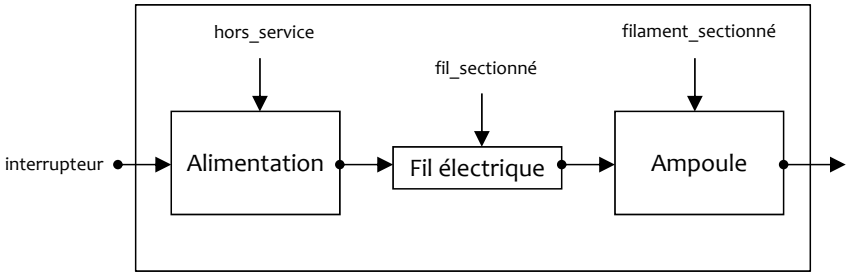

FIG. 5.3 – Système d'allumage

Chaque paramètre est associé à un rang qui est un ensemble fini de modalités.

Modélisation de l'alimentation (C^1) L'alimentation sert à fournir un courant continu aux autres composants du système.

1. Paramètres d'entrée : $\mathcal{IP}^1 = \{\texttt{interrupteur}\}$
   - `interrupteur` : il s'agit du moyen de commander l'alimentation, $r(\texttt{interrupteur}) = \{\texttt{ouvert}, \texttt{ferme}\}$.
2. Paramètres de sortie : $\mathcal{OP}^1 = \{\texttt{sortie\_alimentation}\}$
   - `sortie_alimentation` : soit le courant est présent en sortie de l'alimentation, soit il ne l'est pas, $r(\texttt{sortie\_alimentation}) = \{\texttt{courant}, \texttt{pas\_courant}\}$.
3. Paramètres privés : $\mathcal{PP}^1 = \{\texttt{hors\_service}, \texttt{regime}\}$
   - `hors_service` : il s'agit d'un paramètre de faute qui modélise l'incapacité de l'alimentation à fournir un courant, $r(\texttt{hors\_service}) = \{\texttt{vrai}, \texttt{faux}\}$.
   - `regime` : il s'agit d'un paramètre de fonctionnement interne au composant qui modélise le régime transitoire entre les actions sur l'interrupteur et la sortie du courant, $r(\texttt{regime}) = \{\texttt{permanent}, \texttt{transitoire}\}$.

Le composant C^1 ne dispose que d'une fonction de base dans $\mathcal{FU}^1$, la fonction d'alimentation $Fu_{Alim} \in \mathcal{FU}^1$ qui consiste à fournir le courant en régime permanent quand l'interrupteur est allumé et à ne pas le fournir quand l'interrupteur est éteint. La fonction de base Fu_{Alim} est caractérisée par la condition fonctionnelle suivante :

$$\begin{aligned} Fu_{Alim} \equiv ((&\texttt{sortie\_alimentation} = \texttt{courant} \\ &\Leftarrow (\texttt{regime} = \texttt{permanent} \wedge \texttt{hors\_service} = \texttt{faux} \wedge \texttt{interrupteur} = \texttt{ferme}) \\ &\qquad \wedge (\texttt{sortie\_alimentation} = \texttt{pas\_courant} \\ &\Leftarrow (\texttt{regime} = \texttt{permanent} \wedge \texttt{hors\_service} = \texttt{faux} \wedge \texttt{interrupteur} = \texttt{ouvert})). \end{aligned} \quad (5.5)$$

Lorsque l'alimentation est hors service, elle ne délivre plus de courant quelle que soit la position de l'interrupteur.

- *al* est l'événement "on ferme l'interrupteur",

- *et* est l'événement "on ouvre l'interrupteur",
- *hs* est l'événement "l'alimentation devient hors service",
- *ct* est l'événement "le courant est disponible",
- *non_ct* est l'événement "le courant n'est pas disponible".

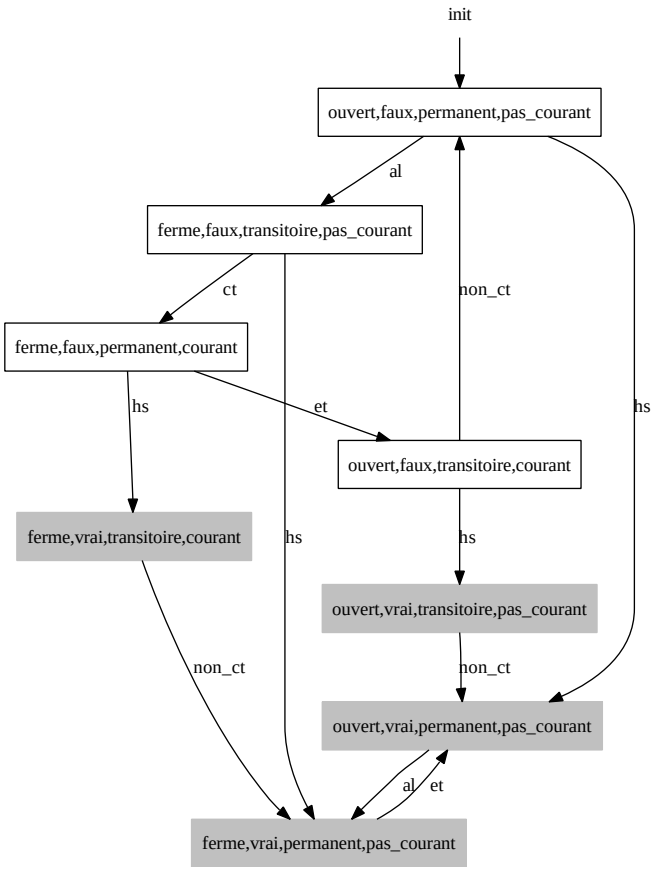

FIG. 5.4 – Modèle de comportement de l'alimentation - C^1

Le comportement de l'alimentation peut être représenté par un automate classique illustré sur la figure 5.4 :

$$G_1 = (X_1, \Sigma_1, T_1, x_{0_1}). \tag{5.6}$$

- X_1 est l'ensemble fini des états. Chaque état est un quadruplet (`interrupteur`, `hors_service`, `regime`, `sortie_alimentation`) qui prend ses valeurs dans l'espace $\{\texttt{ferme}, \texttt{ouvert}\} \times \{\texttt{vrai}, \texttt{faux}\} \times \{\texttt{transitoire}, \texttt{permanent}\} \times \{\texttt{courant}, \texttt{pas\_courant}\}$.
- Σ_1 est l'ensemble des événements de C^1 : $\Sigma = \{al, et, hs, ct, non\_ct\}$.
- T_1 est la fonction de transition définie par $T_1 : X_1 \times \Sigma_1 \rightarrow X_1$.
- $x_{0_1} = \{\texttt{ouvert}, \texttt{faux}, \texttt{permanent}, \texttt{pas\_courant}\}$ est l'état initial.

Dans l'état x_{0_1}, la fonction de base Fu_{Alim} est disponible. Si l'interrupteur est éteint, l'alimentation ne fournit pas de courant : la condition fonctionnelle est vérifiée. L'événement hs est un événement de faute. On peut déterminer deux modes opérationnels pour le composant qui dépendent de la présence de cet événement hs.

- Le mode nominal est défini par l'affectation de la valeur `faux` au paramètre hs ;
- Le mode de faute est défini par l'affectation de la valeur `vrai` au paramètre hs.

Le mode normal est caractérisé dans l'automate par l'ensemble des états à fond blanc et le mode de faute par l'ensemble des états à fond gris.

Modélisation du fil électrique (C^2) Le fil électrique conduit l'électricité fournie par l'alimentation jusqu'à l'ampoule.

1. Paramètres d'entrée : $\mathcal{IP}^2 = \{\texttt{entree\_fil}\}$
   - `entree_fil` : représente la présence ou l'absence de courant en entrée du fil électrique, $r(\texttt{entree\_fil}) = \{\texttt{courant}, \texttt{pas\_courant}\}$.
2. Paramètres de sortie : $\mathcal{OP}^1 = \{\texttt{sortie\_fil}\}$
   - `sortie_fil` : représente la présence ou l'absence d'électricité en sortie du fil électrique, $r(\texttt{sortie\_fil}) = \{\texttt{courant}, \texttt{pas\_courant}\}$.
3. Paramètres privés : $\mathcal{PP}^1 = \{\texttt{fil\_sectionne}, \texttt{regime}\}$
   - `fil_sectionne` : il s'agit d'un paramètre de faute qui modélise l'incapacité du fil à conduire le courant, $r(\texttt{fil\_sectionne}) = \{\texttt{vrai}, \texttt{faux}\}$.
   - `regime` : il s'agit d'un paramètre de fonctionnement interne au composant qui modélise le régime transitoire entre les actions sur l'interrupteur et la sortie du courant, $r(\texttt{regime}) = \{\texttt{permanent}, \texttt{transitoire}\}$.

Le composant C^2 ne dispose lui aussi que d'une fonction de base dans $\mathcal{FU}^2$. La fonction du fil électrique $Fu_{Fil} \in \mathcal{FU}^2$ est de conduire le courant quand l'interrupteur est fermé. La fonction de base Fu_{Fil} est caractérisée par la condition fonctionnelle suivante :

$$\begin{aligned} Fu_{Fil} \equiv (&\texttt{sortie\_fil} = \texttt{courant} \\ \Leftarrow (&\texttt{regime} = \texttt{permanent} \wedge \texttt{fil\_sectionne} = \texttt{faux} \wedge \texttt{entree\_fil} = \texttt{courant}) \\ &\wedge (\texttt{sortie\_fil} = \texttt{pas\_courant} \\ \Leftarrow (&\texttt{regime} = \texttt{permanent} \wedge \texttt{fil\_sectionne} = \texttt{faux} \wedge \texttt{entree\_fil} = \texttt{courant})) \end{aligned} \quad (5.7)$$

- *arrive_ct* est l'événement "le débit de charges électriques augmente dans le fil",
- *stop_ct* est l'événement "le débit de charges électriques diminue dans le fil",
- *coupe* est l'événement "le fil est sectionné".

Le comportement du fil électrique est modélisé par l'automate illustré sur la figure 5.5 :

$$G_2 = (X_2, \Sigma_2, T_2, x_{0_2}). \quad (5.8)$$

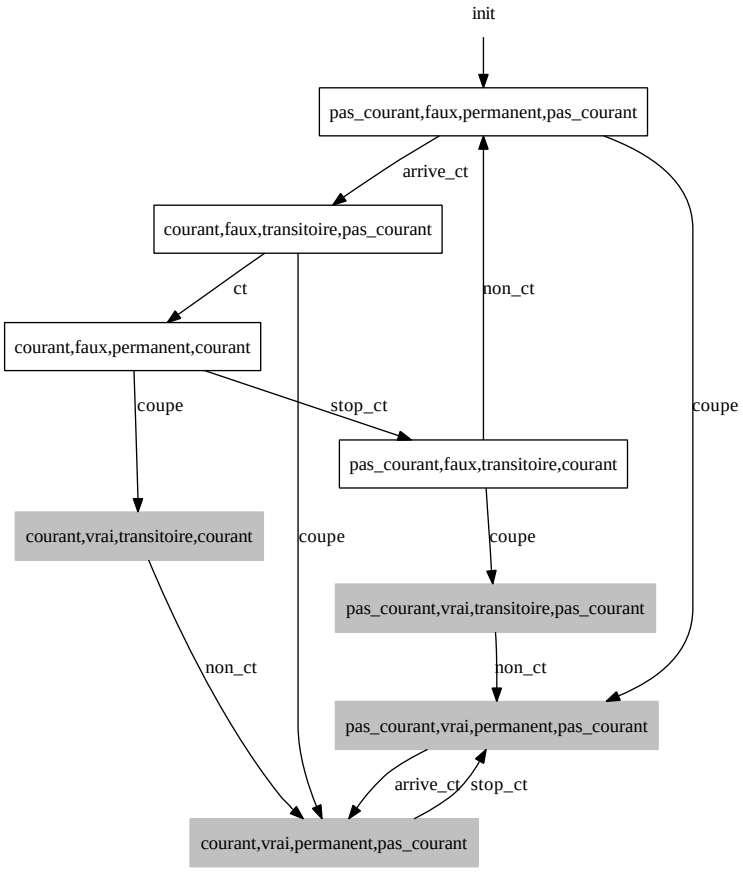

FIG. 5.5 – Modèle de comportement du fil électrique - C^2

- X_2 est l'ensemble fini des états. Chaque état est un quadruplet (`entree_fil`, `fil_sectionne`, `regime`, `sortie_fil`) qui prend ses valeurs dans l'espace {`courant`, `pas_courant`}×{`vrai`, `faux`}×{`transitoire`, `permanent`}×{`courant`, `pas_courant`}.
- Σ_2 est l'ensemble des événements : $\Sigma = \{arrive\_ct, stop\_ct, ct, non\_ct, coupe\}$.
- T_2 est la fonction de transition définie par $T_2 : X_2 \times \Sigma_2 \rightarrow X_2$.
- $x_{0_2} = \{$`pas_courant`, `faux`, `permanent`, `pas_courant`$\}$ est l'état initial.

Pour ce composant C^2, il est également possible de déterminer deux modes de fonctionnement qui dépendent de la présence de l'événement de faute *coupe*. Le mode normal est caractérisé dans l'automate par l'ensemble des états à fond blanc et le mode de faute qui correspond à `fil_sectionne` = `vrai` par l'ensemble des états à fond gris.

En décrivant de la même façon la modélisation structurelle et fonctionnelle du composant C^3, il sera facile de représenter le comportement de l'ampoule par un automate.

Modélisation des interactions On rappelle que les communications (interactions) dans le système sont modélisées par des paramètres (entrée/sortie) partagés par les composants. Le paramètre de sortie de l'alimentation `sortie_alimentation` est connecté au paramètre d'entrée du fil électrique `entree_fil`. Cela signifie que tout événement qui change la valeur du paramètre `sortie_alimentation` est un événement synchronisé avec un événement qui change la valeur du paramètre `entree_fil`. Autrement dit, pour synchroniser les automates G_1 et G_2, on doit synchroniser les événements ct de C^1 et $arrive\_ct$ de C^2 ainsi que non_ct de C^1 et $stop\_ct$ de C^2. Dans le formalisme des SED, ces événements sont appelés des événements d'interaction (voir le paragraphe 5.3.2.1, page 111).

La notion de composant dans un SED est la même que celle définie pour notre modélisation générique structurelle d'un système. Chaque paramètre d'un composant a un domaine de définition fini. Un état dans l'automate d'un composant correspond à une affectation de valeur à tous les paramètres de ce composant. L'espace d'états du composant est donc un sous-ensemble du produit cartésien des paramètres du composant (privé, entrée, sortie). Les paramètres de faute sont représentés par des variables booléennes (la faute est présente ou non). Une fonction de base d'un composant est un invariant sur les paramètres du composant. Une transition à un instant k est la réalisation d'un événement e_k : l'état du composant (i.e. tous ses paramètres) $x(k-1)$ passe à l'état $x(k)$. Par exemple, l'occurrence d'une faute f à l'instant k fait que le paramètre privé $pp^{i,f}$ correspondant à la présence de la faute f passe à `vrai` : $pp^{i,f}(t < k) = \texttt{faux} \rightarrow pp^{i,k}(k) = \texttt{vrai}$.

5.3.2.2 Diagnostic et diagnosticabilité dans les SED distribués

La modélisation générique proposée permet de représenter le comportement d'un système distribué par un ensemble d'automates communicants. Il est maintenant nécessaire de présenter les concepts liés au diagnostic de faute et au problème de diagnosticabilité dans les systèmes à événements discrets distribués.

Concepts généraux sur les SED distribués

Le comportement d'un composant G_i est décrit par le langage clos par préfixe $L(G_i) \subseteq \Sigma_i^\star$ qui est généré par l'automate G_i où $\Sigma_i^\star$ représente la clôture de Kleene de l'ensemble Σ_i. Un automate G_i modélise le comportement du composant G_i comme défini dans le paragraphe 5.3.2.1 (page 111).

Un sous-système γ est défini par un ensemble non vide de m composants de G :

$$\gamma = \{G_{j_1}, \ldots, G_{j_m}\}_{j_k \in I}, \tag{5.9}$$

avec $I \subseteq \{1, \ldots, N\}$ et $|I| = m$. On note $\Sigma_\gamma = \bigcup_{j \in I} \Sigma_j$, $\Sigma_{f_\gamma} = \bigcup_{j \in I} \Sigma_{f_j}$, $\Sigma_{c_\gamma} = \bigcup_{j \in I} \Sigma_{c_j}$, $\Sigma_{o_\gamma} = \bigcup_{j \in I} \Sigma_{o_j}$ et $\Sigma_{l_\gamma} = \bigcup_{j \in I} \Sigma_{l_j}$.

Le comportement du sous-système γ peut être explicitement modélisé par $G_{j_1}||\ldots||G_{j_m}$ obtenu à partir de l'opération de synchronisation classique (notée $\|$) sur les événements d'interaction de Σ_{c_γ}. La synchronisation permet d'obtenir l'automate $\|\gamma\| = (X_\gamma, \Sigma_\gamma, T_\gamma, x_{0\gamma})$ qui génère le langage clos par préfixe $L(\gamma)$. Ce langage contient l'ensemble de toutes les séquences d'événements du sous-système γ qui sont possibles à partir de l'état initial $x_{0\gamma} = (x_{0_{j_1}}, \ldots, x_{0_{j_m}})$. Dans le cas où $\gamma = G$, les notations précédentes sont simplifiées : $\Sigma_f = \Sigma_{f_G}$, $\Sigma_c = \Sigma_{c_G}$, $\Sigma_o = \Sigma_{o_G}$, $\Sigma_l = \Sigma_{l_G}$. Le comportement du système G est alors explicitement représenté par le produit $\|G\| = G_1||\ldots||G_n$ qui est appelé le modèle global dans [SAM 95].

Une première hypothèse est que tout sous-système γ est sans blocage, ce qui signifie que le langage $L(\gamma)$ est vivant :

$$\forall s \in L(\gamma), \exists e \in \Sigma_\gamma, s.e \in L(\gamma). \tag{5.10}$$

Les problèmes de blocage ne sont pas considérés dans la théorie classique de diagnosticabilité des SED [SAM 95]. De nouvelles définitions et des résolutions adhoc sont nécessaires pour traiter de tels problèmes. Une seconde hypothèse est que chaque modèle local G_i est globalement cohérent.

Définition 41 (Cohérence globale). *Un modèle G_i est globalement cohérent si*

$$L(G_i) = P_{\Sigma_{G_i}}(L(G_0\|G_1\|\ldots\|G_n)) \tag{5.11}$$

où $P_{\Sigma_{G_i}}$ représente l'opération de projection classique sur Σ_{G_i} dont la définition est donnée ci-dessous.

Un modèle G_i est globalement cohérent si toutes les transitions définies à partir de chaque état du modèle peuvent être franchies dans un comportement global du système G. L'opération de projection peut être définie de la manière suivante. On note ε, la séquence vide dans Σ^*.

Définition 42 (Projection). *L'opération de projection $P_{\Sigma'} : \Sigma^* \to \Sigma'^*$ est définie par $P_{\Sigma'}(\varepsilon) = \varepsilon$ et pour tout $uv \in \Sigma^*$, $u \in \Sigma$,*

$$P_{\Sigma'}(uv) = \begin{cases} uP_{\Sigma'}(v) & si\ u \in \Sigma' \\ P_{\Sigma'}(v) & sinon. \end{cases} \tag{5.12}$$

Cette seconde hypothèse n'est absolument pas restrictive. Pour un modèle donné $G = \{G_1, G_2, \ldots, G_n\}$, il est toujours possible d'obtenir un ensemble équivalent de modèles $G = \{G_1^{glob}, G_2^{glob}, \ldots, G_n^{glob}\}$ qui sont globalement cohérents :

$$G_1\|\ldots\|G_n = G_1^{glob}\|\ldots\|G_n^{glob}. \tag{5.13}$$

Pour obtenir ces modèles, des techniques comme celles définies dans [SU 04] peuvent être utilisées par exemple.

Diagnostic de faute sur un sous-système

Dans le cadre générique, le diagnostic retourne des séquences de modes de faute. Dans le cadre des SED, un mode de faute correspond à un ensemble d'événements de faute qui ont lieu dans le système. Donc le problème de diagnostic de faute dans un SED revient à déterminer les séquences d'événements de faute possibles à partir des observations disponibles sur le système et des modèles à événements discrets des composants. Le problème du diagnostic peut être caractérisé pour tout sous-système γ. Soit un événement de faute F apparaissant sur un composant du sous-système $\gamma = \{G_{j_1}, \ldots, G_{j_m}\}_{j_k \in I}$, un *diagnostiqueur* pour ce sous-système est un processus qui doit diagnostiquer en ligne l'occurrence d'un événement de faute F à partir d'une séquence d'observations $\sigma \in P_{\Sigma_{o_\gamma}}(L(\gamma))$ émises par le sous-système γ. La notation $F \in s$ signifie que la séquence d'événements $s \in L(\gamma)$ contient la faute F. Le but du diagnostiqueur est de mettre en œuvre une fonction de diagnostic $Diag_\gamma^F : P_{\Sigma_{o_\gamma}}(L(\gamma)) \rightarrow \{F\text{-certain}, F\text{-sain}, F\text{-ambigu}\}$ dont la définition est donnée ci-dessous.

Pour toute séquence d'événements observables $\sigma \in P_{\Sigma_{o_\gamma}}(L(\gamma))$,

1. $Diag_\gamma^F(\sigma) = F$-certain si $\forall s \in L(\gamma)$ tel que $P_{\Sigma_{o_\gamma}}(s) = \sigma$, $F \in s$.
2. $Diag_\gamma^F(\sigma) = F$-sain si $\forall s \in L(\gamma)$ tel que $P_{\Sigma_{o_\gamma}}(s) = \sigma$, $F \notin s$.
3. $Diag_\gamma^F(\sigma) = F$-ambigu si $\exists s, s' \in L(\gamma)$ tel que $P_{\Sigma_{o_\gamma}}(s) = \sigma$ et $P_{\Sigma_{o_\gamma}}(s') = \sigma$, $F \in s$ mais $F \notin s'$.

Le diagnostiqueur est généralement défini à partir du modèle global G comme dans [SAM 95]. Le résultat de diagnostic global du système G pour une séquence d'observations σ est alors donné par $\{Diag_G^F(\sigma), F \in \Sigma_f\}$.

Pour se conformer à [SAM 95], le diagnostiqueur d'une faute F est représenté par un automate déterministe. Cette machine est construite à partir d'une projection de l'automate $\|\gamma\| = (X_\gamma, \Sigma_\gamma, T_\gamma, x_{0_\gamma})$ sur les événements observables de Σ_{o_γ}. Le diagnostiqueur est par conséquent un automate qui génère le comportement observable de γ (*i.e.* $P_{\Sigma_{o_\gamma}}(L(\gamma))$) et dont la définition formelle est donnée ci-dessous.

Définition 43 (F-diagnostiqueur pour un sous-système γ). *Soit un sous-système γ et une faute F de γ, le F-diagnostiqueur est l'automate déterministe*

$$\Delta_\gamma^F = (X_{d\gamma}, \Sigma_{d\gamma}, T_{d\gamma}, x_{d\gamma_0})$$

dans lequel

- $X_{d\gamma} \subseteq 2^{X_\gamma \times \{\{F\}, \emptyset\}}$ *est un ensemble fini d'états ;*
- $\Sigma_{d\gamma} = \Sigma_{o_\gamma}$ *est un ensemble des événements ;*
- $T_{d\gamma} \subseteq X_{d\gamma} \times \Sigma_{d\gamma} \times X_{d\gamma}$ *est l'ensemble des transitions ;*
- $x_{d\gamma_0} = (x_{0\gamma}, \emptyset)$ *est l'état initial.*

Les transitions de $T_{d\gamma}$ correspondent aux transitions $x_{d\gamma} \xrightarrow{e} x'_{d\gamma}$ qui sont accessibles depuis l'état initial $x_{d\gamma_0} = \{(x_{0\gamma}, \emptyset)\}$, avec $x_{d\gamma} = \{(x_1, f_1), \ldots, (x_n, f_n)\}$ et $x'_{d\gamma} = \{(x'_1, f'_1), \ldots (x'_m, f'_m)\}$ tel que pour chaque $(x_i, f_i) \in x_{d\gamma}$, il existe une séquence de

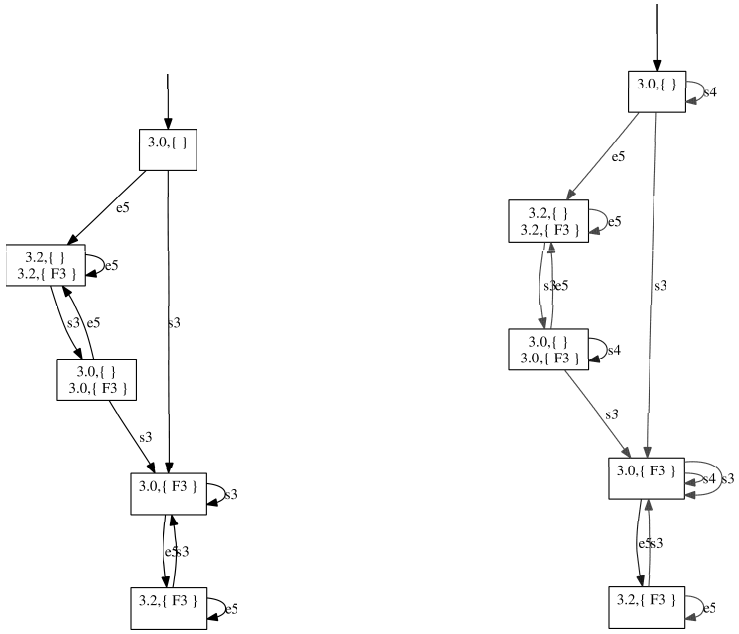

FIG. 5.6 – Exemple SED - Δ_γ^{F3} et Δ_γ^{F3-int} du sous-système $\gamma = \{G_3\}$ pour $\Sigma_{o\gamma} = \{e5, s3\}$

transitions $x_i \xrightarrow{uo_1} x_1 \dots x_{p-1} \xrightarrow{uo_p} x_p \xrightarrow{e} x'_j$ *dans l'automate* $\|\gamma\|$ *avec* $uo_k \in \Sigma_{uo_\gamma}$, $\forall k \in \{1, \dots, p\}$, $\forall j \in \{1, \dots, m\}$ *et* $f' = f \cup (\{F\} \cap \{uo_1, \dots, uo_p\})$.

Le diagnostiqueur Δ_γ^F met en œuvre la fonction $Diag_\gamma^F$. Soit $x_{d\gamma}(\sigma)$, l'état accessible par le diagnostiqueur quelle que soit la séquence d'événements observables σ de γ à partir de l'état initial $x_{d\gamma_0}$. Par construction du diagnostiqueur Δ_γ^F, les équivalences suivantes sont satisfaites :

1. $Diag_\gamma^F(\sigma) = F$-certain ssi $\forall (x_i, f_i) \in x_{d\gamma}(\sigma)$, $f_i = \{F\}$;
2. $Diag_\gamma^F(\sigma) = F$-sain ssi $\forall (x_i, f_i) \in x_{d\gamma}(\sigma)$, $f_i = \emptyset$;
3. $Diag_\gamma^F(\sigma) = F$-ambigu ssi $\exists (x_i, f_i), (x_j, f_j) \in x_{d\gamma}(\sigma)$ tel que $f_i = \{F\}$ et $f_j = \emptyset$.

Le diagnostiqueur $\Delta_{G_3}^{F3}$ du composant G_3 pour la faute $F3$ est représenté sur la figure 5.6.

Diagnosticabilité

La diagnosticabilité est une propriété qui mesure la capacité du système de surveillance à diagnostiquer les fautes qui apparaissent sur le système (voir le paragraphe 5.2.1, page 102). La définition formelle de la diagnosticabilité dans les SED introduite par [SAM 95] est reformulée ci-dessous.

DÉFINITION 44 (Diagnosticabilité globale). *Un événement de faute F est globalement diagnosticable dans le système G ssi*

$$\exists l \in \mathbb{N}, \forall w \in L(G) \text{ tel que } w = uFv,\ |v| \geq l \Longrightarrow Diag_G^F(P_{\Sigma_o}(w)) = F\text{-certain}. \quad (5.14)$$

Il existe plusieurs algorithmes qui permettant de vérifier la diagnosticabilité globale d'un système, [SAM 95], [JIA 01], [YOO 02].

La diagnosticabilité globale, qui est également appelée diagnosticabilité monolithique dans [CON 06], implique l'utilisation d'une architecture de diagnostic global centralisée qui surveille la totalité du système. En pratique, une telle architecture ne peut pas toujours être mise en œuvre surtout pour des systèmes de grande taille. De plus, vérifier la diagnosticabilité globale du système nécessite le calcul du modèle global $\|G\|$ qui est très complexe et pas toujours réalisable dans le cas des systèmes distribués. Afin d'enlever cette limitation, il est préférable de raisonner localement aux sous-systèmes comme dans [PEN 04; CON 06]. On souhaite déterminer les sous-systèmes qu'il suffit d'observer pour diagnostiquer un événement de faute anticipée F apparaissant sur un composant G_i. La définition de la diagnosticabilité locale est fondée sur le résultat de diagnostic pour tout sous-système γ (voir le paragraphe 5.3.2.2).

DÉFINITION 45 (Diagnosticabilité locale). *Un événement de faute F est localement diagnosticable dans un sous-système γ ssi*

$$\exists l \in \mathbb{N}, \forall w \in L(\gamma) \text{ tel que } w = uFv, |v| \geq l \Rightarrow Diag_\gamma^F(P_{\Sigma_{o_\gamma}}(w)) = F\text{-certain}. \quad (5.15)$$

Dans [PEN 04], on prouve que la diagnosticabilité locale sur un sous-système γ correspond à la diagnosticabilité locale sur le système global G. Néanmoins pour que cette propriété soit satisfaite, l'observabilité du système global doit être vivante (*i.e.* $P_{\Sigma_o}(L(G))$ est vivant) comme expliqué dans [SAM 95]. En pratique, pour diagnostiquer une faute F dans un sous-système γ, il faut garantir localement cette condition en prouvant l'observabilité équitable du système. L'observabilité d'un système est *globalement équitable* lorsque l'observabilité de chaque composant est équitable.

DÉFINITION 46 (Observabilité équitable). *L'observabilité d'un système est équitable si et seulement si tous les composants observables du système émettent toujours des observations après un nombre fini d'événements.*

Puisque tous les composants observables du système sont vivants, l'observabilité équitable du système est vérifiée s'il n'existe aucun cycle d'événements non observables dans les composants. Par conséquent, pour un sous-système γ, chaque séquence d'événement de $L(\gamma)$ doit toujours être suivie d'une séquence finie d'événements se terminant par un événement observable de γ :

$$\forall w \in L(\gamma),\ \exists p \in \mathbb{N} \text{ tel que } \forall ww' \in L(\gamma), |w'| = p, \\ \exists \sigma_\gamma \in \Sigma_{o_\gamma}^* \backslash \{\varepsilon\}, P_{\Sigma_{o_\gamma}}(ww') = P_{\Sigma_{o_\gamma}}(w).\sigma_\gamma. \quad (5.16)$$

La propriété suivante présente une relation entre la diagnosticabilité locale et la diagnosticabilité globale. La preuve de cette propriété est développée dans [PEN 04].

PROPRIÉTÉ 1. *Sous l'hypothèse d'observabilité équitable du système, si une faute F est localement diagnosticable dans un sous-système γ alors F est globalement diagnosticable dans le système G.*

D'après cette propriété, il n'est donc pas nécessaire d'observer le système global pour diagnostiquer une faute F apparaissant sur un composant de G. Il peut être suffisant d'observer uniquement un sous-système γ.

5.3.2.3 Méthodologie pour la conception d'un SED diagnosticable

Cette section introduit une méthodologie qui fournit des recommandations et des coûts minimaux aux concepteurs d'un SED. Afin d'éviter une recherche combinatoire exhaustive, la méthodologie est guidée par trois propriétés : la nature distribuée du sous-système, la propriété de précision du diagnostic et la monotonie des propriétés liées au placement de capteurs. Dans un premier temps, on montre comment ces propriétés peuvent améliorer la performance de la méthodologie. On considère dans cette étude qu'il existe déjà une spécification du SED et que la seule opération possible consiste à augmenter l'observabilité des composants du système.

Précision d'un diagnostic

Le diagnostiqueur d'un sous-système γ est précis pour une faute F s'il est toujours capable, à partir d'observations de γ de fournir un diagnostic pour F qui est cohérent avec l'ensemble complet des observations de G (voir le paragraphe 5.2.3.1, page 106). Indépendamment de la diagnosticabilité du système, il est très intéressant de trouver un sous-système dont le diagnostiqueur est précis car c'est un moyen de borner le coût C_S (voir le paragraphe 5.3.1.2, page 106), la surveillance pouvant être limitée à ce sous-système. La fonction de surveillance n'a pas besoin d'informations des autres composants du système. La précision d'un diagnostiqueur est définie de la manière suivante.

DÉFINITION 47. *Le diagnostiqueur d'un sous-système γ est précis pour un événement de faute $F \in \Sigma_{f_\gamma}$ ssi $\forall \sigma.o \in \Sigma_o^\star, o \in \Sigma_{o_\gamma},\ Diag_G^F(\sigma.o) = Diag_\gamma^F(\sigma_\gamma.o)$, où $\sigma_\gamma = P_{\Sigma_{o_\gamma}}(\sigma)$.*

Il est toujours possible de trouver un sous-système dont le diagnostiqueur est précis. Le diagnostiqueur du système global G est forcément précis.

Vérifier la précision d'un diagnostiqueur La précision est une propriété qui compare la performance d'un diagnostiqueur local Δ_γ^F par rapport au diagnostiqueur global Δ_G^F. Une manière simple de vérifier cette propriété est de calculer le diagnosti-

queur global, ce qui est en pratique irréalisable à cause de l'explosion combinatoire. Il existe cependant un critère suffisant qui permet de vérifier si un diagnostiqueur est précis tout en évitant le calcul du diagnostiqueur global. Ce critère repose sur une nouvelle machine à états finis (MEF) que l'on appelle le *diagnostiqueur interactif* (voir la figure 5.6). Un diagnostiqueur interactif Δ_γ^{F-int} est une extension du diagnostiqueur Δ_γ^F défini dans le paragraphe 5.3.2.2 dans lequel une transition est étiquetée soit par un événement observable $e \in \Sigma_{o\gamma}$ soit par un événement d'interaction $e \in \Sigma_\gamma^{int} = \Sigma_{c_\gamma} \cap \Sigma_{c_{G\setminus\gamma}}$.

Le diagnostiqueur interactif est alors défini par le quadruplet

$$\Delta_\gamma^{F-int} = (X_{di\gamma}, \Sigma_{di\gamma}, T_{di\gamma}, x_{di\gamma_0}), \quad \text{où} \quad \Sigma_{di_\gamma} = \Sigma_{o_\gamma} \cup \Sigma_\gamma^{int}. \tag{5.17}$$

Par extension, on note $x_{di_\gamma}(\sigma_\gamma^{int}) = \{(x_1, f_1), \ldots, (x_m, f_m)\}$ l'état atteint par Δ_γ^{F-int} pour tout $\sigma_\gamma^{int} \in P_{\Sigma_{o_\gamma} \cup \Sigma_\gamma^{int}}(L(\gamma))$. Le diagnostic associé à l'état $x_{di_\gamma}(\sigma_\gamma^{int})$ dépend de l'ensemble f_i. Le diagnostic est F-certain si tous les f_i contiennent la faute F, F-sain si aucun d'eux ne contient F, et sinon il est F-ambigu.

Propriété 2. *Le diagnostiqueur Δ_γ^F est précis si le critère suivant est vérifié sur Δ_γ^{F-int} : $\forall \sigma_\gamma \in P_{\Sigma_{o_\gamma}}(L(\gamma))$, $\forall \sigma_\gamma^{int}, \sigma_\gamma^{int'} \in P_{\Sigma_{o_\gamma} \cup \Sigma_\gamma^{int}}(L(\gamma))$ telle que $P_{\Sigma_{o_\gamma}}(\sigma_\gamma^{int}) = P_{\Sigma_{o_\gamma}}(\sigma_\gamma^{int'}) = \sigma_\gamma$, $x_{di_\gamma}(\sigma_\gamma^{int})$ et $x_{di_\gamma}(\sigma_\gamma^{int'})$ sont associés au même résultat de diagnostic.*

La figure 5.6 représente le diagnostiqueur interactif Δ_γ^{F3-int} et un $F3$-diagnostiqueur précis défini sur le sous-système $\gamma = \{G_3\}$ pour $\Sigma_{o_\gamma} = \{e5, s3\}$. Toutes les séquences d'événements dans Δ_γ^{F3-int} qui sont projetées sur les événements observables de Σ_{o_γ} amènent à des états qui contiennent la même information de diagnostic. Par exemple, tous les chemins à partir de l'état initial qui émettent la séquence observable $s3e5s3^*$ sont $F3$-certains et tous les chemins qui émettent la séquence $e5s3e5^*$ sont $F3$-ambigus. Des observations provenant des autres composants du système pourraient apporter plus d'informations sur l'occurrence de l'événement $s4$ qui est non observable mais le diagnostic fourni par Δ_γ^{F3-int} après l'occurrence de $e5$ ne dépend pas de l'occurrence de $s4$. Dans ce cas, les informations qui proviennent des autres composants ne peuvent pas servir à désambiguïser le diagnostic.

Rendre le diagnostic d'un sous-système précis D'après la propriété 2, si toutes les interactions de γ avec les autres composants du système sont observables alors le critère précédent est satisfait. Une manière simple de rendre le diagnostiqueur d'un sous-système précis est d'observer tous les événements d'interaction [RIB 07] (ou événements communs dans [CON 06]).

Propriété 3. *Soit un sous-système γ, le diagnostiqueur Δ_γ^F est précis pour toutes les événements de faute $F \in \Sigma_{f_\gamma}$ si $\Sigma_\gamma^{int} \subseteq \Sigma_{o_\gamma}$.*

Preuve : Soit F, une faute apparaissant dans le sous-système γ (i.e. $F \in \Sigma_{f_\gamma}$). Pour montrer la précision du diagnostiqueur Δ_γ^F, on suppose dans un premier temps

qu'il existe une séquence globale observable σ pour laquelle le diagnostic global est $\Delta(F,\sigma) = F$-certain et le diagnostic local est $\Delta_\gamma(F,\sigma_\gamma) = F$-ambigu. Par conséquent chaque séquence seq de $\|G\|$ qui explique σ (i.e. $P_{\Sigma_o}(seq) = \sigma$) contient la faute F et il existe au moins une séquence locale seq' dans $\|\gamma\|$ qui explique σ_γ (i.e. $P_{\Sigma_{o_\gamma}}(seq') = \sigma_\gamma$) et qui ne contient pas F. Comme $\sigma_\gamma = P_{\Sigma_{o_\gamma}}(\sigma)$ et $\Sigma_\gamma^{int} \subseteq \Sigma_o$, cette séquence locale seq' est nécessairement globalement cohérente, et par conséquent, elle fait partie d'au moins une séquence globale qui explique σ, d'où une contradiction. Supposons maintenant l'existence d'une séquence σ pour laquelle $\Delta(F,\sigma) = F$-sain et $\Delta_\gamma(F,\sigma_\gamma) = F$-ambigu. En utilisant le même type de raisonnement, on arrive également à une contradiction, d'où le résultat. □

Monotonie des propriétés

L'objectif est maintenant de définir une méthodologie pour déterminer un ensemble de modifications qui garantissent les propriétés d'observabilité équitable et de diagnosticabilité tout en considérant la précision de l'architecture de diagnostic. Il est donc important de savoir si chacune des propriétés (observabilité équitable, diagnosticabilité, précision) peut être conservée après les modifications pour garantir les autres.

Définition 48 (Monotonie). *Soit Prop, une application booléenne ($\forall \mathbb{X}$, $Prop : P(\mathbb{X}) \mapsto \{0,1\}$, où $P(\mathbb{X})$ représente l'ensemble de parties de $\mathbb{X}$), Prop est monotone ssi*

$$\forall \Sigma_x,\ \Sigma_y,\ \Sigma_x \subseteq \Sigma_y,\ Prop(\Sigma_x) \Rightarrow Prop(\Sigma_y). \tag{5.18}$$

Cette définition montre que pour deux ensembles d'événements Σ_x et Σ_y, si Σ_x est inclus dans Σ_y, toute propriété vérifiée par Σ_x est aussi vérifiée par Σ_y. Cela vient du fait que le langage L_{Σ_x} généré à partir de l'ensemble d'événement Σ_x est forcément plus restreint que le langage L_{Σ_y} généré à partir de l'ensemble d'événements L_{Σ_x} : $L_{\Sigma_y} \subseteq L_{\Sigma_x}$. Dans notre cas, Σ_x et Σ_y sont deux ensembles d'événements observables ($\Sigma_x \subseteq \Sigma$, $\Sigma_y \subseteq \Sigma$). La monotonie de propriété telle que l'observabilité ou la diagnosticabilité a été étudiée dans [JIA 03]. L'observabilité équitable (définition 46) est de toute évidence une propriété monotone. Si tout composant observable du système émet toujours une observation dans un délai fini, l'ajout d'une nouvelle observation conservera cette propriété.

Propriété 4. *Si F est diagnosticable avec un ensemble d'observations Σ_o qui satisfait l'observabilité équitable alors F est diagnosticable avec un ensemble d'observations $\Sigma'_o = \Sigma_o \cup \{o\}$.*

Preuve : Par la négation de la définition 45 (définition de la diagnosticabilité locale), une faute F n'est pas diagnosticable avec un ensemble d'observations Σ'_o s'il existe une séquence infinie d'observations σ telle que le diagnostic $Diag_\gamma^F(\sigma)$ est F-ambigu.

Cela signifie qu'il existe au moins deux séquences infinies d'événements p_1 et p_2, telles que $p1$ contient F mais pas $p2$, et dont la projection sur les observations de Σ'_o est σ. Soit Σ_o, un nouvel ensemble d'observations privé de l'événement o : $\Sigma_o = \Sigma'_o \backslash \{o\}$. Prouvons que si F n'est pas diagnosticable avec Σ'_o, alors F n'est pas non plus diagnosticable avec Σ_o. Si $o \notin \sigma$, il existe encore une séquence infinie σ telle que le diagnostic est F-ambigu. Si $o \in \sigma$, et si la configuration observable Σ_o satisfait l'observabilité équitable du système (voir la définition 46), alors on obtient une nouvelle séquence infinie d'observations privée de o, que l'on note $\sigma_{\backslash o}$, telle que le diagnostic est F-ambigu (car elle résulte de la projection de p_1 et p_2 sur les observations de Σ_o). □

Propriété 5. *La précision n'est pas une propriété monotone.*

Preuve : La figure 5.6 à la page 120 montre un $F3$-diagnostiqueur précis en observant $\{e5, s3\}$ sur G_3. Une nouvelle observation $e6$ est maintenant considérée sur le composant G_3. Soit σ_1 et σ_2, deux séquences globalement observables de Σ_o^* telles ques $\sigma_1 = e6e7e7e5$ et $\sigma_2 = e6e7e3e5$. Après l'observation de σ_1, le diagnostic global est F-sain tandis qu'après l'observation de σ_2, le diagnostic global est F-certain. La projection des deux séquences d'événements σ_1 et σ_2 sur les événements de G_3 est $e6e5$. Les séquences σ_1 et σ_2 ne sont pas distinguables localement, le diagnostic local après l'observation de $e6e5$ est F-ambigu. Le diagnostic local n'est pas équivalent au diagnostic global donc le diagnostic de $F3$ n'est plus précis en considérant l'observation de $e6$ sur G_3. □

La précision n'est pas une propriété monotone mais elle peut être préservée dans le cas où toutes les interactions du sous-système γ avec les autres composants sont observables. Dans ce cas particulier décrit par la propriété 3, la propriété de précision reste satisfaite après l'ajout de nouvelles observations.

La propriété de diagnosticabilité est toujours conservée en considérant de nouvelles observations alors que la propriété de précision peut être perdue par cet ajout d'information (elle est uniquement préservée dans des cas particuliers). La sélection des capteurs est guidée par la monotonie de manière à préserver les propriétés de diagnosticabilité et de précision.

Algorithme

Ce paragraphe présente un algorithme qui sélectionne un sous-système et propose des modifications à réaliser sur celui-ci afin de le rendre diagnosticable à l'aide d'un diagnostiqueur précis pour un coût total C_G minimal. On considère un système de n composants avec une configuration observable minimale Σ_o. Une spécification du système correspond à un ensemble initial de capteurs déjà disponibles sur le système. La configuration observable initiale peut être vide (aucun capteur n'est placé sur le système), ce cas est illustré sur un exemple dans le paragraphe suivant.

L'algorithme 1 sélectionne un sous-système pour lequel le coût total C_G est minimal (en considérant le coût de surveillance C_S, le coût C_P pour la précision et le coût C_D pour la diagnosticabilité de la faute F intervenant dans le composant G_i). Cet algorithme retourne des recommandations *Rec* sous la forme d'un ensemble de modifications à réaliser sur le sous-système. Dans cette étude, seules les modifications de placement de capteurs (*i.e.* opérations de type 1, 2 ou 3 de la section 5.3.1.1) sont considérées. Comme la diagnosticabilité est une propriété monotone, il est préférable de rendre le sous-système diagnosticable avant de considérer la précision. Si le sous-système est diagnosticable, la propriété de diagnosticabilité sera conservée par l'ajout d'événements observables pour rendre le diagnostic précis. Si le sous-système n'est pas diagnosticable, l'optimisation se fera directement sur le couple de coûts (C_P, C_D). Dans l'algorithme proposé, l'expression `Sous_Systeme`$(G_i,\ k)$ représente l'ensemble des sous-systèmes composés de k composants qui contiennent G_i. Il faut dans un premier temps vérifier l'existence d'une solution au problème de placement de capteurs. La fonction `Existence_Solution`(γ, F) repose sur la propriété 6.

PROPRIÉTÉ 6. *Si l'événement de faute F intervenant sur un composant du sous-système γ n'est pas diagnosticable dans γ en considérant que tous les événements sont observables (à l'exception des événements de faute), alors il n'existe pas de solution au problème de placement de capteurs.*

Si tous les événements sont observables et que F n'est pas diagnosticable, aucune information sur les événements des composants qui interagissent avec γ ne pourra rendre F diagnosticable. La seule façon de rendre F diagnosticable est de reconcevoir le composant sur lequel elle apparaît en considérant des opérations de type 4 ou 5 (voir la section 5.3.1.1, page 108).

La fonction `Cout_Surveillance` induit un coût C_S pour la surveillance du sous-système qui dépend de l'implémentation de l'architecture de supervision comme celà est expliqué dans la section 5.3.1.2. Les fonctions `Verifier_Diagnosticabilite` et `Verifier_Precision` sont booléennes. La première fonction applique le critère de la propriété 2 pour déterminer si le diagnostic du sous-système est précis et la seconde fonction utilise un algorithme pour vérifier la diagnosticabilité d'un sous-système pour une faute F [JIA 01][PEN 04][SCH 07a]. Ces algorithmes sont quadratiques en le nombre d'états de $\|\gamma\|$. Les fonctions `Rendre_Diagnosticable` et `Rendre_Precis` utilisent des modifications de type 1., 2. ou 3. de la section 5.3.1.1 et des techniques de placement de capteurs comme dans [JIA 03; BRI 08] pour obtenir un sous-système γ diagnosticable. La fonction `Rendre_Precis` peut aussi reposer sur la propriété 3, c'est une façon simple d'obtenir un sous-système dont le diagnostic est précis. Les configurations d'observations choisies doivent garantir plusieurs propriétés. Elles doivent tout d'abord rendre la faute F diagnosticable avec un diagnostiqueur précis mais elles doivent également garantir que ces propriétés déjà obtenues pour une autre faute F' ne sont pas perdues, d'où l'importance de la propriété de monotonie.

Algorithme 1 : Recommandations et coûts pour la diagnosticabilité

Entrées : $F \in \Sigma_{f_i}$, $G = \{G_1, \ldots, G_n\}$, Σ_o
Sorties : γ, C_G, Req

1 $C_G^0 \leftarrow \infty$; $k \leftarrow 1$; $Req = \emptyset$
2 **répéter**
3 $C_G^k \leftarrow \infty$
4 **pour chaque** $\gamma \in$ *Sous_Systeme*$(G_i,\ k)$
5 **faire**
6 **si** $\neg$*Existence_Solution*(γ, F) **alors** Aller en 18 ;
7 $C_S \leftarrow$ `Cout_Surveillance`(γ) ; $C_D = 0$; $C_P = 0$;
8 **si** *Verifier_Diagnosticabilite*(γ, F)
9 **alors**
10 **si** $\neg$*Verifier_Precision*(γ) **alors**
 $(Rec, C_P) \leftarrow$ `Rendre_Precis`(γ) ;
11 **sinon** $(Rec, C_P, C_D) \leftarrow$
 `Rendre_Diagnosticable`$(\gamma, F) \wedge$ `Rendre_Precis`(γ) ;
12 $C_G^\gamma \leftarrow C_S + C_P + C_D$;
13 $C_G^k \leftarrow \min(C_G^k, C_G^\gamma)$;
14 $k \leftarrow k + 1$
15 **jusqu'à** $(C_G^{k-1} \geq C_G^{k-2}) \wedge (k \geq 2) \wedge (k \leq n)$;
16 $G_G \leftarrow C_G^{k-2}$
17 Aller en 19 ;
18 **retourner** *Aucune solution*
19 **retourner** γ, G_G, Rec

L'algorithme présenté permet de sélectionner un sous-système pour lequel le coût total C_G est minimal. La solution obtenue n'est pas unique mais l'algorithme peut être étendu afin de fournir un ensemble possible de sous-systèmes qui ont tous un coût minimal.

Exemple illustratif

L'algorithme est maintenant appliqué sur un exemple de spécification pour le système illustré sur la figure 5.2, page 112.

On commence par analyser la diagnosticabilité de la faute $F1$ qui apparaît sur le composant G_1. La configuration observable initiale est supposée vide : $\Sigma_o = \emptyset$. On s'assure qu'il existe bien une solution au problème de placement de capteurs en considérant tous les événements de G_1 comme observables (à l'exception de la faute $F1$). Il faut ensuite déterminer les configurations observables possibles qui rendent $F1$ diagnosticable et $\Delta^F_{G_1}$ précis. La configuration minimale retenue est $\Sigma_{o_1} = \{e1, s2, s3\}$. Le

coût global $C_G^{G_1}$ englobe le coût d'observation de ces événements ($C_D + C_P$), et le coût de surveillance C_S du composant G_1. Tous les sous-systèmes de deux composants qui interagissent et contiennent G_1 sont ensuite étudiés : $\|\gamma_1\| = G_1\|G_2$ et $\|\gamma_2\| = G_1\|G_3$. Dans γ_1, la faute $F1$ est diagnosticable avec un diagnostiqueur précis pour la même configuration observable : $\Sigma_{o_1} = \{e1, s2, s3\}$. Par contre le coût de surveillance de deux composants est forcément plus élevé que pour un seul composant. Le coût global $C_G^{\gamma_1}$ pour ce sous-système est donc plus élevé que $C_G^{G_1}$. En suivant le même raisonnement pour le sous-système γ_2, on trouve que le coût global $C_G^{\gamma_2}$ est aussi plus élevé que $C_G^{G_1}$. Il est donc préférable de considérer uniquement G_1 avec la configuration observable $\Sigma_{o_1} = \{e1, s2, s3\}$ et un coût $C_G^{G_1}$. Dans ce cas, $Rec = \{e1, s2, s3\}$ car la configuration observable était initialement vide. On aurait pu obtenir une configuration observable qui garantit les objectifs avec un coût moins élevé en considérant le sous-système γ_1 (ou γ_2) seulement si le nouveau coût ($C_D + C_P$) du sous-système était inférieur à la différence entre le coût de surveillance du composant G_1 et le coût de surveillance du sous-système γ_1 (ou γ_2).

La diagnosticabilité de la faute $F2$ qui survient dans le composant G_2 est ensuite analysée. La configuration observable initiale n'est plus vide : $\Sigma_o = \{e1, s2, s3\}$. L'existence d'une solution au problème de placement de capteurs est étudiée en considérant tous les événements de G_2 comme étant observables (à l'exception de la faute $F2$). La faute $F2$ ne peut pas être diagnosticable dans G_2 en utilisant ces techniques. La seule solution est de modifier la structure du composant G_2.

La configuration observable initiale est toujours $\Sigma_o = \{e1, s2, s3\}$ pour étudier la diagnosticabilité de $F3$ dans G_3 car l'analyse de $F2$ n'a apporté aucune observation supplémentaire (puisqu'il n'existe pas de solution pour diagnostiquer $F2$). Il existe bien une solution au problème de placement de capteurs dans G_3, il est donc possible de trouver des configurations observables qui rendent $F3$ diagnosticable avec un diagnostiqueur précis : $\{e5, e6, s4\} \in \Sigma_o$. Le coût global $C_G^{G_3}$ englobe le coût des observations et le coût de surveillance du composant G_3. Les sous-systèmes de deux composants qui interagissent et qui contiennent G_3 sont ensuite considérés : $\|\gamma_1\| = G_3\|G_1$ et $\|\gamma_2\| = G_3\|G_2$. La faute $F3$ est diagnosticable dans les sous-systèmes γ_1 et γ_2 avec la même configuration observable que celle qui avait été obtenue pour le composant G_3. Le coût de surveillance de deux composants étant plus élevé que pour un seul composant, le coût global associé à ces sous-systèmes est forcément plus élevé que le précédent. La faute $F3$ est diagnosticable dans G_3 avec $Rec = \{e5, e6, s4\}$ et un coût global $C_G^{G_3}$.

Le système distribué est finalement diagnosticable pour les fautes $F1$ et $F3$ avec la configuration observable $\Sigma_o = \{e1, e5, e6, s2, s3, s4\}$ et une architecture de diagnostic qui comprend deux diagnostiqueurs : un sur le composant G_1 qui permet de diagnostiquer $F1$ et un sur le composant G_3 pour diagnostiquer $F3$. Même si $F2$ n'est pas diagnosticable dans le système, son diagnostic est précis avec la configuration observable obtenue (car tous les événements d'interaction de G_2 sont observables), il serait possible de déployer un autre diagnostiqueur sur le composant G_2.

5.4 Conclusion

Ce chapitre définit les propriétés de diagnosticabilité, de pronosticabilité et de précision à partir desquelles il est possible d'évaluer la performance d'une l'architecture de supervision.

Afin de garantir la diagnosticabilité d'un système distribué, il est possible d'établir des recommandations de conception pour le système sous la forme de modifications dans les spécifications des composants. Ce problème de retour sur conception est défini comme un problème d'optimisation de coût en tenant compte du coût de conception du système mais aussi des coûts liés à l'architecture de surveillance afin de minimiser les coûts d'intégration du système distribué.

Une méthodologie est développée dans le cadre des SED distribués pour aider à sélectionner les modifications à réaliser dans la spécification du système et à choisir l'architecture de diagnostic la plus appropriée à déployer sur le système. Cette méthodologie est guidée par la propriété de précision d'un diagnostic qui un un moyen d'isoler un sous-système pour lequel le diagnostic est aussi précis que possible.

6 Application à la maintenance d'un système aéronautique

Résumé : Ce chapitre présente un exemple d'application développé dans le cadre du projet ARCHISTIC en collaboration avec Airbus et l'ENIT dont le principal objectif est de concevoir une architecture embarquée de surveillance pour les avions du futur. Cette architecture doit permettre d'optimiser au sol l'efficacité des services de maintenance. Les différentes entités impliquées dans la construction d'un diagnostic avion sont décrites ainsi que leurs interactions. L'exemple spécifique du système de génération d'air sous pression à partir des moteurs permettant entre autres de climatiser la cabine de l'avion nous sert d'illustration. Ce système est appelé en jargon aéronautique EBAS (Engine Bleed Air System), nous traduirons cet acronyme par "système de génération pneumatique" dans ce manuscrit. Le formalisme introduit dans cette thèse permet de modéliser le système et de caractériser sur ce modèle la fonction de diagnostic. Enfin, nous montrons l'intérêt d'une fonction de pronostic pour de tels systèmes industriels complexes.

6.1 Présentation du projet Archistic

Le projet ARCHISTIC est un projet en collaboration avec AIRBUS et l'Ecole Nationale d'Ingénieur de Tarbes (ENIT). L'objectif est de caractériser une nouvelle génération d'architecture de surveillance à bord d'un avion civil. Cette architecture de surveillance a pour but d'optimiser les actions de maintenance qui sont réalisées au sol afin d'améliorer la disponibilité de l'avion. Elle devra évidemment respecter les exigences requises par Airbus et ses sociétés de sous-traitance.

L'architecture doit contenir un module de diagnostic en ligne dont la tâche est d'analyser en vol les différentes sources d'observations et de fournir un diagnostic de panne sur les équipements à remplacer. L'objectif du diagnostic est double : il doit non seulement fournir un résultat le moins ambigu possible pour la maintenance, c'est-à-dire désigner uniquement les équipements à remplacer, mais aussi expliquer l'ensemble des alarmes qui ont été émises par l'avion. La fonction de diagnostic est caractérisée en logique du premier ordre (issue de la théorie logique du diagnostic à base de modèle, [REI 87]) afin de définir un diagnostic correct et complet en fonction des connaissances disponibles et du cahier des charges.

Des critères de performances sur l'architecture de surveillance ainsi caractérisée ont ensuite été définis. Les moyens d'améliorer la diagnosticabilité d'un système aéronautique ont en particulier été étudiés en prenant en compte non seulement la possibilité de mettre en place des capteurs sur des équipements (recommandations retournées aux équipementiers) mais également l'architecture logicielle distribuée de l'algorithme de surveillance.

Un autre objectif du projet ARCHISTIC est de mettre en place un module de pronostic couplé avec le module de diagnostic. L'objectif du module de pronostic est de pouvoir établir en fonction du passé la durée de vie résiduelle des équipements (RUL) afin de prévoir la disponibilité opérationnelle à un instant futur et d'organiser une maintenance prédictive. Une difficulté majeure réside dans l'hétérogénéité des équipements et dans leur intégration. Une fonction de pronostic générique a été définie et propose une représentation du pronostic identique pour chaque équipement. À partir de cette représentation, il est ainsi possible de caractériser le pronostic sur la durée de vie d'une fonction du système par une combinaison des pronostics sur les équipements afin de fournir une information pertinente et synthétique pour la décision de maintenance prédictive.

6.2 Diagnostic de faute dans un système aéronautique

L'équipement avionique est constitué d'unités remplaçables que l'on appelle des LRU (*Line Replaceable Unit*). Un LRU est donc un composant matériel ou logiciel qui peut être déposé et remplacé à une escale par un opérateur de maintenance.

L'objectif du diagnostic est de faciliter les opérations de maintenance sur l'avion après l'occurrence d'une ou plusieurs fautes. Il s'agit donc dans un premier temps d'isoler et d'identifier les LRU en faute. Il faut ensuite relier ces LRU en faute avec les effets qu'il peuvent avoir sur le comportement du système global. Cette section présente les différentes entités impliquées dans la construction d'un diagnostic de faute pour un système aéronautique.

6.2.1 Agents de surveillance

L'architecture modulaire (IMA, *Integrated Modular Avionics*) dans les avions permet à un module d'héberger plusieurs applications (fonctions) qui ne sont pas forcément mises en œuvre par le même système ou LRU. La plateforme IMA assure le partage des ressources des différents LRU ou sous-systèmes (ensemble de LRU). Cette architecture est constituée, entre autres, de CPIOM (*Core Processing Input/Output Module*) qui sont des modules capables d'exécuter des applications indépendantes et de gérer les entrées-sorties spécifiques de ces applications. Un CPIOM est donc un calculateur qui peut héberger plusieurs applications d'autres LRU du système. Chaque CPIOM pos-

sède un agent de surveillance et de diagnostic que l'on appelle un agent HMon (*Health Monitoring*).

Un HMon envoie pendant le vol des messages en temps réel concernant le statut fonctionnel des applications des LRU qu'il héberge et un message concernant le statut fonctionnel propre au CPIOM à un système de maintenance centralisé que l'on appelle le CMS (*Centralized Maintenance System*). Ces messages de statut fonctionnel (ou messages de faute) provenant des différents HMon du système correspondent à des diagnostics locaux réalisés au niveau d'un ensemble de LRU.

6.2.2 Prise en compte de la chaîne de sécurité

Lorsqu'une défaillance dans le système est détectée par des agents HMon, une alarme se déclenche au niveau du FWS (*Flight Warning System*). Les alarmes affichées par le FWS alertent le pilote de l'occurrence d'une condition de fonctionnement anormale de l'avion. Lorsqu'une alarme est émise, un RFDCE (*Reported Flight Deck and Cockpit Effet*) est automatiquement produit. Ce RFDCE est une information standardisée associée à l'alarme qui vient de se déclencher. Il est envoyé au CMS et utilisé pour calculer le diagnostic (voir la figure 6.1). Le pilote et l'équipage rapportent tous les symptômes qu'ils observent (entre autres les alarmes affichées par le FWS) dans le logbook. Le logbook contient alors toutes les plaintes du pilote.

Le diagnostic doit prendre en compte la chaîne de sécurité de l'avion en expliquant toutes les plaintes du pilote rapportées dans le logbook et en indiquant à l'opérateur de maintenance les actions à réaliser pour "éteindre" les alarmes du FWS.

6.2.3 Diagnostic pour la maintenance d'un avion

Lorsqu'une défaillance est détectée par un HMon, une ou plusieurs alarmes du FWS sont déclenchées et les RFDCEs correspondants sont envoyés au CMS. Ces alarmes sont donc considérées au niveau du CMS comme des symptômes qui représentent l'effet visible d'une condition anormale du système. Le diagnostic doit expliquer les symptômes observés en identifiant les éléments du système (LRU) qui sont en faute. Il doit également expliquer les plaintes du pilote rassemblées dans le logbook et permettre une corrélation des alarmes des RFDCE avec les diagnostics calculés pour l'avion au niveau des différents HMon.

Un message de faute envoyé par un HMon est associé à la perte d'une application (une fonction). Un message contient un ensemble de données expliquant un ou plusieurs symptômes (une ou plusieurs défaillances détectées) avec des assertions logiques sur l'état de santé d'objets accusés dans le système. Pour chaque défaillance, un seul message est envoyé au CMS.

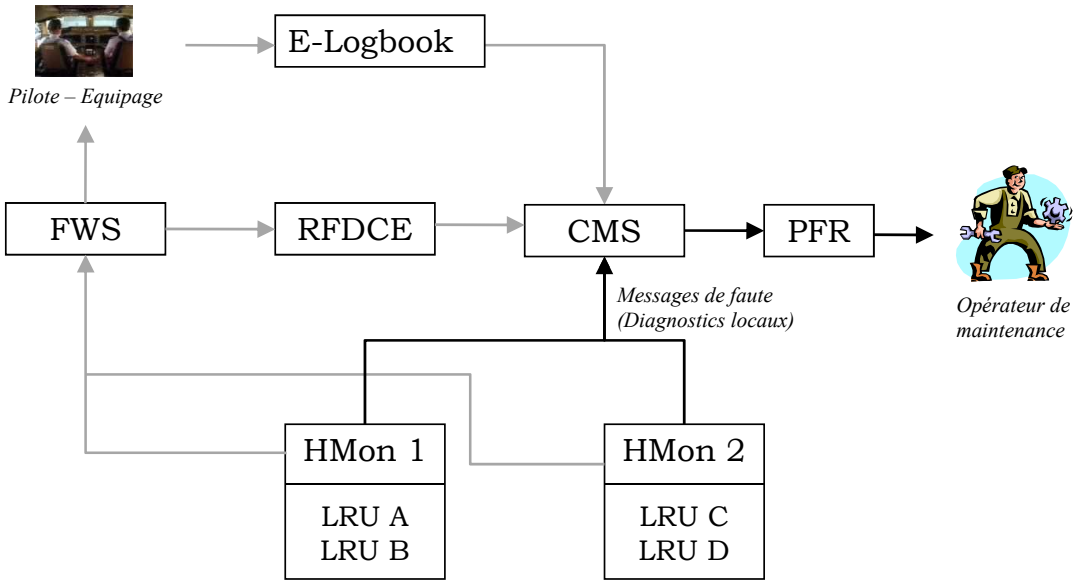

FIG. 6.1 – Diagnostic pour la maintenance d'un avion

Ces assertions logiques contenues dans un message de faute proviennent généralement d'une AMDE. Une AMDE se présente sous la forme d'une table donnant les relations entre les fautes/défaillances et les moyens de détection dans le système. Les HMon utilisent une AMDE étendue pour établir leur diagnostic, c'est-à-dire associer une liste d'objets accusés à une défaillance détectée. Un objet accusé peut être un candidat de faute, une condition opérationnelle, un RIM (*Regular Inoperative Mode*) ou une PFC (*Predefined Fonctional Condition*).

- Un candidat de faute est une accusation d'un unique élément structurel (logiciel, matériel) qui peut être remplacé.
- Une condition opérationnelle correspond à une accusation d'un mode de fonctionnement spécifique d'un LRU (ou de son HMon) qui peut être observée.
- Un RIM est un mode inopérant générique correspondant à un candidat de faute (par exemple un reset).
- Une PFC désigne la condition d'une défaillance dans une interface fonctionnelle avec un système externe. Certaines conditions de faute sont prédéfinies dans la base de données du CMS, de telle sorte que les agents HMon peuvent s'y référer pour envoyer leur message de faute sans avoir à transmettre tout leur contenu.

Le CMS construit ensuite un diagnostic avion qui s'appuie sur les diagnostics locaux fournis par les différents agents HMon, qui seuls, n'ont pas forcément la connaissance suffisante pour isoler les LRU qui sont en faute. Le CMS utilise les messages de faute envoyés par les HMon pour expliquer également chaque RFDCE reçu en terme de liste

d'objets accusés. Il corrèle enfin ces explications avec les plaintes du logbook afin de leur associer une TSP (*Troubleshooting Procedure*) permettant à l'opérateur de maintenance de réparer l'avion.

Les différents entités impliquées par la fonction de diagnostic pour la maintenance d'un avion sont illustrées sur la figure 6.1.

6.2.4 Pronostic pour la maintenance d'un avion

Le pronostic est un domaine émergent qui suscite de plus en plus d'intérêt pour l'aéronautique. Il s'agit d'un moyen d'augmenter la sécurité et la disponibilité des avions tout en réduisant les coûts associés à la maintenance.

Le problème du pronostic n'est pas encore pris en compte dans la conception des architectures de surveillance des avions. Il n'a forcément pas la maturité du diagnostic qui est étudié depuis de nombreuses années. Le terme de pronostic n'est d'ailleurs pas utilisé dans les applications aéronautiques, on parlera plutôt de méthodes de suivi de tendances (*Trend Monitoring*) qui permettent de prédire l'évolution du système. Des études de fiabilité sont néanmoins réalisées par chaque concepteur pour déterminer le MTBF (*Mean Time Between Failure*) des composants du système. Pour le moment, il n'est pas possible d'obtenir des données concernant l'évolution de l'usure des composants selon des conditions réelles de stress et d'utilisation pour déterminer un pronostic adaptatif.

6.3 Application : système de génération pneumatique

L'objectif est ici de montrer que le formalisme générique présenté dans cette thèse peut être facilement utilisé pour modéliser un cas d'application industrielle réel. Les contraintes temporelles et économiques liées à la plupart des systèmes industriels complexes sont prises en compte pour optimiser la maintenance.

On étudie le système de génération pneumatique qui peut être modélisé structurellement et fonctionnellement par un ensemble de paramètres, de rangs et de relations. La définition des modes opérationnels pour ses composants permet d'illustrer la fonction de diagnostic du système caractérisée dans le chapitre 3.

6.3.1 Description du système étudié

Un système avion se décompose en chapitres que l'on appelle des ATA (*Air Transport Avionic*). Ces ATA correspondent à des normes définies par les compagnies aériennes pour représenter les différents programmes (sous-systèmes) dans un avion. L'ATA 36

correspond au système de génération pneumatique qui permet à l'avion d'être ventilé et régulé en pression et en température.

Le système de génération pneumatique (*Bleed System*) prélève l'air chaud sous pression provenant des quatre moteurs de l'avion et de l'APU (*Auxiliary Power Unit*) pour fournir la totalité de l'air nécessaire à l'avion. Il alimente ainsi le système d'air de service (système hydraulique, système de ventilation), les deux AGU (*Air Generation Unit*) qui fournissent une quantité d'air suffisante pour pressuriser la cabine et maintenir une température ambiante confortable, le système de dégivrage des ailes de l'avion (*Wing anti-ice*) et permet le démarrage des moteurs de l'avion.

Un moteur peut être démarré avec un autre moteur de l'avion ou l'APU. L'APU est une turbine qui est utilisée pour le démarrage des moteurs principaux et permet de fournir l'énergie électrique et notamment de faire fonctionner le système de conditionnement d'air de la cabine lorsque l'avion est au sol. Les moteurs et l'APU sont les sources d'air primaires, l'HPGC (*High Pressure Ground Connector*) est une source d'air externe qui peut également être utilisée pour fournir de l'air sous pression uniquement quand l'avion est au sol.

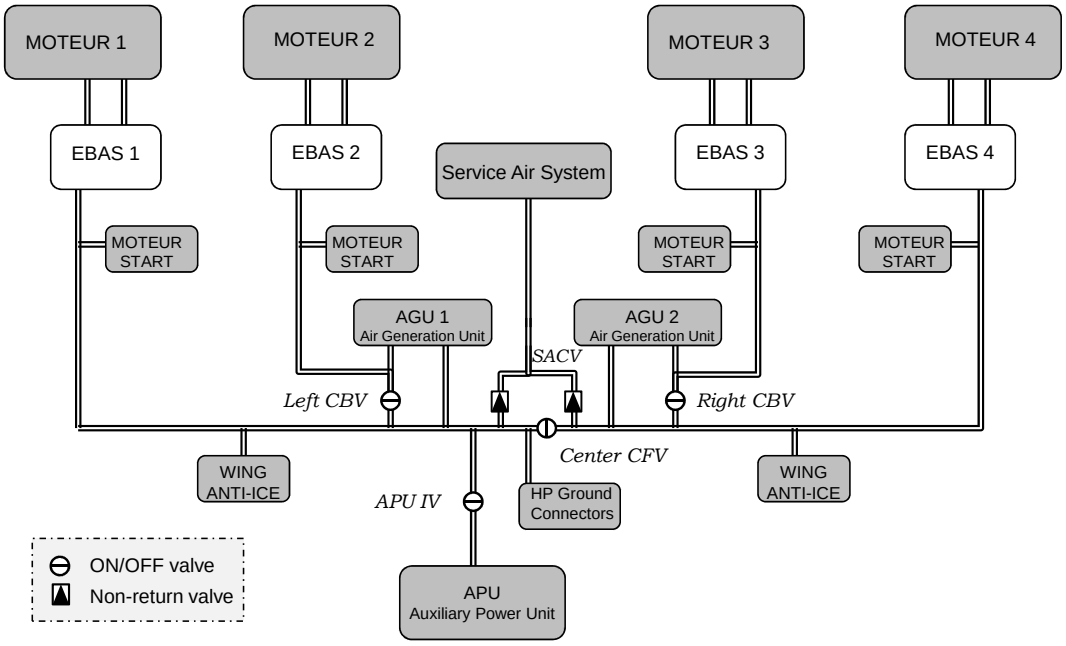

FIG. 6.2 – Système de génération pneumatique

Le système pneumatique, qui est illustré sur la figure 6.2, est principalement composé de six sous-systèmes : un système de distribution pneumatique, un système de détection de fuite d'air et quatre systèmes de prélèvement d'air moteur (un système par moteur).

Le système de distribution pneumatique (ATA 36-12, PADS - *Pneumatic Air Sources*

& Distribution System) est composé de quatre valves permettant l'échange d'air entre les moteurs, et l'APU : deux CBV (*right/left cross bleed valve*), une CFV (*cross-feed valve*), une APU IV (*APU Isolation Valve*). La fonction objectif du système de distribution pneumatique est de fournir l'air pressurisé aux systèmes consommateurs via les canalisations.

Le système de détection de fuites et de surchauffe (ATA 36-22, OHDS - *Overheat Detection System*) est constitué de deux calculateurs (deux OHDU - *Overheat Detection Unit*) dont la tâche est de détecter et localiser les surchauffes dues aux fuites dans les conduits d'air.

Les systèmes de prélèvement d'air moteur (ATA 36-11, EBAS - *Engine Bleed Air System*) sont chacun constitués d'un système de refroidissement (*Pre-Cooler*) et de cinq vannes (IPCV - *Intermediate Pressure Check Valve*, HPV - *High Pressure Valve*, PRV - *Pressure Regulating Valve*, OPV - *Over pressure Valve* et FAV - *Fan Air Valve*). Ces systèmes sont localisés dans l'avion au niveau des quatre moteurs. Les différents composants d'un système EBAS peuvent être repérés sur la figure 6.3.

Quatre capteurs P_{IP} (*Intermediate Pressure transducer*), P_t (*Transferred Pressure transducer*), P_r (*Regulated Pressure transducer*) et ΔP (*Differential Pressure Transducer*) permettent de mesurer la pression de l'air à différents endroits dans le système et un capteur Ts permet de mesurer la température.

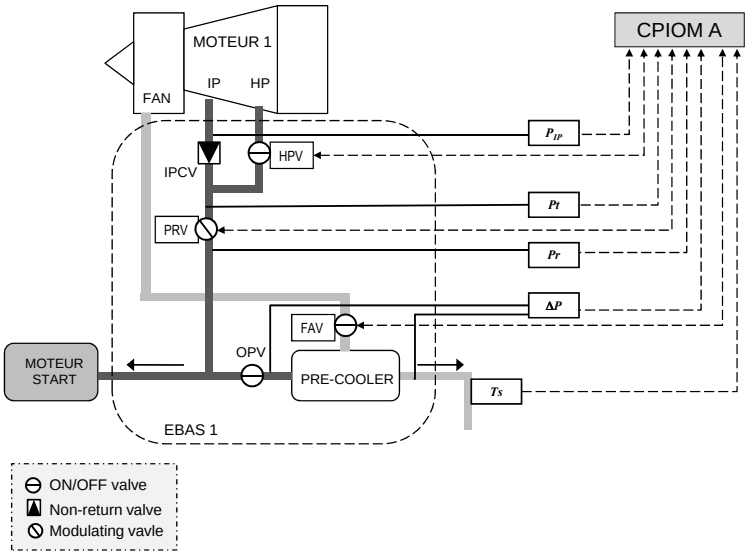

FIG. 6.3 – Système de prélèvement d'air moteur

La fonction objectif du système de prélèvement d'air moteur (EBAS) est de prélever l'air sous pression d'un moteur et de fournir un courant atmosphérique régulé en température et en pression au système de distribution pneumatique qui le transmettra aux différents consommateurs.

Pour assurer cette fonction objectif, l'EBAS doit mettre en œuvre plusieurs fonctions de base qui impliquent un ou plusieurs composants du système. Les fonctions et les commandes des composants de l'EBAS sont intégrées dans l'IMA comme des applications qui sont implémentées sur un module CPIOM A. Les capteurs de pression et de température envoient les valeurs acquises au CPIOM A. Le pilote peut également commander le système depuis la cabine par le panneau AIR au moyen de boutons-poussoirs et de commutateurs. Par ce panneau, il surveille également les alarmes déclenchées par le FWS lorsqu'une défaillance est détectée.

6.3.2 Modélisation de la vanne de régulation de pression

On va plus particulièrement s'intéresser à la vanne de régulation de pression (PRV) du système de prélèvement d'air moteur. La PRV doit ajuster la pression de l'air provenant du moteur entre 30 psig et 40 psig (pression en livres par pouce carré) pour répondre aux besoins des systèmes consommateurs d'air comprimé.

Pour être actionnée, la vanne PRV doit être tout d'abord activée par une tension U_s de $24V$ appliquée à un solénoïde qui joue le rôle d'un interrupteur. Lorsque $U_s = 0V$, la vanne est inactive et fermée quelle que soit la pression en amont de la PRV. Lorsque $U_s = 24V$, la vanne est active et s'ouvre pour une pression en amont supérieure à 20 psig. La pression P_{up} en amont de la PRV est mesurée par le capteur P_t et la pression P_{dw} à la sortie est mesurée par le capteur P_r.

Lorsque la vanne est activée, la pression P_{up} provenant du moteur est tout d'abord réduite grossièrement par un système réducteur de pression. Une régulation plus fine est effectuée par un servomoteur (TM pour *Torque Motor*) commandé par un courant I_{tm}. Ces éléments de réduction et de régulation, illustrés sur la figure 6.4, positionnent le papillon de la PRV permettant d'avoir une pression régulée P_{dw} en sortie de la vanne.

Les commandes en tension du solénoïde (U_s) et en intensité du moteur I_{tm} sont implémentées sur le CPIOM A qui héberge l'ensemble des applications de l'EBAS. Le CPIOM récupère également les mesures faites par les capteurs sur le système. Un agent HMon permet de surveiller et d'analyser le statut de santé des composants du système.

La description du principe de fonctionnement de la vanne permet de dégager certains paramètres du composant PRV impliqués dans la réalisation de la fonction de base Fu^{PRV} qui consiste à réguler la pression de l'air entre 30 psig et 40 psig. Le composant peut être modélisé par un triplet $\langle \mathcal{P}^{PRV}, \mathcal{R}^{PRV}, \mathcal{A}^{PRV} \rangle$. L'ensemble des paramètres (d'entrée, de sortie et privés) et des relations entre les paramètres du composant PRV sont illustrés sur la figure 6.5.

Paramètres d'entrée de la PRV : $\mathcal{IP}^{PRV} = \{U_s, I_{tm}, P_{up}, T_{up}, Q, P_a, T_a\}$.
- U_s est la tension appliquée au solénoïde,
- I_{tm} est le courant appliqué au moteur (TM),

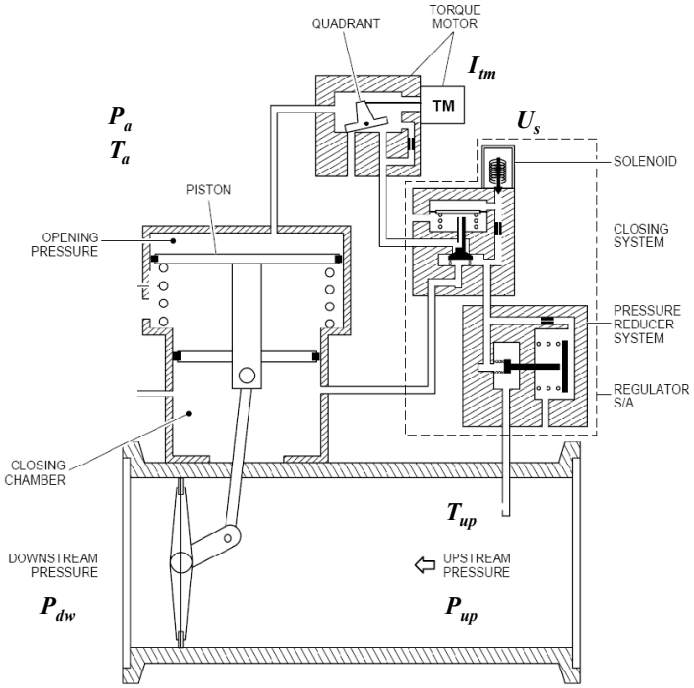

FIG. 6.4 – Schéma de principe de la PRV

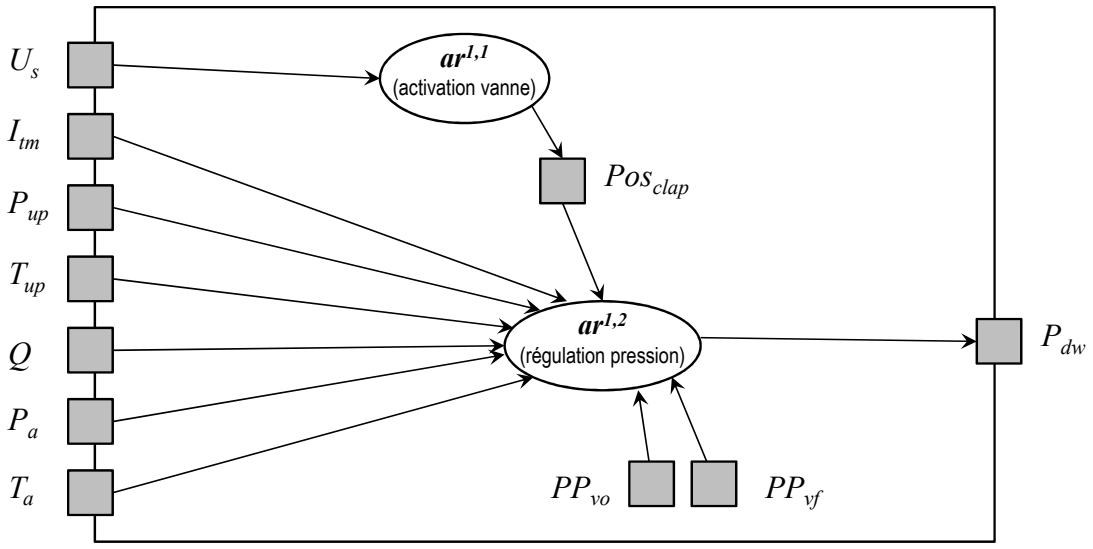

FIG. 6.5 – Modèle structurel de la PRV

- P_{up} est la pression mesurée en amont de la PRV par P_t,
- T_{up} est la température de l'air en amont de la PRV,
- Q est le débit de l'air sous pression,

- P_a est la pression de l'air ambiant et
- T_a est la température de l'air ambiant.

Paramètre de sortie de la PRV : $\mathcal{OP}^{PRV} = \{P_{dw}\}$.
- P_{dw} représente la pression de l'air mesurée en sortie de la PRV par le capteur P_r.

Paramètres privés de la PRV : $\mathcal{PP}^{PRV} = \{Pos_{clap}, PP_{vo}, PP_{vf}\}$.
- Pos_{cl} est un paramètre interne intermédiaire représentant la position du clapet qui permet de déclencher la réduction d'air lorsque la PRV est activée,
- PP_{vo} et PP_{vf} sont des paramètres associé à l'état du papillon de la vanne introduits pour le diagnostic. Ce sont des paramètres de fautes.

Condition fonctionnelle associée à Fu^{PRV} : la fonction élémentaire de régulation de pression est disponible sur la PRV si la condition fonctionnelle à laquelle elle est associée est vérifiée. Cette condition fonctionnelle se modélise à l'aide d'une relation entre les paramètres des composants :

$$\begin{array}{rcl} P_{dw} & = & ar^{1,2}(I_{tm}, P_{up}, T_{up}, Q, P_a, T_a, Pos_{cl}), \\ & = & ar^{1,2}(I_{tm}, P_{up}, T_{up}, Q, P_a, T_a, ar^{1,1}(U_s)). \end{array} \tag{6.1}$$

Des rangs sont associés à chaque paramètre du composant. La condition fonctionnelle associée à la fonction de base Fu^{PRV} est satisfaite lorsque les paramètres du système sont dans leur rang nominal de $\mathcal{R}_n^{PRV}$ défini par :

- $r_n(U_s) = 24\text{V}$, $r_n(P_a) = [2,3]$ psig, $r_n(T_a) = < 230^{\circ}\text{c}$, $r_n(I_{tm}) = [0,250]\text{mA}$, $r_n(T_{up}) = \leq 500^{\circ}\text{c}$, $r_n(P_{up}) = [30,140]$ psig,
- $r_n(P_{dw}) = [30,40]$ psig,
- $r_n(Pos_{cl}) = \{1\}$, $r_n(PP_{vo}) = \{0\}$, $r_n(PP_{vf}) = \{0\}$.

6.3.3 Diagnostic de la vanne de régulation de pression

Le diagnostic consiste à affecter un mode opérationnel m_x^{PRV} au composant PRV qui est cohérent avec les modèles disponibles du composant et les observations OBS^{PRV} mesurées en ligne par les capteurs sur la vanne. Des capteurs permettent de visualiser les valeurs des paramètres U_s, I_{tm}, P_{up}, Q et P_{dw} : $OBS^{PRV} = \{U_s, I_{tm}, P_{up}, Q, P_{dw}\}$.

Mode nominal Le modèle du mode nominal de la PRV est défini par $PRV_n = \langle(\mathcal{P}^{PRV}, \mathcal{R}_n^{PRV}), \mathcal{A}_n^{PRV}\rangle$. L'ensemble des paramètres $\mathcal{P}^{PRV} = (\mathcal{IP}^{PRV} \cup \mathcal{PP}^{PRV} \cup \mathcal{OP}^{PRV})$ est fixe quel que soit le mode opérationnel considéré. Les relations de $\mathcal{A}^{PRV} = \{ar^{1,1}, ar^{1,2}\}$ sont satisfaites et permettent la régulation de pression dans le mode nominal du composant. Le mode nominal m_n^{PRV} est défini par l'ensemble des rangs $\mathcal{R}_n^{PRV}$ donnés dans le paragraphe précédent. Ces valeurs représentent des conditions nominales de fonctionnement dans lesquelles la pression P_{up} est bien régulée par la PRV entre 30 et 40 psig.

Mode anormal Le composant PRV est dans un mode anormal m_a^{PRV} s'il existe un paramètre d'entrée de $\mathcal{IP}^{PRV}$ qui est en dehors de son rang nominal de $\mathcal{R}^{PRV}$.

Modes de faute Deux modes de faute dans lesquels la PRV n'assure plus sa fonction sont connus : un mode m_{fo}^{PRV} représentant la vanne bloquée en position ouverte et un mode m_{ff}^{PRV} représentant la vanne bloquée en position fermée.

Le mode de faute m_{fo}^{PRV} peut être détecté en mesurant une pression P_{up} supérieure au seuil de régulation maximale, c'est-à-dire à 40 psig. Les rangs des paramètres dans ce mode de faute sont différents de ceux définis pour le mode nominal : $r_{fo}(P_{dw}) = > 40$ psig et $r_{fo}(PP_{vo}) = \{1\}$.

Le mode de faute m_{ff}^{PRV} est détecté en mesurant une pression P_{up} inférieure au seuil de régulation minimale, c'est-à-dire à 30 psig. Les rangs des paramètres dans ce mode sont définis par : $r_{ff}(P_{dw}) = < 30$ psig et $r_{ff}(PP_{vf}) = \{1\}$.

Les moyens de détection des défaillances sur le composant PRV nous permettent de déterminer si le composant est en mode nominal ou en mode de faute directement à partir de la valeur mesurée par le capteur P_r. Le composant est donc ici diagnosticable c'est-à-dire que le mode de la PRV peut être directement déterminé à partir de l'observation de P_{dw} et des rangs définis pour le paramètre P_{dw} dans les différents modes.

Le HMon en charge de surveiller la PRV détecte une défaillance dès que $P_{dw} \notin [30, 40]$ psig et envoie le message de faute associé au mode diagnostiqué (*FailedClosed*, *FailedOpen*) au CMS. À l'envoi du message de faute, une alarme est déclenchée par le FWS pour avertir le pilote de la situation.

6.3.4 Pronostic de la vanne de régulation de pression

Le besoin d'une méthode de pronostic pour la maintenance d'un système aéronautique est illustrée sur le cas de la vanne de régulation de pression.

Lorsque le message de faute de la PRV *FaildedOpen* est envoyé au CMS, un message d'alarme informe le pilote que la PRV n'est pas correctement fermée. Le pilote peut alors effectuer une liste d'opérations de sécurité si elle est disponible. Dans le cas du message *FailedOpen*, le pilote peut commander le système de génération pneumatique par un commutateur OFF qui force l'arrêt du système. Le pilote rapporte ensuite l'alarme dans le logbook.

Les phases de maintenance sont périodiquement programmées après un certain nombres d'heures de vol et le plus souvent pendant la nuit. Des opérations de maintenance non programmées peuvent être tout de même effectuées au sol entre deux vols afin d'attendre la prochaine phase de maintenance programmée et faire repartir l'avion en éteignant les alarmes. Les opérateurs de maintenance effectuent les actions nécessaires

pour répondre à la plainte du pilote et éteindre l'alarme.

Dans le cas où la PRV est bloquée en position ouverte, les opérateurs de maintenance doivent la fermer manuellement pour que l'avion puisse repartir. Cette opération nécessite deux personnes et dure plus d'une heure. Elle ne peut être réalisée qu'une fois que le moteur de l'avion est froid, il faut pour cela attendre une heure. L'avion n'est alors pas disponible pendant presque trois heures. Une fois la vanne fermée, l'avion peut repartir mais la vanne devra néanmoins être remplacée au bout de dix jours. Ces opérations de maintenance non programmées sont très coûteuses en terme d'indisponibilité et de logistique.

La défaillance de la PRV aurait pu être prédite par une fonction de pronostic estimant son RUL. La fonction de pronostic aiderait alors à programmer la prochaine phase de maintenance en tenant compte des missions prévues de l'avion. La fonction de pronostic améliorerait également la logistique avec l'équipementier qui doit renouveler la PRV défaillante et permettrait de préparer l'opération de maintenance à un moment choisi précédant la date pronostiquée pour l'occurrence de la défaillance. Le gain en terme de disponibilité pouvant être apporté par le pronostic est évident dans ce cas. Il permettrait également de réduire les coûts directs et indirects de maintenance.

Pour estimer le RUL de la PRV, il faut disposer d'une connaissance sur son vieillissement. Les données relatives à l'usure des composants ne sont aujourd'hui pas disponibles pour établir un modèle de vieillissement tenant compte des sollicitations réelles de la vanne car le pronostic n'est pas encore considéré dans les avions. La seule information disponible pour estimer la défaillance de la PRV est son MTBF provenant d'une analyse de fiabilité : $MTBF(PRV) = 12,000\text{H}$. Cette connaissance permet de mettre en œuvre uniquement une politique de maintenance programmée. Cette connaissance non adaptative peut être représentée par un modèle de Weibull illustré sur la figure 6.6 qui représente la probabilité de défaillance de la PRV (associée indifféremment au mode de faute m_{fo}^{PRV} ou m_{ff}^{PRV}). Les caractéristiques η, β et θ de la distribution de Weibull sont fixées car aucune connaissance sur le vieillissement ne permet de les faire évoluer (voir le chapitre 4).

L'obtention d'un tel modèle de vieillissement repose sur l'étude des différents éléments constituant la vanne et repose sur l'analyse à un niveau très détaillé des phénomènes physico-chimiques appelés de manière générique "usure". Un exemple d'une telle étude concernant le colmatage de filtres, utilisés notamment en aéronautique, peut être consultée dans [BEN 05]. Ce type de résultat est sans aucun doute le passage obligé pour obtenir une fonction pronostic efficace.

6.4 Conclusion

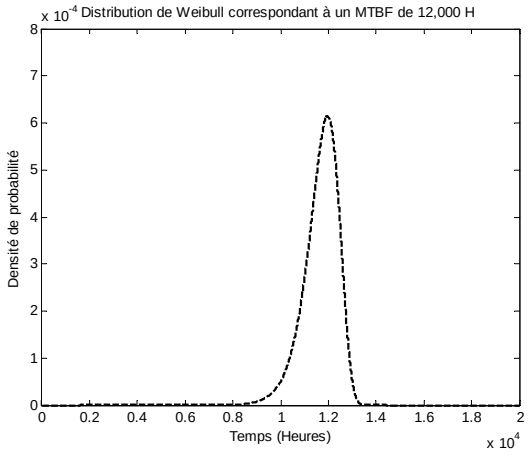

FIG. 6.6 – Distribution de Weibull associée au vieillissement de la PRV

L'objectif du projet ARCHISTIC est de concevoir une nouvelle architecture embarquée de surveillance pour optimiser la maintenance des avions du futur. L'architecture doit intégrer une fonction de diagnostic et une fonction de pronostic. Le problème de la détection de défaillances et du diagnostic est déjà pris en compte dans les avions et possède une certaine maturité. Ce chapitre décrit les différentes entités impliquées dans la construction d'un diagnostic de maintenance pour un système aéronautique. Le pronostic est un domaine émergent qui intéresse de plus en plus le domaine aéronautique car il s'agit d'un moyen de réduire les coûts de maintenance, d'améliorer la logistique avec les divers équipementiers et d'augmenter la disponibilité de l'avion.

L'exemple du système de génération pneumatique est étudié. La vanne de régulation de pression (PRV) est modélisée dans notre formalisme par un ensemble de paramètres, de relations et de rangs qui permettent de définir des modes opérationnels pour le diagnostic. L'intérêt d'une fonction de pronostic et les connaissances nécessaires pour une telle méthode sont illustrés sur la vanne de régulation de pression.

7 Conclusions et perspectives

Conclusions

Les travaux présentés dans cette thèse traitent de l'intégration du diagnostic et du pronostic pour optimiser la maintenance des systèmes complexes. Le diagnostic permet de déterminer en ligne les composants en faute du système qui doivent être réparés. Le pronostic permet d'anticiper les défaillances du système en fournissant des informations sur les états futurs des composants à partir desquels des actions de maintenance préventive peuvent être envisagés.

Cette thèse a pour objectif de développer une architecture de supervision qui intègre des capacités de diagnostic et de pronostic dans le but d'aider à la prise de décision d'actions de maintenance pour un système complexe.

Un système complexe est un ensemble de composants hétérogènes. La connaissance sur chaque composant peut être très différente d'un composant à l'autre et plusieurs types de techniques sont nécessaires pour surveiller et analyser leur état de santé. Le chapitre 1 montre l'intérêt d'une stratégie de maintenance préventive et dégage dans un premier temps les modèles et méthodes qui s'appuient sur une connaissance plus ou moins approfondie des composants pour répondre au problème du diagnostic et du pronostic.

Le premier apport de cette thèse est la proposition d'un cadre de modélisation formel pour les systèmes complexes (Chapitre 2). L'idée est de tenir compte de l'hétérogénéité des composants pour représenter de manière abstraite mais homogène la connaissance disponible pour chaque composant du système complexe. Cette modélisation s'appuie sur un ensemble de paramètres caractéristiques des composants, un ensemble de rangs pour ces paramètres ainsi qu'un ensemble de relations entre les paramètres. À des fins de diagnostic et de pronostic, des modes de fonctionnements (modes opérationnels) ont été définis pour les composants.

Ce formalisme générique s'appuyant sur les notions de paramètres et de modes opérationnels est utilisé pour caractériser les problèmes de diagnostic et de pronostic (Chapitre 3). Nous avons établi dans cette thèse un couplage diagnostic-pronostic ori-

ginal en définissant le problème de pronostic comme une extension du problème de diagnostic. Le diagnostic consiste à déterminer la séquence de modes qu'a suivi le système en calculant l'ensemble des séquences possibles cohérentes avec les modèles et les observations du système. Le pronostic consiste à prédire la succession de modes de faute futurs du système qui a la plus grande probabilité de se produire selon les modèles de vieillissement des composants à partir d'un candidat de diagnostic.

Afin d'évaluer la séquence future de modes du système qui a la plus grande probabilité de se produire, une fonction générique de pronostic a été définie à l'aide d'un modèle de Weibull (Chapitre 4). Quel que soit le type de modèle de vieillissement disponible, cette fonction de pronostic établit une probabilité de faute pour chaque paramètre privé des composants. Dans un système complexe, les composants pouvant être redondants, il est nécessaire d'obtenir également une probabilité de défaillance associée à la disponibilité de chaque fonction mise en œuvre par les composants du système. La durée de vie résiduelle du système complexe est ensuite évaluée à partir de la composition des pronostics de fonctions.

Des critères de performance sont déterminés pour évaluer les résultats de diagnostic et de pronostic fournis par l'architecture de supervision (Chapitre 5). Ces critères reposent sur les propriétés de diagnosticabilité, de pronosticabilité et de précision d'un diagnostic et d'un pronostic que nous avons défini à partir du formalisme générique. Nous avons proposé un retour sur conception du système complexe afin d'assurer la performance de la fonction de diagnostic par une étude de diagnosticabilité. Ce problème de retour sur conception est défini comme un problème d'optimisation de coût en tenant compte du coût de conception du système mais également des coûts liés à l'architecture de surveillance. Nous avons développé une méthodologie dans le cadre des systèmes à événements discrets distribués qui établit des recommandations de conception sur les spécifications pour les composants à surveiller. L'objectif de cette méthodologie est de garantir ou d'améliorer la diagnosticabilité d'un système à événements discrets distribué en augmentant l'observabilité des composants par des techniques de placement de capteurs et en minimisant le coût d'intégration du système (coût de conception et de surveillance).

Le dernier chapitre de cette thèse porte sur le projet ARCHISTIC en collaboration avec Airbus et l'École Nationale d'Ingénieurs de Tarbes. L'objectif de ce projet était de concevoir une nouvelle architecture embarquée de surveillance pour les avions du futur. Nous avons modélisé un composant d'un système aéronautique à l'aide de notre formalisme et défini des modes opérationnels afin d'illustrer la caractérisation du diagnostic que nous avons proposé dans cette thèse. L'intérêt pour le pronostic est croissant mais, dans le milieu industriel, les moyens pour considérer les données liées au vieillissement des composants ne sont pas toujours disponibles. L'étude menée au cours de cette thèse met en lumière l'impossibilité de réaliser une fonction de pronostic ayant une véritable valeur ajoutée sans ce type de données.

La plupart des résultats obtenus dans cette thèse ont été présentés dans le cadre

du projet ARCHISTIC. Les différents thèmes étudiés (diagnostic, pronostic, critères de performance) étaient directement liés aux attentes des industriels d'Airbus dans l'objectif d'améliorer la maintenance des avions du futur. Notre rôle dans ces études a porté sur la formalisation des problèmes posés par la maintenance d'un système complexe et hétérogène.

Discussions et perspectives

Tout au long de ce manuscrit, des hypothèses sur le problème ou sa résolution sont posées. Il serait intéressant de les remettre en cause pour des développements à venir.

Modes opérationnels

Dans le chapitre 2, nous avons défini les modes opérationnels des composants à partir des paramètres privés et des paramètres d'entrée qui peuvent évoluer en dehors de leur rang défini pour le mode nominal. Il serait intéressant de considérer un nouveau type de mode pour un composant C correspondant au cas où tous les paramètres du composant (paramètres d'entrée, de sortie et privés) sont dans leur rang défini pour le mode nominal mais où une fonction élémentaire F n'est plus disponible sur C. Seule la relation entre les paramètres modélisant la condition fonctionnelle de la fonction F n'est donc pas respectée. Pour répondre au problème de diagnostic, selon les capacités de surveillance du système, il faudrait pouvoir identifier ce nouveau mode traduisant la violation d'une relation, d'un mode de faute et d'un mode anormal tels que nous les avons définis. En introduisant un nouveau type de mode opérationnel, on lève l'hypothèse de complétude du modèle.

Sélection des modèles de vieillissement

Dans le chapitre 3, nous supposons qu'une application lv permet de déterminer le modèle de vieillissement d'un paramètre privé en fonction du mode opérationnel du composant auquel il appartient. Il faudrait préciser comment cette application permet d'obtenir le bon modèle de vieillissement d'un paramètre en fonction des conditions opérationnelles du composant. Il serait également intéressant de considérer le cas où le modèle de vieillissement d'un paramètre est sélectionné dans plusieurs modes du composant. Des modes de dégradation pourraient alors être définis selon le niveau de stress induit sur le composant par leurs conditions opérationnelles.

Dans le cas où le diagnostic global obtenu contient plusieurs candidats, il y a une ambiguïté sur les modes des composants du système complexe. Plusieurs modèles de vieillissement sont alors sélectionnés pour les paramètres privés des composants en considérant l'ensemble de modes possibles du système. Il faudrait trouver une méthode permettant

de sélectionner le modèle de vieillissement le plus approprié pour chaque paramètre du système. Une solution possible serait de probabiliser le diagnostic global du système, c'est-à-dire d'associer une distribution de probabilité à chaque candidat de diagnostic de manière à avoir un pronostic plus absolu.

Représentation du pronostic par le modèle de Weibull

Dans le chapitre 4, nous proposons de représenter la probabilité de faute d'un paramètre privé par un modèle de Weibull. Ce modèle est très intéressant par sa flexibilité qui permet de représenter une infinité de lois de probabilité. Dans notre étude, nous considérons des composants totalement hétérogènes, le modèle de Weibull apparaissait alors comme un bon moyen pour décrire les différents types de connaissance disponible sur chaque composant. On peut néanmoins se poser la question de l'obtention des caractéristiques du modèle de Weibull. Des hypothèses ont été faites sur la région de vie à considérer pour les composants et devraient être levées dans le futur pour considérer toutes les phases de vie d'un composant quel que soit le modèle de vieillissement disponible pour les paramètres. Nous présentons une manière d'adapter le pronostic en fonction des facteurs de stress en mettant à jour la caractéristique η du modèle de Weibull. Il faut pour cela pouvoir quantifier d'une façon générique les différents facteurs de stress qui sollicitent réellement le composant : occurrence d'une faute dans ce même composant ou dans un composant interagissant, conditions environnementales, ... Il s'agira ensuite de pouvoir illustrer concrètement l'application de ces distributions de Weibull sur un cas industriel réel.

Méthodologie de retour sur conception dans les SED

Le problème du placement de capteurs est étudié dans le chapitre 5 pour améliorer la diagnosticabilité d'un système à événements discrets distribué. Il pourrait être étendu pour le pronostic en considérant des capteurs spécifiques pouvant fournir des informations sur l'usure des composants. La méthodologie de retour sur conception serait alors modifiée pour établir des recommandations sur les spécifications des composants dans l'objectif de rendre le système à la fois diagnosticable et pronosticable. Cette méthodologie serait également guidée par les propriétés de précision du diagnostic et du pronostic afin de limiter le coût lié à la surveillance du système complexe. La monotonie de ces différentes propriétés doit être étudiée plus en détail de manière à appréhender les impacts des modifications réalisées dans les spécifications des composants sur l'ensemble des propriétés à garantir.

Les perspectives de ces travaux de thèse s'orientent dans quatre directions :
- la prise en compte d'un mode opérationnel dans lequel une relation n'est plus respectée alors que tous les paramètres du composant sont dans leur rang nominal,
- la sélection du modèle de vieillissement le plus approprié pour les paramètres lorsqu'un ensemble de modes possibles pour le système est déterminé par le diagnostic,
- la détermination des caractéristiques du modèle de Weibull,
- la généralisation de la méthode de retour sur conception pour rendre le système diagnosticable et pronosticable.

Bibliographie

[Afn] "Norme AFNOR NF EN 13306 "Terminologie de la maintenance", Ed. Afnor, Paris 2001.".

[AGU 99] J. Aguilar-Martin, "Knowledge-based supervision and diagnosis of complex process", proceedings of *IEEE International Symposium on Intelligent Control, Intelligent Systems and Semiotics (ISI'99)*, Cambridge, USA, 1999, pages 225–230.

[BAL 88] P. Balle and R. Isermann, "Fault detection and isolation for nonlinear processes based on local linear fuzzy models and parameter estimation", proceedings of *American Control Conference*, 1988, pages 1605–1609.

[BAS 93] M. Basseville and L. Nikiforov, *Detection of abrupt changes*, Prentice Hall, Englewood Cliffs, NJ., 1993.

[BAY 08] M. Bayoudh, L. Travé-Massuyès and X. Olive, "Hybrid systems diagnosis by coupling continuous and discrete event techniques", proceedings of *the 17th IFAC World Congress*, Seoul, Corée du Sud, July 2008.

[BEN 05] K. Benmachou, "Etude et modélisation du colmatage d'un filtre plissé", PhD thesis, Institut National Polytechnique de Toulouse, 2005.

[BER 06] H. Berenji and Y. Wang, "Case-Based Reasoning for Fault Diagnosis and Prognosis", proceedings of *the IEEE International Conference on Fuzzy Systems*, July 2006, pages 1316-1321.

[BIS 04] G. Biswas, M. Cordier, J. Lunze and L. Travé-Massuyès, "Diagnosis of complex systems : Bridging the methodologies of the FDI and DX communities", *IEEE Transactions on Systems, Man and Cybernetics - Part B. Special section*, vol. 34, no. 5, 2004, pages 2159–2244.

[BIS 06] G. Biswas and E. Manders, "Integrated systems health management to achieve in complex systems", proceedings of *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process (SAFEPROCESS)*, Beijing, China P.R., 2006, pages 1207-1212.

[BRI 08] L. Briones, A. Lazovik and P. Dague, "Optimal observability for diagnosability", proceedings of *19th International Workshop on Principles of Diagnosis*, A. Grastien, W. Meyer and M. Stumptner, editors, Blue Mountains, Australia, 2008, pages 31–38.

[BRO 00] T. Brotherton, G. Jahns, J. Jacobs and D. Wroblewski, "Prognosis of Faults in Gas Turbine Engines", proceedings of *IEEE Aerospace Conference Proceedings*, vol. 6, Big Sky, MT, USA, 2000, pages 163-171.

[BUC 84] B. Buchanan and E. Shortliffe, *Rule Based Expert Systems : The Mycin Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984.

[BYI 02] C. Byington, M. Roemer and T. Galie, "Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance", proceedings of *IEEE Aerospace Conference Proceedings*, 2002, pages 2815-2824.

[BYI 04] C. Byington, P. Kalgren, B. Dunkin and B. Donovan, "Advanced Diagnostic/Prognostic Reasoning and Evidence Transformation Techniques for Improved Avionics Maintenance", proceedings of *IEEE Aerospace Conference Proceedings*, vol. 5, 6-13 March 2004, pages 3424-3434.

[CAM 07] F. Camci, G. Valentine and K. Navarra, "Methodologies for Integration of PHM Systems with Maintenance Data", proceedings of *IEEE Aerospace Conference Proceedings*, 3-10 March 2007.

[CAO 89] X. CAO, "The predictability of discrete event systems", *IEEE Transactions Automatic Control*, vol. 34, no. 11, 1989, pages 1168–1171.

[CHI 93] L. Chitttaro, G. Guida, C. Tasso and E. Toppano, "Functional and Teleological Knowledge in the Multimodeling Approach for Reasoning about Physical Systems : A Case Study in Diagnosis", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, 1993, pages 1718–1751.

[CHO 84] E. Chow and A. Willsky, "Analytical redundancy and the design of robust detection systems", *IEEE Transactions On Automatic Control*, vol. 29, 1984, pages 603–614.

[CON 91] L. Console and P. Torasso, "A spectrum of logical definitions of model-based diagnosis", *Computational intelligence*, vol. 7, no. 3, 1991, pages 133–141.

[CON 02] L. Console, C. Picardi and M. Ribaudo, "Process algebra for systems diagnosis", *Artificial Intelligence*, vol. 142, 2002, pages 19–51.

[CON 06] O. Contant, S. Lafortune and D. Teneketzis, "Diagnosability of Discrete Event Systems with Modular Structure", *Discrete Event Dynamical Systems*, vol. 16, 2006, pages 9–37.

[COQ 05] V. Coquempot, T. El Mezyani and M. Staroswiecki, "Hybrid dynamical system monitoring using structured analytical redundancy relations", proceedings of *17ème IMACS World Congress*, Paris, France, 2005.

[COR 04] M. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroviecki and L. Travé-Massuyès, "Conflicts versus analytical redundancy relations : a comparative analysis of the model-based diagnostic approach from the articficial intelligence and automatic control perpectives", *IEEE Transactions on Systems, Man and Cybernetics - Part B.*, vol. 34, no. 52163-2177, 2004.

[DEB 99] R. Debouk, S. Lafortune and D. Teneketzis, "On an Optimization Problem in Sensor Selection for Failure Diagnosis", proceedings of *38th Conference on Decision and Control*, Phoenix, Arizona, USA, December 1999, pages 4990-4995.

[DEB 02] R. Debouk, S. Lafortune and D. Teneketzis, "Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems", *JDEDS : Theory and Application*, vol. 10, no. 1–2, 2002, pages 33–86.

[DEK 87] J. De Kleer and B. Williams, "Diagnosing multiple faults", *Artificial Intelligence*, vol. 32, no. 1, 1987, pages 97–130.

[DEK 89] J. De Kleer and B. Williams, "Diagnosis with behavioral modes", proceedings of *11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, 1989, pages 1324–1330.

[DEK 92] J. De Kleer, A. Mackworth and R. Reiter, "Characterizing diagnoses and systems", *Artificial Intelligence*, vol. 56, no. 2-3, 1992, pages 197–222.

[DEL 09] E. Deloux, B. Castanier and C. Bérenguer, "Predictive maintenance policy for a gradually deteriorating system subject to stress", *Reliability Engineering and System Safety*, vol. 94, 2009, pages 418–431.

[DES 81] M. Desai and A. Ray, "A fault detectetion and isolation methodology", proceedings of *the 20th IEEE Conference on Decision and Control*, 1981, pages 1363–1369.

[DES 84] M. Desai and A. Ray, "A fault detection and isolation methodology. Theory and application", proceedings of *Proc. of American Control Conference*, 1984, pages 262—270.

[DRA 07] O. Dragomir, R. Gouriveau, N. Zerhouni and F. Dragomir, "Framework for a distributed and hybrid prognostic system", proceedings of *4th IFAC Conference Management and Control of Production and Logistics (MCPL'2007)*, Romania, 2007, pages 431–436.

[DUB 90] B. Dubuisson, *Diagnostic et reconnaissance de formes*, Hermes, 1990.

[DUB 01] B. Dubuisson, *Diagnostic, Intelligence Artificielle et Reconnaissance de Formes*, Hermes, 2001.

[EIS 97] R. S. Eisenmann and R. J. Eisenmann, "Applied condition monitoring", *Machinery malfunction diagnosis and correction*, vol. 13, 1997, pages 703–741.

[ENG 00] S. Engel, B. Gilmartin, K. Bongort and A. Hess, "Prognostics, The Real Issues Involved With Predicting Life Remaining", proceedings of *IEEE Aerospace Conference*, vol. 6, USA, 2000, pages 457-469.

[FAB 05] E. Fabre, A. Benveniste, S. Haar and C. Jard, "Distributed Monitoring of Concurrent and Asynchronous Systems", *Journal of Discrete Event Dynamic Systems*, vol. 15, 2005, pages 33–83.

[FEL 07] R. Fellouah, "Contribution au diagnostic de pannes pour les systèmes différentiellement plats", PhD thesis, Institut National des Sciences Appliquées de Toulouse, 2007.

[FER 08] S. Ferreiro and A. Arnaiz, "Prognosis Based on Probabilistic Models and Reliability Analysis to improve aircraft maintenance", proceedings of *the International Conference on Prognostics and Health Management, PHM'08*, Denver, USA, 2008.

[FOU 03] M. Fouladirad and I. Nikiforov, "Optimal statistical fault detection with nuisance parameters", proceedings of *the American Control Conference*, Denver, Colorado, USA, 2003.

[FRA 87] P. Frank, "Advanced fault detection and isolation schemes using nonlinear and robut observers", proceedings of *10th IFAC Congress*, vol. 3, Munich, Germany, 1987, pages 63–68.

[FRA 96] P. Frank, "Analytical and qualitative model-based fault diagnosis, a survey and some new results", *European Journal of Control*, vol. 2, 1996, pages 6–28.

[GEN 06] S. Genc and L. Lafortune, "Predictability in discrete-event systems under partial observations", proceedings of *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process (SAFEPROCESS)*, Beijing, China P.R., 2006, pages 1531–1536.

[GER 95] J. Gertler, "Diagnosing parametric faults : from parameter estimation to parity relations", proceedings of *the American Control Conference*, 1995, pages 1615–1620.

[GER 98] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*, Marcel Deker, 1998.

[GHE 06] S. Ghelam, Z. Simeu-Abazi, J.-P. Derain, C. Feuillebois, S. Vallet and M. Glade, "Integration of Health Monitoring in the Avionics Maintenance System", proceedings of *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process (SAFEPROCESS)*, Beijing, China P.R., 2006, pages 1519-1524.

[GOE 07] K. Goebel and N. Eklund, "Prognostic Fusion for Uncertainty Reduction", proceedings of *Proceedings of AIAA Infotech@ Aerospace Conference*, Rohnert Park, California, 7-10 May 2007.

[GOH 06] K. Goh, B. Tjahjono, T. Baines and S. Subramaniam, "A Review of Research in Manufacturing Prognostics", proceedings of *Proceedings of the IEEE International Conference on Industrial Informatics*, New York, 2006, pages 417-422.

[HAM 92] W. Hamscher, L. Console and J. De Kleer, *Readings in model-based diagnosis*, Morgan Kaufmann Publishers Inc., 1992.

[HAM 99] H. Hammouriand, M. Kinnaert and E. Yaagoubi, "Observer-based approach to fault detection and isolation for non-linear systems", *IEEE Transactions and Automatic Control*, vol. 44, 1999, pages 1879–1884.

[HER 06] H. Hernandez De Leon, "Supervision et diagnostic des procédés de production d'eau potable", PhD thesis, Institut National des Sciences Appliquées, Toulouse, 2006.

[HOH 09] C. Hohmann, *Techniques de productivité*, 2009.

[HUM 02] D. Humphrey, W. Schawlee, P. Sandborn and D. Lorenson, "Utilization Life of Electronic Systems - Aging avionics usable life and wear-out issues", proceedings of *World Aviation Congress*, Phoenix, AZ, USA, November 2002.

[ISE 84] R. Isermann, "Process fault detection based on modeling and estimation methods - a survey", *Automatica*, vol. 20, 1984, pages 387–404.

[ISE 93a] R. Isermann, "Fault diagnosis of machines via parameter estimation and knowledge processing : tutorial paper", *Automatica*, vol. 29, no. 4, 1993, pages 815–835.

[ISE 93b] R. Isermann, "Process fault detection based on modeling and estimation methods - a survey", *Automatica*, vol. 29, 1993, pages 815–835.

[ISE 97] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods. An introduction", *Control Engineering Practice*, vol. 5, 1997, pages 639–652.

[JAC 98] P. Jackson, *Introduction to Expert Systems*, Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1998.

[JAR 87] A. Jardine, *Maintenance, replacement and reliability*, Pitman, Boston, MA, 1987.

[JAR 06] A. Jardine, D. Lin and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance", *Mechanical systems and signal processing*, vol. 20, no. 7, 2006, pages 1483–1510.

[JER 07] T. Jeron, H. Marchand, S. Genc and S. Lafortune, "Predictability of Sequence Patterns in Discrete Event Systems", *IRISA internal publication*, vol. 1834, 2007.

[JIA 01] S. Jiang, Z. Huang, V. Chandra and R. Kumar, "A Polynomial Time Algorithm for Diagnosability of Discrete Event Systems", *IEEE Transactions on Automatic Control*, vol. 46, no. 8, 2001, pages 1318–1321.

[JIA 03] S. Jiang, R. Kumar and H. E. Garcia, "Optimal Sensor Selection for Discrete Event Systems Under Partial Observation", *IEEE Transactions on Automatic Control*, vol. 48, 2003, pages 369-381.

[KAU 75] A. Kaufman, D. Grouchko and R. Cruon, *Modèles mathématiques pour l'étude de la fiabilité des systèmes*, 1975.

[KEL 07] K. Keller, "Health management technology integration", proceedings of *the 18th International Workshop on Principles of Diagnosis (DX-07)*, Nashville, USA, May 2007.

[KEM 04] T. Kempowsky, "Surveillance de procédés à base de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic des défaillances", PhD thesis, Institut National des Sciences Appliquées, Toulouse, 2004.

[KIR 04] L. Kirkland, T. Pombo, K. Nelson and F. Berghout, "Avionics Health Management : Searching for the Prognostics Grail", proceedings of *IEEE Aerospace Conference*, vol. 5, March 6-13 2004, pages 3448-3454.

[KOT 06] R. Kothamasu, S. H. Huang and W. VerDuin, "System health monitoring and prognostics : a review of current paradigms and practices", *The International Journal of Advanced Manufacturing Technology*, vol. 28, no. 9–10, 2006, pages 1012–1024.

[LAM 03] G. Lamperti and M. Zanella, *Diagnosis of active systems*, Kluwer Academic Publishers, 2003.

[LEB 01] M. Lebold and M. Thurston, "Open Standards for Condition-Based Maintenance and Prognostic Systems", proceedings of *Maintenance and Reliability Conference*, 2001.

[MAG 94] J. Magni and P. Mouyon, "On residual generation by observer and parity space approaches", *IEEE Transactions on Automatic Control*, vol. 39, 1994, pages 441–447.

[MAQ 97] D. Maquin, V. Cocquempot, J. Cassar, M. Staroswiecki and J. Ragot, "Generation of analytical redundancy relations for FDI purposes", proceedings of *IFAC Sympoisum on Diagnosis for Electrical Machines, Power Electronics and Drives (SDEMPED'97)*, 1997.

[MIR 80] L. Mironovski, "Functional diagnosis of dynamic system - a survey", *Remote Control*, vol. 41, 1980, pages 1122–1143.

[MOB 90] R. Mobley, *An introduction to preventive maintenance : plant engineering series*, Van Nostrand, New York, 1990.

[NAR 98] S. Narasimhan, P. Mosterman and G. Biswas, "A Systematic Analysis of Measurement Selection Algorithms for Fault Isolation in Dynamic Systems", proceedings of *the 9th International Workshop on Principles of Diagnosis*, 1998.

[NAR 03] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems", proceedings of *the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, Acapulco, Mexico, 2003, pages 376–381.

[NAR 07] C. I. Narvaez, "Diagnostic par techniques d'apprentissage floues : concept d'une méthode de validation et d'optimisation des partitions", PhD thesis, Université Paul Sabatier, Toulouse, 2007.

[PAT 91] R. Patton and J. Chen, "A re-examination of the relationship between parity space and observer-based approaches in fault diagnosis", *Revue européenne Diagnostic et Sûreté de Fonctionnement*, vol. 1, no. 2, 1991, pages 183–200.

[PEN 02] Y. Pencolé, "Diagnostic décentralisé de systèmes à événements discrets : application aux réseaux de télécommunications", PhD thesis, Université de Rennes 1, IRISA, 2002.

[PEN 04] Y. Pencolé, "Diagnosability analysis of distributed event systems", proceedings of *European Conference on Artificial Intelligence*, Valencia, Spain, August 2004.

[PEN 05] Y. Pencolé and M.-O. Cordier, "A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks", *Artificial Intelligence*, vol. 164, 2005, pages 121–170, Elsevier.

[RAM 87] R. Ramadge and W. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control and Otpimization*, vol. 25, no. 1, 1987, pages 206–230.

[RAU 04] M. Rausand and A. Hoyland, *System reliability theory : models, statistical methods and applications*, Wiley, 2004.

[REI 87] R. Reiter, "A theory of diagnostic from first principles", *Artificial Intelligence*, vol. 32, 1987, pages 57–95.

[RIB 07] P. Ribot, Y. Pencolé and M. Combacau, "Characterization of requirements and costs for the diagnosability of distributed discrete event systems", proceedings of *5th Workshop on Advanced Control and Diagnosis (ACD'07)*, Grenoble, November 15-16 2007.

[RIB 08] P. Ribot, Y. Pencolé and M. Combacau, "Prognostics for the Maintenance of Distributed Systems", proceedings of *the International Conference on Prognostics and Health Management (PHM'08)*, Denver, USA, 6-10 October 2008.

[RIB 09] P. Ribot, Y. Pencolé and M. Combacau, "Functional prognostic architecture for the maintenance of complex systems", proceedings of *the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SafeProcess'09)*, Barcelona, Spain, July 1-3 2009.

[ROE 05] M. Roemer, C. Byington, G. Kacprzynski and G. Vachtsevanos, "An Overview of Selected Prognostic Technologies with Reference to an Integrated PHM Architecture", proceedings of *the First International Forum on Integrated System Health Engineering and Management in Aerospace*, 2005.

[SAM 95] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis, "Diagnosability of Discrete Event System", *IEEE Transactions on Automatic Control*, vol. 40, no. 9, 1995, pages 1555–1575.

[SAM 08] B. Samanta and C. Nataraja, "Prognostics of machine condition using soft computing", *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 6, 2008, pages 816–823.

[SCH 03] A. Schwarte, F. Kimmich and R. Isermann, "Model-based fault detection of a diesel engine with turbo charger - a case study", proceedings of *Conference Safeprocess'2003*, Washington D.C., USA, 2003.

[SCH 07a] A. Schumann and Y. Pencolé, "Scalable Diagnosability Checking of Event-Driven System", proceedings of *20th International Joint Conference on Artificial Intelligence (IJCAI07)*, 2007.

[SCH 07b] M. Schwabacher and K. Goebel, "A survey of Artificial Intelligence for Prognostics", proceedings of *AAAI Fall Symposium*, 2007.

[SHE 94] C. Sheu and L. Krajewski, "A decision model for corrective maintenance management", *IJPR*, vol. 32, no. 6, 1994, pages 1365–1382.

[SPA 04] S. Spanache, T. Escobet and L. Travé-Massuyès, "Sensor placement optimisation using genetic algorithms", proceedings of *15th International Workshop on Principles of Diagnosis (DX'04)*, Carcassonne (France), 11-14 Juin 2004, pages 179-184.

[STA 89] M. Staroswiecki and P. Declerck, "Analytical redundancy in nonlinear interconnected systems by means of structural analysis", proceedings of *IFAC AIPAC'89 Symposium*, vol. 2, 1989, pages 23–27.

[STA 91] M. Staroswiecki, V. Cocquempot and J. Cassar, "Observer based and parity space approaches for failure detection and identification", proceedings of *IMACSIFAC International Symposium*, 1991, pages 536–541.

[STA 01] M. Staroswiecki and G. Comtet-Varga, "Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems", *Automatica*, vol. 37, no. 5, 2001, pages 687–699.

[SU 04] R. Su and W. Wonham, "Distributed diagnosis under global consistency", proceedings of *43 rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 14-17 2004, pages 525-530.

[TAK 85] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control", proceedings of *IEEE International Conference on Systems, Man and Cybernetics*, vol. 15, 1985, pages 116–132.

[THO 07] D. Thorsley and D. Teneketzis, "Active acquisition of information for diagnosis and supervisory control of discrete event systems", *Discrete Event Dynamic Systems*, vol. 17, no. 4, 2007, pages 531–583.

[TOR 07] G. Torta and P. Torasso, "Computation of Minimal Sensor Sets from Precompiled Disciminability Relations", proceedings of *18th International Worshop on Principles of Diagnosis (DX'07)*, Nashville, TN, USA, May 2007, pages 202-209.

[VAC 06] G. Vachtsevanos, F. L. Lewis, M. Roemer, A. Hess and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, Wiley, 2006.

[VAL 03] R. Valentin, M. Osterman and B. Newman, "Remaining Life Assessment of Aging Electronics in Avionic Applications", proceedings of *Annual Reliability and Maintainability Symposium*, 2003, pages 313-318.

[VEN 03] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri and K. Yin, "A review of precess fault detection and diagnosis - Part III : Process history based methods", *Computers and Chemical Engineering*, vol. 27, 2003, pages 327–346.

[VIL 88] A. Villemur, *Sûreté de fonctionnement des systèmes industriels : Fiabilité, Facteurs humains, Informatisation*, Eyrolles, Paris, 1988.

[VOI 09] A. Voisin, E. Levrat, P. Cocheteux and B. Iung, "Generic prognosis model for proactive maintenance decision support : application to pre-industrial e-maintenance test bed", *Journal of Intelligent Manufacturing*, , 2009, page Accepted article.

[WAN 99] P. Wang and G. Vachtsevanos, "Fault prognosis using dynamic wavelet neural networks", proceedings of *AAAI Symposium*, Polo, Alto, March 22-24 1999.

[WIL 04] C. Wilkinson, D. Humphrey, B. Vermeire and J. Houston, "Prognostic and Health Management for Avionics", *IEEE Aerospace Conference Proceedings*, vol. 5, 2004, pages 3435-3446.

[YOO 02] T. Yoo and S. Lafortune, "Polynomial-Time Verification of Diagnosability of Partially-Observed Discrete-Event Systems", *IEEE Transactions of Automatic Control*, vol. 47, no. 9, 2002, pages 1491–1495.

[ZAD 65] L. Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, no. 3, 1965, pages 338–353.

[ZWI 95] G. Zwingelstein, *Diagnostic des défaillances : Théorie et pratique pour les systèmes industriels*, Hermes, 1995.

Pauline RIBOT

Vers l'intégration diagnostic/pronostic pour la maintenance des systèmes complexes

Directeurs de thèse : Yannick Pencolé et Michel Combacau

Discipline : Systèmes automatiques

Présentée le 4 décembre 2009 au LAAS-CNRS, Toulouse

Résumé

L'efficacité de la maintenance des systèmes industriels est un enjeu économique majeur pour leur exploitation commerciale. Les principales difficultés et sources d'inefficacité résident dans le choix des actions de maintenance. Un mauvais choix peut mener à une maintenance non satisfaisante et un surcoût dû à l'indisponibilité du système.

Cette thèse propose une architecture générique de supervision pour aider à la prise de décisions d'actions de maintenance pour un système complexe. Cette architecture intègre des capacités de diagnostic et de pronostic permettant de connaître l'état actuel et l'état futur du système. La fonction de diagnostic détermine les composants en faute à l'origine des défaillances. La fonction de pronostic calcule la durée avant la prochaine défaillance du système.

Nous présentons un cadre de modélisation générique formel pour un système complexe qui capture l'ensemble des connaissances nécessaires aux fonctions de diagnostic et de pronostic. Il permet de caractériser un couplage diagnostic/pronostic original. Une fonction générique et adaptative de pronostic est définie à l'aide d'un modèle de Weibull afin d'évaluer de façon probabiliste la durée de vie résiduelle du système. Des critères de performance pour l'architecture de supervision proposée reposant sur des propriétés du diagnostic et du pronostic sont caractérisés. Une méthodologie de retour sur conception est proposée dans le but d'assurer la performance de la fonction de diagnostic en garantissant la diagnosticabilité du système.

L'application de ce travail de recherche aux systèmes aéronautiques s'inscrit dans le cadre du projet ARCHISTIC en collaboration avec Airbus et l'ENIT.

Mots clés : diagnostic, pronostic, supervision, maintenance préventive, diagnosticabilité, systèmes complexes.

Laboratoire d'Analyse et d'Architecture des Systèmes - UPR 8001
7, avenue du Colonel Roche, 31077 Toulouse Cedex 4

Pauline RIBOT

Towards diagnosis/prognosis integration for the maintenance of complex systems

Supervisors : Yannick Pencolé and Michel Combacau

Abstract

The maintenance efficiency of industrial systems is an important economical and business issue. The main difficulties come from the choice of maintenance actions. A wrong choice can lead to maintenance costs that are not acceptable.

A generic supervision architecture is proposed in this thesis and supports the decision of maintenance actions for a complex system. This architecture integrates some diagnostic and prognostic capabilities to determine the current and future state of the system. The diagnostic function aims at identifying faulty components that may cause system failures. The prognostic function estimates the remaining time until the next system failure.

A formal generic modeling framework for a complex system is presented and encapsulates the knowledge that is used by the diagnostic and prognostic functions. An original coupling of diagnosis and prognosis is then characterised. A generic and adaptive prognostic function is defined with a Weibull model in order to evaluate the system remaining useful life by the use of probability. Performance criteria for the proposed supervision architecture are characterised. They rely on diagnostic and prognostic properties. A methodology that provides a feedback to the system design is proposed to improve diagnostic function efficiency by guaranteeing some diagnosability objectives.

This work is partly supported by the ARCHISTIC project in collaboration with Airbus France and National Engineering School of Tarbes, France.

Key words : diagnosis, prognosis, supervision, preventive maintenance, diagnosability, complex systems.