Contribution à la mise au point d'une approche intégrée analyse diagnostique / analyse de risques

Matthieu Desinde

Université Joseph Fourier Grenoble 1

No. attribué par la bibliothèque

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER - GRENOBLE

***Spécialité :* AUTOMATIQUE-PRODUCTIQUE**

préparée au Laboratoire d'Automatique de Grenoble

dans le cadre de l'École Doctorale :
Électronique, Électrotechnique, Automatique, Traitement du Signal

présentée et soutenue publiquement

par

Matthieu DESINDE

le 13 Décembre 2006

Titre :

Contribution à la mise au point d'une approche intégrée analyse diagnostique / analyse de risques

Directeurs de thèse :

M. Jean-Marie FLAUS	(LAG - UJF)
M. Stéphane PLOIX	(LAG - INPG)

JURY :

Mme Suzanne LESECQ	Président
Mme Louise TRAVE-MASSUYES	Rapporteur
M. Yves DUTUIT	Rapporteur
M. Jean-Pierre BOVEE	Examinateur
M. Jean-Marie FLAUS	Directeur de thèse
M. Stéphane PLOIX	Co-encadrant

à Audrey pour sa patience,

à mes parents, loin des yeux mais toujours dans mon coeur.

Si j'avais une heure de ma vie pour résoudre un problème dont ma vie dépende, je passerais :

- *40 mn à l'analyser*
- *15 mn pour en faire la revue critique*
- *5 mn pour le résoudre*

A. Einstein

Remerciements

Je souhaiterais remercier sur cette page tous ceux et celles qui ont contribué de près ou de loin à ces travaux de thèse, en particulier :

- Madame Louise TRAVE-MASSUYES et Monsieur Yves DUTUIT, rapporteurs de mes travaux de thèse, pour m'avoir fait l'honneur d'évaluer les résultats de mes recherches et d'avoir pris le temps de venir assister à ma présentation orale
- Monsieur Jean-Pierre BOVEE, examinateur, pour avoir apporter son point de vue industriel sur mes travaux
- Madame Suzanne LESCEQ d'avoir accepté d'évaluer mon travail et de participer à mon jury en tant que Président
- Monsieur Stéphane PLOIX, mon co-encadrant, et Monsieur Jean-Marie FLAUS, mon directeur de thèse, pour m'avoir supporté pendant 3 ans :) et pour m'avoir apporté énormément de choses sur le plan personnel et scientifique (la sagesse et la rigueur entre autres choses). J'ai eu un grand plaisir de travailler avec eux et je souhaite à tout doctorant qui prépare sa thèse d'avoir des encadrants tels qu'eux

Je voudrais également remercier ceux et celles que j'ai côtoyés au sein du laboratoire et qui travaillent "dans l'ombre", mais qui, je pense, participent également indirectement à la réussite des thèses par leur sympathie et enthousiasme : je pense à Patricia, Marie-Rose, Marie-Thèrèse, Virginie, Marielle, et tous les autres.

Enfin, je remercie tous mes amis et amies au sein du laboratoire, avec qui j'ai partagé mes moments de bonheur et aussi mes moments difficiles. Sans amitié solide et sans une bonne ambiance, je pense que le travail est voué à l'échec. C'est donc aussi grâce à vous mes amis que mes travaux de recherches se sont bien déroulés : je pense à Cindy, David, Laurent, Mimine, Manu, John, Abed, Kyarash, Julien, Marc, Samir, Olivier, Jean-Luc, Alexis, Eric, tout ceux dont j'ai oublié le prénom et qui se reconnaîtront.

Encore à tous : merci.

Table des matières

Table des figures

Liste des tableaux

Introduction

Italie, 1er Janvier 1999.
Dépôt de liquides inflammables d'essences, de gazole et de GPL.
Sur le site, les eaux de purge issues des réservoirs de stockage sont collectées vers un bac de récupération (3000 m^3).
Ce jour là, une purge, opération effectuée manuellement par un opérateur, est en cours. Auparavant, le bac de récupération contient 680 m^3 avec quantité d'Hydrocarbure surnageante estimée à 20 m^3.
Un écoulement d'Hydrocarbure, apparaissant via les évents sur le toit flottant du réservoir, est récupéré normalement par le dispositif de drainage des eaux pluviales et est renvoyé en pied de bac où il forme une flaque. Un nuage de vapeurs d'Hydrocarbure se forme et dérive jusqu'à la route, située à $60m$ du réservoir. Une explosion est initiée, probablement par le passage de 2 camions. Il est suivi d'autres explosions, après quelques secondes. Le retour de flammes provoque l'incendie des flaques puis du bac de récupération et de dispositifs connexes.
Le "Plan d'Organisation Interne (POI)" est déclenché ainsi que l'arrêt d'urgence des installations et les dispositifs fixes de refroidissement sont activés. Les pompiers maîtrisent le sinistre 1h30 plus tard. La circulation est interrompue sur la route voisine. L'accident fait 2 blessés (les 2 chauffeurs des camions, remis au bout de 7 et 15 jours) et des dégâts matériels évalués à 500000 euros [BARPI, n.d.].
En Europe, la directive SEVESO de 1982 puis la directive SEVESO II de 1996, intégrée en France via les directives sur les ICPE (Installations Classées pour la Protection de l'Environnement) oblige les entreprises "à risques" à évaluer leurs risques de manière qualitative. Par suite, le Préfet de la région où la société est implantée, suivant l'avis de la DRIRE (Direction Régionale de l'Industrie, de la Recherche et de l'Environnement) qui a étudié le dossier, donne l'autorisation d'exploiter à la société ainsi qu'une liste d'arrêtés préfectoraux à respecter.
Plus tard, un décret publié le 5 Novembre 2001, impose à toutes entreprises, de toutes natures et de tous domaines, de réaliser une évaluation des risques à matérialiser à travers un "document unique". Cette évaluation doit conduire à proposer et réaliser des améliorations et doit être mise à jour chaque année.
Dernièrement, la loi Bachelot demande aux entreprises soumises à la réglementation SEVESO de prendre en compte la cinétique et la probabilité des scénarii de dangers.
Enfin, il ne faut pas oublier non plus les lois sur l'environnement (loi sur l'air, l'eau, les déchets, les rejets, etc...) que doivent respecter les entreprises françaises.
Depuis la catastrophe de Seveso (Italie) en 1976 (d'où tire son nom la directive européenne),

l'Europe s'est donné les moyens de faire face à des accidents technologiques majeurs. Malheureusement, tous les accidents technologiques qui se seront produits ces dernières décennies montrent que le monde n'a pas encore pris totalement conscience des dangers de certaines exploitations et que trop souvent la sécurité est reléguée au second plan.

L'origine de l'accident en Italie est due à une probable corrosion du serpentin de réchauffage interne du réservoir qui a provoqué une fuite de vapeur vive dans le réservoir : la température à l'intérieur du bac de récupération atteint 60°C, la montée de pression provoque l'ouverture des évents. Après analyse, il s'avère que la quantité de produit contenue dans le bac était plus élevée que celle prévue. L'opérateur ne disposait pas d'indicateur de niveau, ni d'autre instrumentation qui aurait permis de détecter l'anomalie. Le réservoir avait été modifié pour ajouter le serpentin de réchauffage sans intégration de la surveillance de ce dispositif dans les procédures de maintenance.
Ainsi, si la présence de ce produit dans le bac de récupération ou si la température anormale dans le bac avait pu être détectées (via des capteurs par exemple), une analyse et une évaluation des risques préliminaires de l'installation aurait permis de prévoir une possible corrosion de la paroi et donc d'une fuite de vapeur. Les responsables de l'usine auraient pu prendre toutes les dispositions pour éviter ce désastre. Avec les outils et les méthodes disponibles aujourd'hui pour évaluer les risques, une analyse de l'installation aurait permis de mettre en évidence les points névralgiques du système et du procédé.

On voit donc bien qu'une approche combinée diagnostic/analyse des risques en ligne donnerait les moyens de suivre en temps réel les défaillances et les fautes (dans le cas de l'exemple précédent l'excès de température dans le bac ou la présence d'Hydrocarbure en quantité trop importante) et les risques potentiels associés (ici la fuite de vapeur). Mieux encore, en fonction des risques potentiels établis, il serait possible de lister les composants (ou les fonctions) à surveiller pour éliminer les risques. C'est une telle approche qui est proposée dans ce mémoire.
Dans un premier temps, un parallèle entre les outils d'analyse de risques et les méthodes de diagnostic sera établi pour mettre en avant leurs points communs et leurs différences.
Dans un second temps, nous proposons de définir des outils de modélisation structurelle, comportementale et fonctionnelle pour établir une base commune pour l'analyse des risques et le diagnostic. Cette base permettra l'utilisation combinée des méthodes et l'échange des données entre l'analyse des risques et le diagnostic.
Dans un troisième temps nous développerons un outil de diagnostic dont les résultats permettront de faire le lien avec l'analyse des risques. D'un autre côté un outil de pronostic de défaillances et de défauts en ligne sera proposé, outil réalisé à partir des méthodes actuelles d'analyse des risques et complété pour permettre de tenir compte de la probabilité d'occurrence des défauts et de leur délai d'apparition. Cet outil permettra également, en fonction d'un danger, de déterminer les points névralgiques à surveiller dans une installation pour limiter, voire éliminer ce danger. Enfin, nous verrons comment la modélisation comportementale proposée précédemment permet d'avoir un diagnostic plus fin que les méthodes de diagnostic usuelles et comment la modélisation pour l'analyse des risques peut s'intégrer dans le raisonnement diagnostic pour lui apporter de l'information supplémentaire pour également l'affiner.

Dans un dernier temps, nous appliquerons nos résultats à un système de refroidissement d'un réacteur siège d'une réaction exothermique.
En conclusion, nous établirons un bilan de l'approche proposée en dressant ses points forts et ses points faibles et nous donnerons ses perspectives en matière de recherche et d'applications.

Chapitre 1

État de l'art

L'une des plus grandes craintes des industriels, en termes de rendement, est que leurs installations tombent en panne ou qu'un événement indésirable les mette en péril. Pour faire face à ces éventualités, deux types d'analyses peuvent être utilisées : *l'analyse diagnostique* et *l'analyse des risques.*

En montrant et en s'appuyant sur cette complémentarité entre ces analyses, cette thèse propose, d'une part, une approche intégrant les résultats de l'analyse des risques aux procédures de diagnostic, et, d'autre part, une méthode permettant de pronostiquer des défauts, des défaillances ou de manière plus générale des dangers d'une installation.
Ce chapitre introduit donc un ensemble de méthodes de diagnostic existantes, ainsi que des méthodes d'analyse des risques en mettant l'accent sur les points communs (existants et à concevoir) des modèles pour une collaboration entre ces méthodes. En fin de chapitre, des méthodes de modélisation de danger seront présentées de manière à évaluer comment un modèle commun à l'analyse des risques et au diagnostic peut être envisagé.

Mais qu'est-ce qu'un danger ? Un risque ? Et pourquoi chercher à les déterminer, les évaluer ? Ces questions légitimes définissent le contexte de l'analyse des risques et trouveront des éléments de réponses dans cette première partie.

1.1 Contexte de l'analyse des risques

1.1.1 Notions de risques et notions connexes

Selon la norme OHSAS 18001 expliquée dans [Gey & Courdeau, 2005] :
- Un ***danger*** est une source ou une situation pouvant nuire par blessure ou atteinte à la santé, dommage à la propriété et à l'environnement du lieu de travail ou une combinaison de ces éléments
- un ***risque*** est la combinaison de la probabilité et de la (des) conséquence(s) de la survenue

d'un ***événement dangereux*** spécifié.

Pour compléter ces définitions, on peut également rajouter que :

– Un ***événement dangereux*** est un événement susceptible de causer un ***dommage***. On distingue :
  - *Les événements dangereux "accidentels"* : l'accident ou le presque accident.
    → Un accident est un événement "inattendu et soudain" qui entraîne un dommage corporel et ou matériel.
    → Un presque accident est un événement "inattendu et soudain" mais qui n'entraîne aucun dommage ; ce sont des événements dont on dit "J'ai peur ! Il s'en est fallu de peu ! Il l'a échappé belle ! etc. . . " La notion de presque accident est encore peu connue mais elle est extrêmement utile à exploiter en prévention
  - Les *événements dangereux "pouvant entraîner une atteinte à la santé"* : Ce sont des événements qui n'ont "pas de caractère de soudaineté" et qui se déroulent donc dans le temps. Le dommage éventuel apparaîtra lui aussi dans le temps au rythme de l'événement dangereux. Une atteinte à la santé peut se traduire par un malaise passager, un trouble de santé durable, une maladie professionnelle (reconnue ou non).
– Un ***dommage*** est une lésion physique et/ou une atteinte à la santé ou aux biens. Un dommage est la conséquence éventuelle d'un événement dangereux. On peut parler aussi de "dommage corporel" ou de dommage "matériel".

Ces mêmes notions, orientées plutôt vers les risques pour l'homme peuvent être consultées dans [AFNOR, 1991] et [AFNOR, 1997].

De manière plus formelle, un risque peut être mesuré par sa criticité, qui est fonction de sa probabilité et de sa gravité :

$$c = p \times g$$

Le critère de Farmer [Farmer, 1967] permet alors de définir les notions de risques acceptables et inacceptables (Fig 1.1).

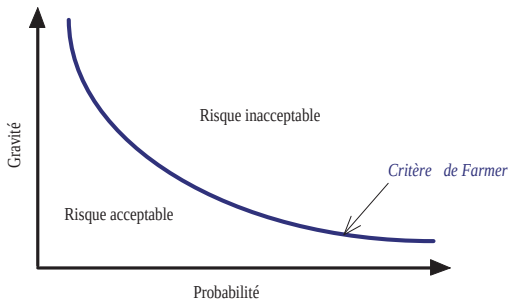


Fig. 1.1 – Critère de Farmer

Ces mêmes notions peuvent être représentées de manière qualitative par le tableau suivant, issu de [Lievens, 1976], qui définit des objectifs de sécurité dans l'aviation (Fig 1.2)

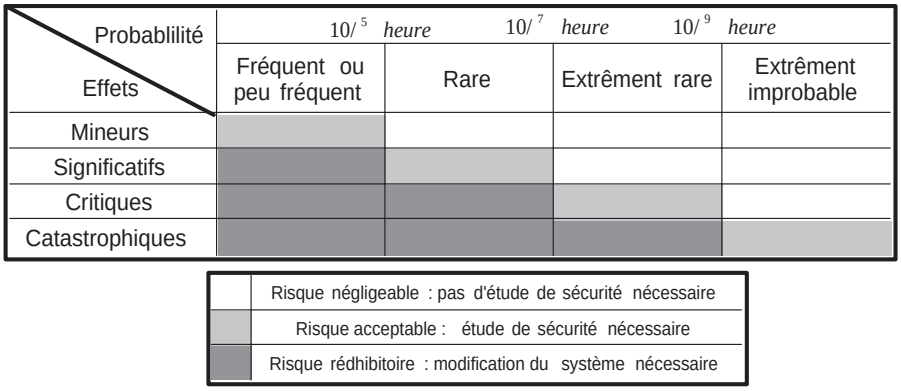


FIG. 1.2 – Criticité qualitative

1.1.2 Origine et finalité de l'analyse des risques

Bien que l'analyse des risques pourrait être une fin en soi, elle se veut d'abord l'instrument de la prévention et de la protection, tant d'un point de vue législatif que volontaire.

D'un point de vue juridique

Les accidents technologiques majeurs de ce siècle et du siècle dernier, comme l'accident de Seveso (Italie) en 1976, celui de Bhopal (Inde) en 1984, celui de Tchernobyl (1986) ou encore celui d'AZF (France) en 2001, ont fait prendre conscience aux autorités de la nécessité de définir des lois pour protéger les opérateurs d'une installation, les habitants et habitations environnants ainsi que l'environnement (faune et flore) de tous dangers et pollutions. C'est ainsi que sont nées différentes lois européennes et françaises (directives Seveso, loi sur l'air, loi sur l'eau...).

Cet objectif de maîtrise des risques est aussi actuellement le principal thème du projet européen ARAMIS [*ARAMIS : Accidental Risk Assessment Methodology for IndustrieS in th context of the Seveso II directive*, 2004]. La finalité de ce projet est de répondre aux besoins industriels en matière d'identification, d'évaluation et de réduction des risques. Pour cela, ce projet propose différentes méthodes permettant :

- l'identification de risques majeurs
- l'identification des barrières de sécurité et l'évaluation de leurs performances
- l'évaluation de l'efficacité de la gestion de la sécurité
- l'identification de scénarii d'accidents de référence
- l'évaluation et l'inventaire de la gravité des risques des scénarii de référence
- l'évaluation et l'inventaire de la vulnérabilité de l'environnement des installations

Une démarche volontaire

Outre le respect de la réglementation, la gestion du risque pour une entreprise est un gage de qualité. Après les normes ISO 9001 [ClubDesCertifiésDuMFQ, 1994] [Boéri, 2003] pour le management de la qualité, ISO 14001 [Baracchini & Thalmann, 2005] pour le management de l'environnement, la tendance actuelle des grands groupes est de mettre en place un système de management de la sécurité dont le plus connu est l'OHSAS 18001 [Gey & Courdeau, 2005]. Certaines sociétés vont plus loin encore en fusionnant tous ces systèmes de management pour en faire un système de management QSE intégré (Qualité/Sécurité/ Environnement)

L'un des principaux objectifs de cette thèse est de faire le lien entre les méthodes d'analyse de risque et le diagnostic. Pour cela, nous allons détailler dans cette seconde partie les méthodes les plus usuelles et les plus pertinentes en matière d'analyse de risques pour mettre en avant leurs besoins et leurs finalités.

1.2 Les différents types de méthode d'analyse de risques

Pour pronostiquer les défaillances et les défauts futurs, nous allons nous appuyer sur une analyse des risques du système. Une telle analyse permet de mettre en évidence les dangers d'une installation et bien souvent leurs causes et leurs conséquences, ces conséquences pouvant servir de base à un éventuel "pronostic".

De manière générale, les méthodes d'analyse de risque comportent au moins une des phases suivantes [Tixier et al., 2002] :

- une phase d'*identification* : cette phase essentielle consiste à décrire l'installation à analyser en termes d'activités dangereuses, de produits et d'équipement. Plus cette phase sera détaillée, plus l'analyse des risques sera exhaustive
- une phase d'*évaluation* des risques établis dans la phase précédente, réalisée dans le but de quantifier ces risques. Cette phase peut se faire deux manières complémentaires :
  - en évaluant les dommages conséquents à ces risques (approche déterministe)
  - en évaluant la probabilité d'occurrence du risque (approche probabiliste)
- une phase de *hiérarchisation* suite à l'évaluation des risques de manière à établir un ordre de priorité quant au traitement de ces risques

Dans cette partie seront détaillées les méthodes les plus usuelles et les plus pertinentes en matière d'analyse de risques pour mettre en avant leurs besoins et leurs finalités. De nombreuses méthodes d'analyse des risques sont disponibles dans la littérature, méthodes que l'on peut classer dans deux grandes familles : les méthodes inductives et déductives.

1.2.1 Les méthodes dites déductives

Une méthode est dite déductive si elle cherche les causes d'un événement non souhaité (noté ENS). Les plus connues et les plus utilisées sont les méthodes dites de l'***arbre des causes*** et de l'***arbre des défaillances***.

L'arbre des défaillances

L'arbre des défaillances est une méthode qui part d'un événement final (ENS) pour remonter vers les causes et conditions dont les combinaisons peuvent le provoquer [Mortureux, 2002]. Il vise à représenter l'ensemble des combinaisons qui peuvent induire l'événement étudié d'où sa représentation schématique. On construit et on utilise un arbre de défaillance dans le cadre d'une étude a priori d'un système. Ayant pour point de départ un ENS (dysfonctionnement ou accident), la démarche consiste à s'appuyer sur la connaissance des éléments constitutifs du système étudié pour identifier tous les scénarios conduisant à cet ENS. Graphiquement (Figure 1.3), un arbre de défaillance se représente en deux dimensions matérialisant les enchaînements qui peuvent conduire à l'ENS. Cette représentation peut également être utilisée pour calculer la probabilité de l'événement redouté à partir des probabilités des événements élémentaires qui se combinent pour le provoquer.

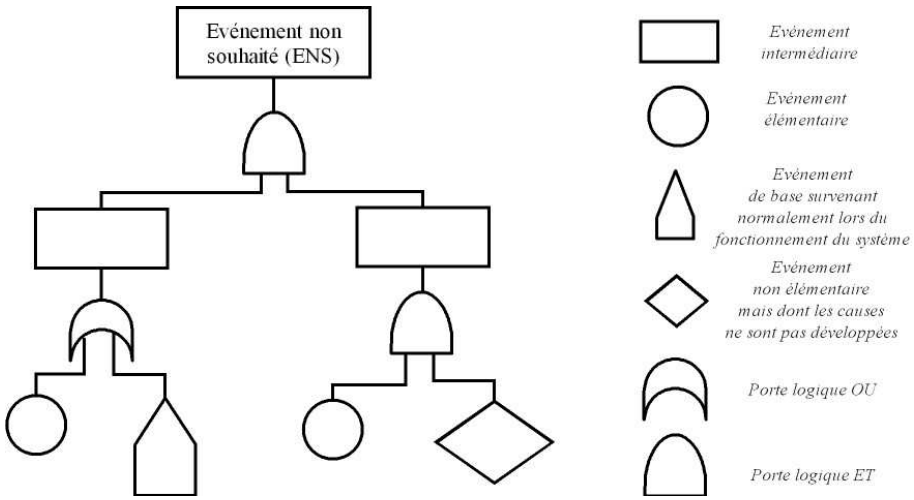


FIG. 1.3 – Représentation graphique d'un arbre de défaillances

Les combinaisons d'événements sont habituellement représentées par des portes logiques "ET", "OU", "ET" avec conditions, "OU" avec conditions, "SI" et des portes du type "ET m/n" (porte franchissable si m événements parmi n sont réalisés) [Limnios, 2005]. Par ailleurs, certains auteurs [Lievens, 1976] [Pages & Gondran, 1980] mentionnent d'autres portes logiques utilisées dans des applications particulières : porte "MATRICIELLE", porte "SOMMATION", etc...

L'arbre des causes

L'arbre des causes part d'un événement qui s'est produit et organise l'ensemble des événements ou conditions qui se sont combinés pour le produire [Mortureux, 2002]. Il repose sur un raisonnement dans le même sens que l'arbre des défaillances mais ne décrit qu'un seul scénario. Il se représente également en deux dimensions (Figure 1.4)

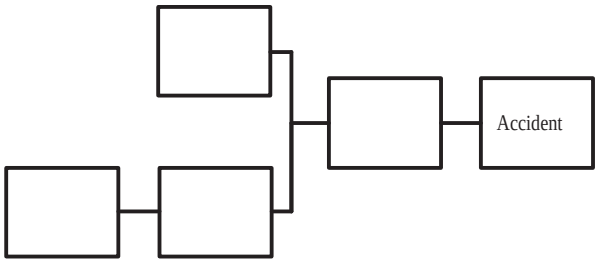


FIG. 1.4 – Représentation graphique d'un arbre des causes

On construit un arbre des causes en général dans une démarche de retour d'expérience. Lors de la recherche des causes, on peut s'appuyer sur l'arbre d'Ishikawa [Ishikawa, 1996] qui donne une méthode pour détailler tous les facteurs possibles liés à un événement indésirable (Figure 1.5).

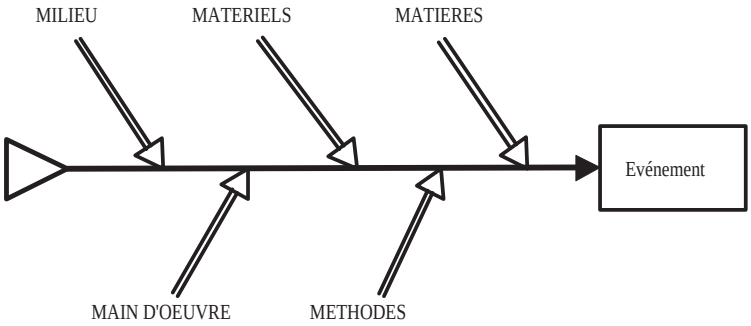


FIG. 1.5 – Arbre d'Ishikawa

Quel arbre choisir ?

L'arbre des défaillances a pour objectif de mettre en exergue les points vulnérables d'une installation dont les défaillances pourraient être à l'origine d'un événement indésirable. Dans ce type d'analyse, on cotera chaque événement de l'arbre par des probabilités de manière à rechercher les **coupes minimales** [Tang & Dugan, 2004] [Schneeweiss, 1999]. D'un autre côté, l'arbre des causes cherche à analyser une situation (un accident par exemple) **a posteriori** a partir d'événements passés provenant d'analyses "post-accident". L'arbre des causes cherche à expliquer l'événement indésirable qui s'est produit alors que l'arbre des défaillances cherche à déterminer les points faibles pour éviter un événement indésirable particulier.

1.2.2 Les méthodes dites inductives

A l'opposé des méthodes déductives, une méthode est dite ***inductive*** si elle cherche les conséquences d'un événement non souhaité (noté ENS). Les méthodes les plus utilisées et que nous présentons ici sont l'HAZOP [Lawley, 1974] et l'AMDEC [AFNOR, 1986][MIL-STD1629-A, 1983].

L'analyse des modes de défaillances et de leurs effets AMDE

L'analyse des modes de défaillances et de leurs effets (AMDE) (FMEA en anglais : Failure Mode and Effect Analysis) [MIL-STD1629-A, 1983] est une technique issue du domaine de la fiabilité. Elle repose sur la notion de mode de défaillance, définie comme l'effet par lequel une défaillance est observée sur un élément du système. Les défaillances d'un composant ont des effets sur les fonctions de ce dernier ; le mode de défaillance est dénommé du nom de l'effet.
L'AMDE est une approche inductive : à partir d'un mode de défaillance d'une entité, on détermine les défaillances/dégradations provoquées par ce mode défaillance.
L'AMDEC, ("C" pour criticité), est une extension naturelle de l'AMDE [Jordan & Marshall, 1972]. Elle considère la probabilité d'occurrence de tous les modes de défaillance et la gravité des effets pour hiérarchiser les criticités.

La méthode AMDEC se divise en quatre étapes [Zwinglestein, 1999] :

1. *Définition du système, de ses fonctions et de ses composants :*
   Il s'agit de définir les fonctions principales du système ainsi que ses spécifications
2. *Recensement des modes de défaillances et détermination de leurs causes pour chacun des composants du système :*
   C'est l'étape la plus délicate car elle doit être la plus complète possible. Pour déterminer les modes de défaillance, on peut s'appuyer sur des modes de défaillance génériques [Zwingelstein, 1995][Flaus, 2004] (Figure 1.6)
   On peut encore s'appuyer sur les modes de défaillances génériques prévues par la norme [AFNOR, 1986] (Figure 1.7) :
3. *Etude des effets des modes de défaillances :*
   Pour étudier les effets, on ne considère qu'un seul mode de défaillance pour le composant et on considère également que tous les autres composants fonctionnent correctement.

   On peut classer les effets en deux catégories [Villemeur, 1988] :
   - les effets sur le système, i.e. les effets des modes de défaillance sur les composants, le système et leurs fonctions
   - les effets sur les systèmes externes, i.e. les effets des modes de défaillance sur les systèmes (et leurs composants) qui sont en étroite interaction avec le système étudié
4. *Etude de la criticité :*
   À cette dernière étape on calcule les criticités pour ainsi hiérarchiser les modes de défaillance pour faire ressortir les plus critiques. La mise en place de moyens de préventions

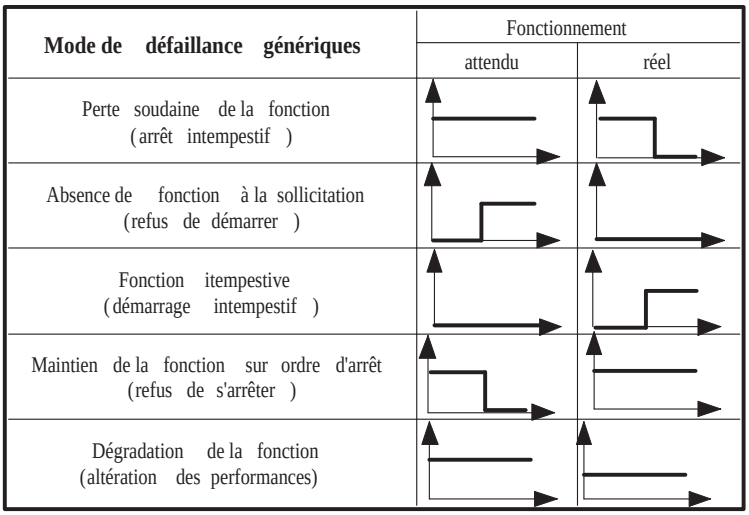


FIG. 1.6 – Modes de défaillances génériques

et de protections peut alors être décidée pour diminuer la criticité des points faibles de l'installation.

Le résultat et le contenu de l'analyse AMDEC se représentent habituellement sous la forme d'un tableau (Tableau 1.1). Elle permet de lister par ordre d'importance les sources potentielles de défaillance, susceptibles d'entraîner un scénario critique. Cette analyse, pouvant être préliminaire à d'autres, permet d'éviter des modifications de systèmes onéreuses en détectant au plus tôt les faiblesses du système.

TAB. 1.1 – Résultats de l'AMDEC

Composant	Fonctions	Mode de défaillance	Causes	Effets	Mesures préventives	Probabilité	Gravité	Criticité

L'HAZOP (HAZard and OPerability Study)

L'HAZOP (HAZard and OPerability Study) a été initialement développée par la société "Imperial Chemical Industries" au début des années 1970 et s'applique à l'industrie chimique [Lawley, 1974]. La méthode est de même type que l'AMDE mais est mieux adaptée pour l'analyse des circuits thermohydrauliques. Elle a pour objectif l'identification des risques et l'étude de leur prévention/protection en s'appuyant sur un modèle **entité/flux physique** du système, un flux étant matérialisé par une variable, et en étudiant chacune des déviations possibles. Elle consiste donc à remplir un tableau standard contenant préalablement un certain nombre de mots-clefs (Tableau 1.2). Ceux-ci permettent de passer en revue les déviations des paramètres importants (Tableau 1.3) en mettant en évidence les causes et les conséquences de leurs dé-

Modes génériques de défaillance	
1. Défaillance structurelle	18. Mise en marche erronée
2. Blocage physique ou coincement	19. Ne s'arrête pas
3. Vibrations	20. Ne démarre pas
4. Ne reste pas en position	21. Ne commute pas
5. Ne s'ouvre pas	22. Fonctionnement prématuré
6. Ne se ferme pas	23. Fonctionnement après le délai prévu (retard)
7. Défaillance en position ouverte	24. Entrée erronée (augmentation)
8. Défaillance en position fermée	25. Entrée erronée (diminution)
9. Fuite interne	26. Sortie erronée (augmentation)
10. Fuite externe	27. Sortie erronée (diminution)
11. Dépasse la limite supérieure tolérée	28. Perte de l'entrée
12. Est au- dessous de la limite inférieure tolérée	29. Perte de la sortie
13. Fonctionnement intempestif	30. Court-circuit (électrique)
14. Fonctionnement intermittent	31. Circuit ouvert (électrique)
15. Fonctionnement irrégulier	32. Fuite (électrique)
16. Indication erronée	33. Autres conditions de défaillance exceptionnelles suivant les caractéristiques du système , les conditions de fonctionnement et les contraintes opérationnelles
17. Écoulement réduit	

FIG. 1.7 – Modes de défaillances génériques selon l'AFNOR

viations éventuelles, ainsi que les moyens de détection et les actions correctrices possibles. Une hiérarchisation d'après la fréquence et la gravité des déviations possibles est alors effectuée.

TAB. 1.2 – Liste des mots-clés pour l'analyse HAZOP

Mots-clés	*Significations*
PAS	Variable absente
TROP	Excès de la variable
PAS ASSEZ	Trop peu de la variable
INVERSE DE	Inversion logique
AUTRE	Complète substitution
PLUS DE	Sur-ensemble
PARTIE DE	Sous-ensemble

TAB. 1.3 – Liste de paramètres standards

Liste de paramètres standards	
Température	Fréquence
Débit	Viscosité
Pression	Tension
Niveau	Information
Temps	Agitation
Composition	Ajout
pH	Separation
Vitesse	Réaction
Temps de séjour	Concentrations
Impuretés	Propriétés physiques
Confinement	Utilités

La démarche générale de la méthode HAZOP est décrite par l'organigramme en Figure 1.8 [Froquet, 2005].

Les résultats sont fournis en général sous forme de table (Tableau 1.4)

TAB. 1.4 – Résultats de l'HAZOP

N°	Déviation	Causes	Conséquences	Prévention Existante	Actions à prendre	Probabilité	Gravité

1.2.3 Les limites de ces méthodes

La méthode de l'arbre de défaillance trouve ses faiblesses dans la notion d' "événement", très vague, qui laisse un très grand champ de possibilités derrière cette terminologie. Il est donc facile, d'une personne à l'autre d'obtenir des arbres différents, donc difficilement comparables, et cela d'autant plus que le système à étudier est complexe. Par ailleurs, l'emploi de cette méthode se révèle difficile pour l'étude de systèmes en interactions et fortement dépendant du temps [Villemeur, 1988].

Bien que d'un emploi courant, **les limites de l'AMDE** sont relatives à l'impossibilité de traiter le cas de défaillances multiples. Une telle défaillance dite "de mode commun" est due à

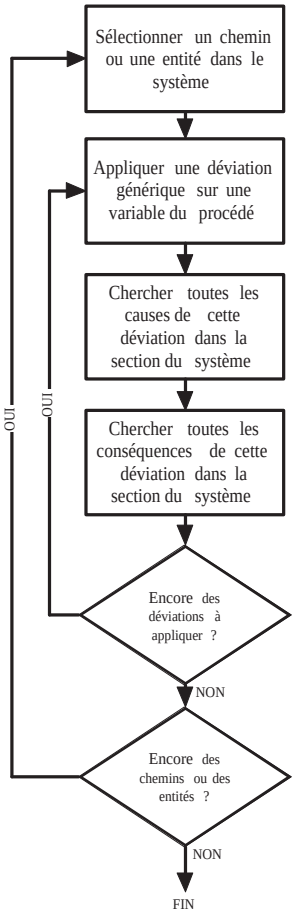


Fig. 1.8 – Organigramme d'analyse HAZOP

des pannes survenant simultanément dans différents composants du système, une même cause étant à l'origine de ces pannes.

Le point faible de l'HAZOP vient de son analyse qui repose sur la description par flux : on ne tient pas compte des défaillances des entités du système. D'autre part, les enchaînements de déviations ou les déviations simultanées ne sont également pas étudiées.

1.2.4 Conclusion

De manière générale, on remarque que quelques soient les méthodes d'analyse utilisées, elles sont basées sur un découpage soit structurel (pour les arbres de défaillance, les arbres de cause et l'HAZOP) soit fonctionnel du système (pour l'analyse AMDEC).
Afin d'intégrer les résultats de l'analyse des risques et de pronostiquer les défaillances futures, nos travaux vont s'appuyer en partie sur des relations de type cause/conséquences. Dans les prochains chapitres, nous allons privilégier une analyse de type AMDEC pour son aspect systémique et pour son analyse des défaillances du systèmes. D'autre part, pour tenir compte

des combinaisons possibles d'événements, nous allons intégrer les résultats de l'AMDEC dans un arbre de défaillance.
Cependant, bien que ces analyses de risques soient basées sur ces découpages, aucun formalisme n'est indiqué pour les modélisations structurelles et fonctionnelles. Ainsi, pour envisager un lien entre l'analyse des risques et le diagnostic, un formalisme commun devra être donc développé, tant structurellement que fonctionnellement.

Après cette présentation des principaux outils d'analyse de risques, la section suivante présente différentes méthodes de modélisation de dangers. Les notions abordées dans ces méthodes et leurs applications à des systèmes apportent de l'information que nous intégrerons aux outils de modélisations que nous proposerons dans la suite.

1.3 Modélisation systémique - Modèles de dangers

A travers les différents outils de modélisation décrits dans cette section, un certain nombre d'informations et de notions provenant de ces outils seront intégrées aux modèles que nous proposerons dans la suite de ces travaux.
Ces informations proviennent, d'une part, de l'analyse systémique, dont nous reprendrons un certain nombre de notions et principes, et, d'autre part, des modèles de danger MADS, MoDyF et Scenarisk.

1.3.1 Approche systémique

On peut définir un ***système*** comme étant un ensemble déterminé d'éléments discrets (ou composants) interconnectés ou en interaction [Veseley et al., 1981]. La modélisation structurelle d'un système est la modélisation la plus naturelle qui existe : il s'agit de "découper" le système en différentes entités en essayant de respecter une certaine granularité, quitte à introduire des découpages intermédiaires en "sous-systèmes".
Pour aller plus loin, pour modéliser un système, on peut aussi s'appuyer sur l'approche systémique. Cette approche est une méthodologie de représentation, de modélisation d'objets actifs (eux-mêmes ensembles d'entités actives en interaction dynamique) finalisés, physiques ou immatériels, en interaction avec leur environnement par l'intermédiaire de flux (énergétiques, informationnels ou matériels - de l'énergie informée-) sur lesquels le système exerce une action : flux qu'il modifie et "processe" [LeMoigne, 1994] [LeMoigne, 1990].

Modélisation structurelle

Structurellement, un système comprend quatre composants :
– **Les entités**, qui sont les parties constituantes : on peut en évaluer le nombre et la nature (même si ce n'est qu'approximativement). Ces éléments sont plus ou moins homogènes.

- **Une limite (ou frontière)** qui sépare la totalité des entités de l'environnement du système ; cette limite est toujours plus ou moins perméable et constitue une interface avec le milieu extérieur.
- **Des réseaux de relation** : les entités sont en effet interreliées. Les relations peuvent être de toutes sortes. Les deux principaux types de relations sont :
  - le transport
  - les communications

  En fait, ces deux types peuvent se réduire à un seul, puisque communiquer c'est transporter de l'information, et transporter sert à communiquer (faire circuler) des matériaux, de l'énergie ou de l'information.
- **Des stocks (ou réservoirs)** où sont entreposés les matériaux, l'énergie ou l'information, et qui doivent être transmis ou réceptionnés

Modélisation fonctionnelle/comportementale

Fonctionnellement, un système peut se décomposer sous forme :
- De **flux** de matériaux, d'énergie ou d'informations, qui empruntent les réseaux de relations et transitent par les stocks. Ils fonctionnent par entrées/sorties avec l'environnement.
- Des **centres de décision** qui organisent les réseaux de relations, c'est-à-dire coordonnent les flux et gèrent les stocks.
- Des **boucles de rétroaction** qui servent à informer, à l'entrée des flux, sur leur sortie, de façon à permettre aux centres de décision de connaître plus rapidement l'état général du système.

Dans la suite, 3 modèles de représentation de la notion de danger et de scenario de danger sont proposés :
- Le modèle MADS
- Le modèle MoDyF
- Le modèle Scenarisk, basée sur la modélisation MoDyF

1.3.2 Le modèle MADS

Le modèle MADS (Méthode d'Analyse des Dysfonctionnements des Systèmes), est un outil qui permet de modéliser la notion de danger. Il est construit sur les bases des principes de la modélisation systémique développée par Jean-Louis Le Moigne [LeMoigne, 1994].
La modélisation systémique repose sur le principe de *processus* qui peut être défini comme l'ensemble ordonné des changements qui affectent :
- la position dans le temps
- l'espace
- la forme
- la nature

Un tel processus est décrit par :

– les relations entre les objets subissant le changement et ceux produisant le changement
– la liste des changements subits par les objets processés

Les relations des objets entre eux dans le cadre d'un processus et les relations avec l'environnement sont représentées :
– par des *flux* (ou transactions) d'information, d'énergie ou de matière
– par une capacité d'influence de l'environnement sur le système, appelée *Champ*

Dans ce contexte, le modèle MADS représente la notion de danger sous la forme de flux de danger. Le modèle est composé de deux systèmes appelés système source de danger et système cible en interaction dans un environnement actif. Le champ de danger est l'environnement susceptible d'influencer les systèmes sources et cibles du flux de danger (figure 1.9). L'idée principale de cette méthode est donc de décomposer un système à analyser et ensuite à recenser les flux de danger entre les différents sous-systèmes. Ces flux permettent de déterminer de manière exhaustive les scénarios de danger pour mettre en place par la suite des moyens de protections/préventions.

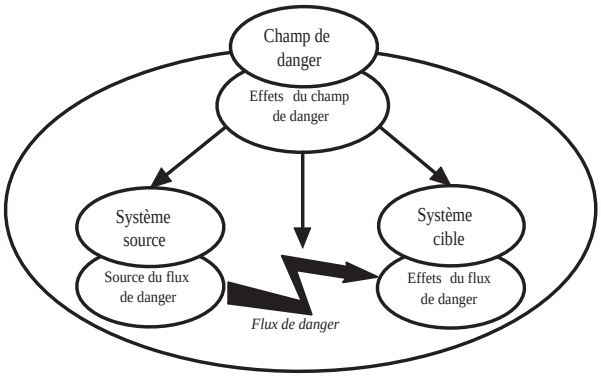


FIG. 1.9 – Modèle MADS

1.3.3 Le modèle MoDyF

Un autre modèle est le modèle MoDyF (Modèle de Dysfonctionnement Formel) [Flaus, 2003] [Flaus & Granddamas, 2002] qui apparaît comme un complément au modèle MADS. En effet, il s'intéresse à la description d'un réseau d'entités en relation de dysfonctionnement (figure 1.10) :
– l'état de chaque entité est décrit par un ensemble de variables caractérisant la situation dans laquelle se trouve l'entité. A chaque jeu de valeurs caractérisant l'état physique est associé un état de fonctionnement, qui prend ses valeurs dans {*état normal, état anormal non dangereux, état anormal dangereux*}
– les dysfonctionnements sont propagés par des relations de cause à effet entre les entités

Ce modèle s'appuie sur une représentation discrète : les relations de dysfonctionnement sont de type événementiel.

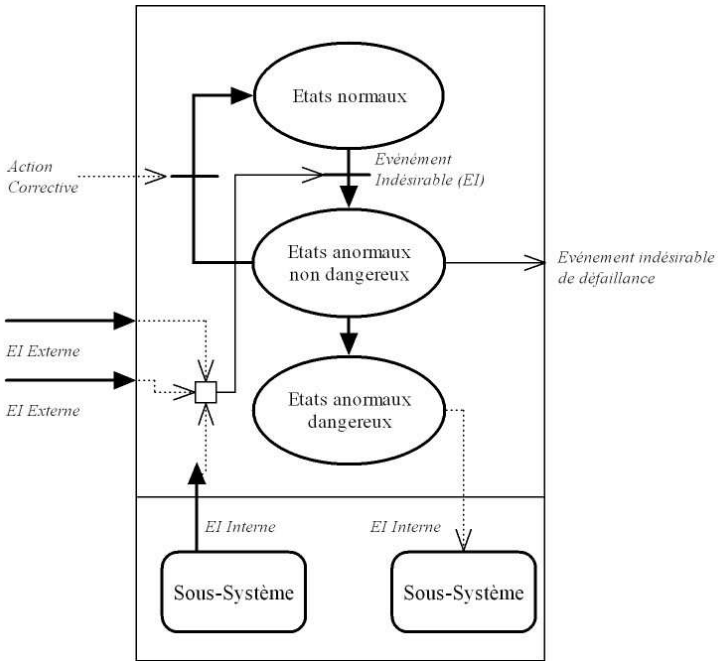


FIG. 1.10 – Modèle MoDyF Elémentaire

Le système décrit est placé dans un environnement avec lequel il interagit. Les effets des dysfonctionnements peuvent alors se manifester en interne ou vers l'extérieur. De la même façon, les conditions qui peuvent entraîner un dysfonctionnement peuvent être liées à des faits internes ou aux interactions avec l'extérieur.

Pour réaliser le modèle de dysfonctionnement de manière systématique, [Flaus, 2001] propose de s'appuyer sur un modèle structurel et fonctionnel repris et complété dans [Flaus et al., 2006] appelé **FISE** qui comporte quatre aspects de description :

- ***F**onctions* : les fonctions permettent de décrire la liste des fonctions que peut assurer l'entité
- ***I**nteractions* : les interactions, comme dans le modèle MADS, décrivent sous forme de flux de matière, d'énergie et d'information les relations entre les entités et avec l'environnement
- ***S**tructure* : les informations de structure définissent la structure physique de l'entité, et précisent la frontière physique qui délimite le système considéré. Elles permettent aussi de spécifier si cette entité est un objet contenant ou contenu
- ***E**tat* interne : c'est un ensemble de grandeurs permettant de caractériser la situation dans laquelle l'entité se trouve

Ces aspects sont à la fois décrits dans le cadre d'un bon fonctionnement et d'un mauvais fonctionnement. Ensuite, sous certaines hypothèses, le graphe causal de dysfonctionnement est construit à partir du modèle physique du système, en considérant que tout dysfonctionnement se propage via une interaction physique anormale.
Cette modélisation est étendue à chaque entité du système de manière à pouvoir faire évoluer

les dysfonctionnements dans le système complet.

1.3.4 La modélisation par scénarii : la méthode ScénaRisK

La méthode ScénaRisK [Froquet, 2005] a pour objectif de générer des scénarios à partir d'une librairie d'éléments de scénarios existante et d'une représentation spécification de l'installation. Le modèle de danger sur lequel elle se base est représenté sous la forme d'un automate à états et s'appuyant sur le modèle MoDyF décrit précédemment [Flaus, 2003]. Le principe, comme tout modèle de danger, est de représenter le fonctionnement de l'apparition et l'évolution pouvant conduire à l'apparition d'un événement non souhaité.
Le modèle de danger proposé repose sur deux principes (figure 1.11) :

- Le premier principe est la notion d'*état de pré-danger* et d'*état de danger* d'une entité. L'état de danger se réalise et fait suite à l'état de pré-danger sous condition de réalisation d'un événement
- L'autre idée est que chaque entité dans un état de danger est susceptible de générer un événement dangereux

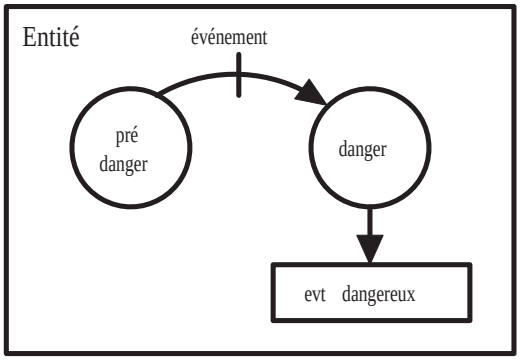


Fig. 1.11 – Modèle ScénaRisk

On voit alors très bien les enchaînements possibles, et donc la génération de scénario, du fait que chaque état de danger générant un événement dangereux peut être à l'origine d'une ou plusieurs transitions d'un état de pré-danger d'une entité vers un état de danger (figure 1.12).

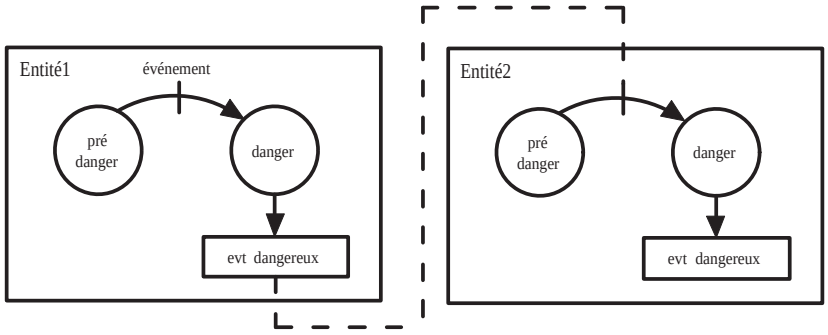


FIG. 1.12 – Modèle ScénaRisk avec enchaînement

1.3.5 Conclusion

Les notions qui ressortent de ces modèles et qui apparaissent comme pertinentes à être intégrées dans nos modèles sont les suivantes :

- l'aspect systémique ainsi que la modélisation FISE qui nous serviront de base pour développer un formalisme pour la modélisation fonctionnelle et structurelle que nous proposerons dans la suite
- la notion de flux, qui permettra de prendre en compte les relations de cause à effet entre composants et d'intégrer cette relation dans l'analyse diagnostique

Ainsi, nous avons en notre possession une base pour proposer un formalisme pour une modélisation fonctionnelle et structurelle en vue d'une analyse AMDEC tout en intégrant l'aspect relation cause/effet entre composant (aspect non pris en compte dans l'AMDEC qui recherche uniquement les causes de modes de défaillance et non de défaut).
Reste désormais à définir l'analyse diagnostique qui pourra intégrer les résultats de l'analyse AMDEC. Pour cela, la section suivante propose un état des lieux des outils de diagnostic.

1.4 L'analyse diagnostique

Nous savons désormais quels outils utiliser pour mettre en évidence les relations de type causes/conséquences dans une installation.
Notre but est, d'une part, d'intégrer ces relations dans les procédures de diagnostic et, d'autre part, d'utiliser ces "conséquences" pour établir notre pronostic de défaillances et de défauts futurs.
Pour répondre à nos besoins, il est nécessaire de fait appel à un moyen de déterminer les "causes" pour amorcer l'analyse des relations causes/conséquences. Pour être plus précis, nous avons besoin d'un outil de **détection**, autrement dit, nous avons besoin d'outils d'analyse diagnostique.

1.4.1 Introduction

Diagnostiquer un système pour en rechercher les éléments en défaut est une problématique qui a fait couler beaucoup d'encre. De nombreuses méthodes existent et s'appuient sur différents principes d'analyse. Dans [Price, 1999], les méthodes de diagnostic les plus usuelles sont présentées, des plus intuitives aux plus complexes.
Dans cette partie, les principaux outils d'analyse diagnostique vont être présentés. Cette partie répond à deux questions essentielles : de quels modèles de connaissance du système ces méthodes ont-elles besoin pour effectuer un diagnostic, et quelles sont ces méthodes de diagnostic.

On distingue différentes familles d'analyse diagnostique :
- L' analyse diagnostique basée sur les arbres de défauts
- L' analyse diagnostique basée sur l'analyse de cas similaires de défauts
- L'analyse diagnostique dite "sans modèles"
- L'analyse diagnostique dite "à base de modèles".

Bien entendu, les points cités ci-dessus sont également à base de modèles. Mais par abus de langage, derrière l'appellation "à base de modèles" on entend "modèles d'évolution du système" (ce que ne sont pas les deux premiers points).
Dans cette sous-famille, on peut distinguer
- Le raisonnement causal
- Le raisonnement FDI (Fault Detection and Isolation)
- Le raisonnement DX (diagnostic logique)

Le diagnostic à base de modèle repose sur deux types de modèles d'évolution du système : les modèles continus (équations différentielles) et les modèles qualitatifs. Dans la suite de nos travaux, nous proposons une modélisation comportementale qualitative des installations ; c'est pourquoi, nous présentons le principe et les outils les plus usuels de la modélisation qualitative.

La modélisation qualitative

a. Principe

Pour construire des représentations qui permettent une meilleure compréhension des phénomènes physiques mis en jeu dans un procédé, l'Intelligence Artificielle (IA) s'intéresse au milieu des années 1980 au **raisonnement qualitatif**. Le développement de techniques de représentation qualitative du monde pour en "comprendre schématiquement le fonctionnement" est une idée très naturelle et par conséquent très ancienne [Montmain, 2005].
La modélisation qualitative a pour objectif de faciliter la construction de modèles. Le moyen de simplification couramment utilisé est la diminution de la précision de la modélisation numérique, voire même une simplification extrême dans [DeKleer & Brown, 1984]. Les équations quantitatives décrivant classiquement le comportement d'un système sont alors transformées en

équations qualitatives ou confluences, qui sont des fonctions de variables qualitatives.
Le raisonnement qualitatif trouve, en majeure partie, ses origines dans les travaux de P.J. Hayès [Hayès, 1985] sur la physique naïve. L'objectif de ses travaux est de construire une modélisation de notre perception de sens commun du monde physique, et ainsi s'affranchir de la difficulté à concevoir des modèles mathématiques.
De manière générale le raisonnement qualitatif a pour vocation de combler l'insuffisance des méthodes numériques dans des domaines où les connaissances sont peu formalisées et/ou difficilement quantifiables [Travé-Massuyès & Dague, 2003]. Ce type de raisonnement est fondé sur une description comportementale de l'état des **variables** du modèle en considérant les **qualités** qui les caractérisent [DeKleer & Brown, 1983]. De manière pratique, les qualités sont des symboles ordonnés. Par exemple, les qualités associées à une variable représentant l'état du trafic autoroutier peuvent correspondre à trois états :

$$fluide, dense, bouchon$$

en les ordonnant ainsi

$$fluide < dense < bouchon$$

On peut donc raisonner sans connaître la véritable valeur du trafic autoroutier, qui pourrait être exprimée en $vehicules/m$ ou $vehicules/s$.

b. Les graphes causaux

Les graphes causaux sont basés sur des relations de cause à effet : l'expression de telles relations est un principe naturel du raisonnement. L'approche causale est une méthode de modélisation qui est largement utilisée dans les sciences. Les modèles causaux sont très répandus dans tous les formalismes du raisonnement qualitatif [Iwasaki & Simo, 1994] [Travé-Massuyès et al., 1993] [Kuipers, 1984] [DeKleer & Brown, 1983] [Dague & Travé-Massuyès, 2004].
Une définition générale de la causalité est de la considérer comme une relation spatio-temporelle qui lie la cause à son effet. Les modèles causaux ont donc l'avantage de représenter les relations et les interactions qui lient tous les éléments d'un modèle.

La structure la plus simple de graphe causal est celle du graphe signé orienté (SDG pour *Signed Directed Graph*). Les noeuds d'un SDG correspondent à l'état des variables, et les arcs orientés portent les signes des influences correspondantes :

- le signe est "+" lorsque les variables se rapportant à l'arc évoluent dans le même sens
- le signe est "-" lorsque les variables se rapportant à l'arc évoluent en sens opposé

L'état d'une variable appartient à l'ensemble $\{+, 0, -\}$, selon que la valeur est normale (0), trop haute (+) ou trop basse (-) (Figure 1.13).

Les graphes causaux apparaissent comme des outils adéquats pour la modélisation causale. Ils peuvent servir de support à des raisonnements variés, comme la simulation, le filtrage de défauts, la localisation de pannes... Comme les arcs d'un graphe symbolisent les relations causales

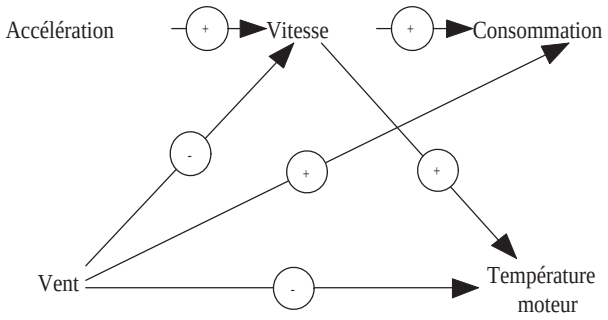


FIG. 1.13 – Exemple d'un SDG issu du comportement d'une voiture

entre les variables, le graphe est une représentation de la causalité des phénomènes physiques dans l'installation.

c. Graphes causaux et modèles dynamiques

Dans [Feray-Beaumont et al., 1989] et [Leyval, 1991], une modélisation particulière est proposée à partir de graphes causaux. Les noeuds représentent toujours les variables du système et les arcs les relations de cause à effets entre ces variables. La particularité vient du fait que chaque arc est en fait le support d'une Fonction de Transfert Qualitative (FTQ), traduisant l'influence de la variable amont de l'arc sur la variable aval en termes de gain statique, temps de réponse, de retard, etc. Dans [Leyval, 1991], cette modélisation est utilisée pour la simulation de procédé ; l'évolution des variables dans le temps est représentée par une fonction affine par morceaux et leurs évolutions sont propagées dans l'arbre grâce aux FTQ portées par les arcs. Les FTQ, comme leur nom le laisse supposer, s'apparentent aux fonctions de transferts (elles reprennent des notions tels que le gain, le temps de réponse, etc.) mais traitent des variables quantifiées et non des grandeurs numériques. Elles sont dites "qualitatives" parce qu'elles opèrent sur des domaines quantifiés associés aux variables (domaine de valeurs à l'origine continu) et qu'elles traitent également leurs signes. On distingue quatre types de FTQ [Feray-Beaumont, 1989] selon le type de comportement que l'on souhaite décrire :

- les ***FTQ m +/- g tr rp*** : ce sont des fonctions de transfert construites sur le modèle des réponses à un système du premier ordre, avec g représentant le gain et prenant ses valeurs dans l'ensemble $\{fort, moyen, faible\}$ et où le signe (+ ou -) indique si le sens de variation du gain, avec tr le temps de réponse et rp un retard numérique
- les ***FTQ id +/- tr rp*** : elles permettent de décrire la relation entre deux variables liées par une constante de proportionnalité (gain) et éventuellement un retard, le signe du gain étant défini par le signe + ou −, avec tr et rp ayant les mêmes définitions
- les ***FTQ reg rb tr rp*** : elles décrivent les régulations dues à une perturbation et possèdent les mêmes paramètres que la *FTQ m+* avec rb qualifiant le retard de boucle
- les ***FTQ comp-act*** : elles permettent d'intégrer le comportement d'un actionneur dans la boucle de régulation. Cette FTQ est définie comme la somme, épisode à épisode, des réponses d'une *FTQ m+* et d'une *FTQ reg*

La modélisation comportementale du système qui sera proposée dans la suite de ce mémoire reprendra l'idée de la représentation par noeuds et arcs où les noeuds représenteront les variables et les arcs les liens entre les variables mais sans lien de causalité.
Par contre, en ce qui concerne la modélisation fonctionnelle du système, une modélisation par graphe sera également proposée, mais elle n'aura aucun lien avec la notion de graphe causal défini ci-dessus puisque dans la suite, nos graphes fonctionnels comporteront des modes (et non des variables) et seront pourvus de combinaisons logiques (à l'instar des arbres de défaillance).

1.4.2 Le diagnostic à partir d'arbres de défaut

En utilisant les arbres de défauts détaillés dans la section 1.2.1, on peut créer des modèles pour diagnostiquer un système. Le principe est de construire un arbre à questions successives dont la réponse ne peut être que *OUI* ou *NON*. Selon la réponse à la question, la branche suivante de l'arbre atteint soit une autre question, soit un diagnostic. Ce principe de checklist successif ressemble fortement à l'aide interactive proposée par les services supports d'aide à distance pour résoudre un problème.
Par exemple la figure 1.14 montre les questions successives à se poser en cas de panne de sa voiture.

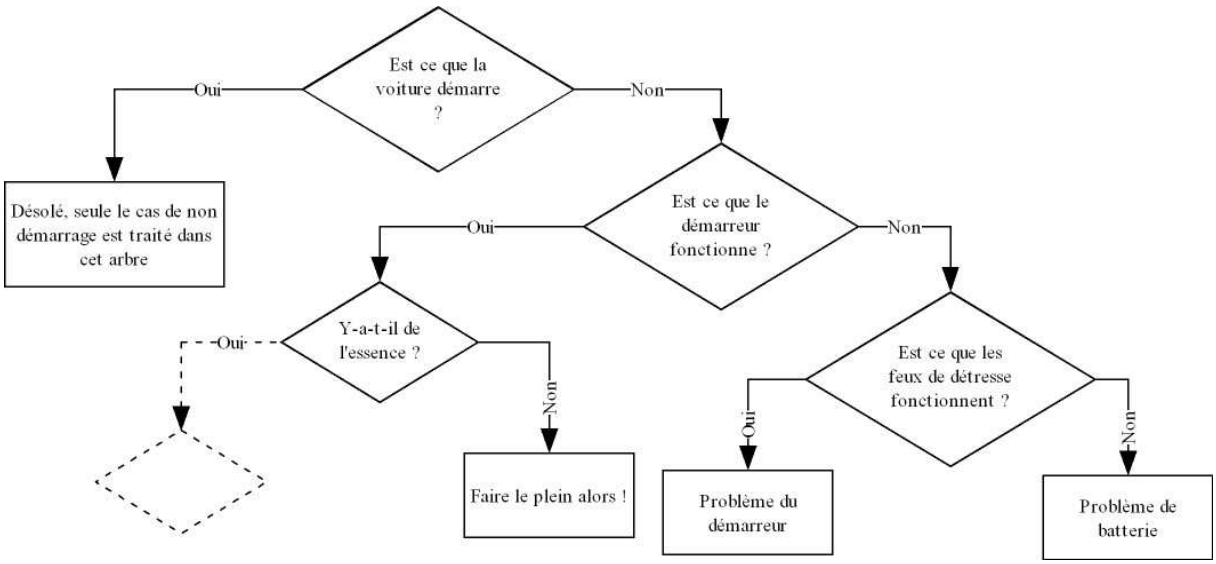


FIG. 1.14 – Système de diagnostic à base d'arbre de défaut

Ces arbres de décision pour le diagnostic présentent un certain nombre d'avantages :
- ils peuvent être facilement compris
- ils sont encore utilisés par les services de maintenance pour décrire des procédures de résolution de problèmes
- ils peuvent être utilisés pour représenter un large champ de problèmes de diagnostic

Par la suite, ce principe de diagnostic a été perfectionné, en permettant par exemple d'analyser des systèmes très complexes et en permettant une analyse croisée de plusieurs arbres de défauts pour résoudre un même problème.

1.4.3 Le diagnostic à base de cas particuliers

Une autre manière de diagnostiquer un système consiste à utiliser des solutions antérieures d'un problème pour résoudre des cas similaires [Kolodner, 1993]. Cette méthode convient à l'analyse de système dont les principes de fonctionnement ne sont pas très bien connus ou pour des problèmes qu'il est difficile de décomposer, mais où un large passé en matière de diagnostic est disponible.

Ce type de diagnostic se déroule de la façon suivante :

- Il faut tout d'abord choisir parmi les anciennes analyses diagnostiques dont les problèmes résolus ressemblent en beaucoup de points à celui actuellement étudié
- Ensuite on adapte la solution proposée pour l'ancien problème
- On en déduit une nouvelle solution, et on l'évalue
- Si la solution est bonne, on enregistre ce cas particulier dans la base de donnée des analyses diagnostiques pour peut-être le réutiliser un jour

Un exemple trivial d'application de cette méthode est la méthode de diagnostic des médecins généralistes : en fonction de symptômes différents que présente un patient, ils recherchent, en croisant ces symptômes dans leur base de données, la maladie ou plus généralement le phénomène expliquant ces symptômes.

De manière synthétique, le diagramme donné en figure 1.15 (page 45) représente le principe du diagnostic à base de connaissance de cas particuliers.

1.4.4 Le diagnostic "sans modèles"

Cette méthode ne sera pas particulièrement approfondie ici parce que nous souhaitons utiliser dans la suite des modèles comportementaux de type structurel pour permettre le lien avec les méthodes d'analyse de risques.
Néanmoins, parmi les méthodes de diagnostic sans modèles, on peut citer :

- les outils de traitement du signal reposant sur l'analyse fréquentielle (transformée de Fourier), utilisée à l'origine pour détecter les défauts dans les machines électriques [Thomson, 1999]. Avec le développement des applications à vitesse variable, les recherches actuelles portent plus particulièrement sur les méthodes adaptées à la caractérisation de signaux non-stationnaires : temps-fréquence, temps-échelle (décomposition en ondelettes [Lesecq et al., 2001]).
- la redondance matérielle, consistant à multiplier les capteurs critiques d'une installation

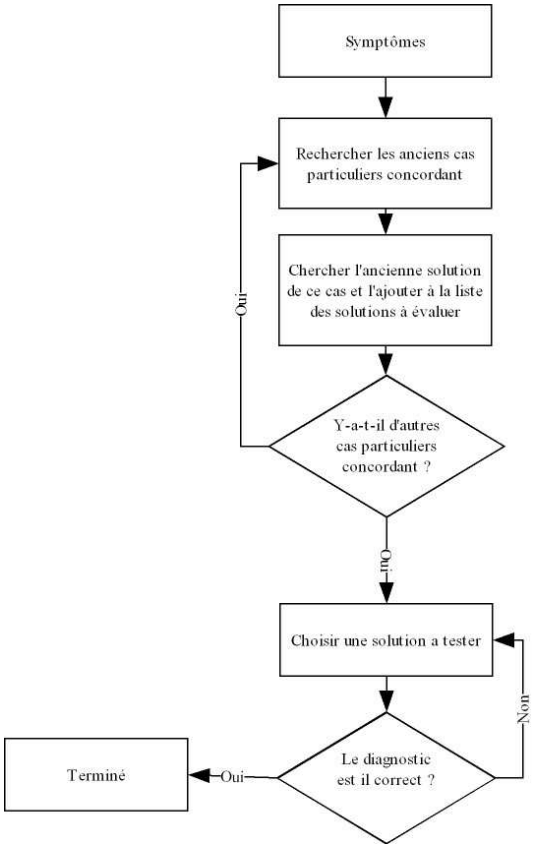


FIG. 1.15 – Procédure d'analyse diagnostique à partir de cas particuliers

– les réseaux de neurones artificiels (RNA) [Bishop, 1994], jouant le rôle de "boîte noire" quand la connaissance du procédé à surveiller est insuffisante.
– les capteurs spécifiques permettant de donner directement l'état d'un composant, comme par exemple les capteurs de fin de course.

1.4.5 Le diagnostic à base de modèles

Principe

L'idée consiste à vérifier la consistance entre le comportement attendu d'une installation, d'un procédé à l'aide du modèle de son comportement, et le comportement réellement observé. La présence de différences entre le comportement attendu et le comportement réel est le point de départ de la recherche de diagnostics. Cette idée fondamentale du diagnostic à base de modèle est représentée par la figure 1.16 [Piechowiak, 2003].

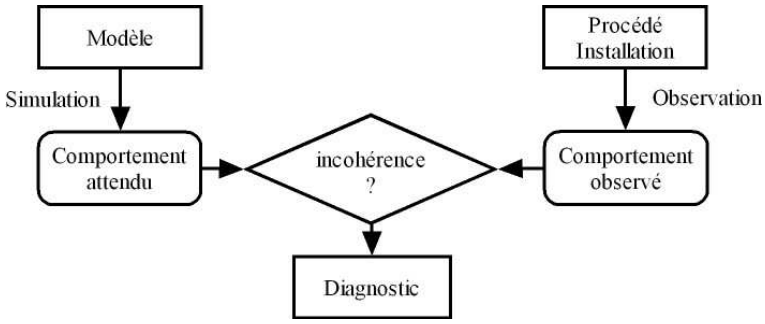


FIG. 1.16 – Principe de la détection à base de modèle

Le raisonnement causal

Dans [Gentil et al., 2004], l'idée consiste à déterminer les composants a priori défaillants expliquant le comportement anormal observé. Un aspect significatif de la connaissance des mécanismes en termes de causalité est nécessaire au moment de l'analyse des régimes perturbés. Cette connaissance peut être représentée sous la forme de graphes causaux (page 42).
Ainsi, $x \rightarrow y$ signifie que l'état de y au temps t dépend de l'état de x au temps $t' \leqslant t$; x est la cause et y l'effet. Ce principe est valide aussi longtemps que les liens causaux ne sont pas modifiés par les défauts.
Le procédé fournit en ligne les mesures de l'ensemble des variables mesurées ; la comparaison entre grandeurs simulées et mesurées donne à chaque instant la liste des défauts : lorsqu'un écart significatif est signalé entre les deux grandeurs, la variable associée est considérée comme étant en défaut. C'est l'étape de détection.
La procédure de détection fournit donc une liste de variables en défaut. Cette liste risque de croître au fur et à mesure que les effets d'une défaillance se propagent dans l'installation et qu'un plus grand nombre de variables sont atteintes par des défauts selon que l'on est en présence d'un système en boucle fermée ou non. Pour localiser l'origine du défaut, appelé défaut source, il faut obtenir une liste ordonnée de variables, depuis le défaut source jusqu'aux ultimes conséquences, en passant par toutes les variables intermédiaires affectées. Pour cela, on cherche à évaluer l'écart entre valeurs prédites et valeurs mesurées, de manière récursive en recherchant les antécédents suspects dans le graphe [Montmain & Gentil, 2000].

Le raisonnement FDI

Dans la communauté FDI (Fault Detection and Isolation) [Frank, 1996] [Iserman, 1997], chaque composant d'un système est décrit par un modèle comportemental défini par un ensemble de contraintes statiques ou dynamiques liant ses variables d'entrée et de sortie.
Comme son nom l'indique, l'approche FDI se déroule en deux étapes : la partie détection puis la partie localisation.

Avant de réaliser l'analyse diagnostique, des tests de détection doivent être réalisés permettant de générer des symptômes. Ces tests sont orientés bon ou mauvais fonctionnement, reposant sur différents modèles. Plus précisément, un test de détection repose sur un modèle de comportement ne faisant intervenir nécessairement que des variables et paramètres connus mesurés ; un autre nom de ce test est par exemple *relation de redondance analytique* (RRA). On la note généralement $\omega(OBS) = 0$, où OBS est un ensemble d'observations.

Autrement dit, en fonctionnement normal, les mesures satisfont les RRAs. En présence de défauts, les RRAs ne sont plus forcément satisfaites et on peut avoir $\omega(OBS) = r \neq 0$ où r est appelé *résidu*. Les RRAs sont obtenues par combinaison des contraintes [Staroswiecki et al., 2000] [Düstegör et al., 2004] [Ploix et al., 2005].

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

- la génération de résidus directionnels [Gertler, 1991] : les résidus sont conçus de telle sorte que le vecteur des résidus reste confiné dans une direction particulière de l'espace des résidus, en réponse à un défaut particulier. Cette approche est en fait une spécialisation des résidus structurés dans laquelle chaque défaillance est représentée par un vecteur dans l'espace des symptômes
- la génération de résidus structurés [Gertler & Monajemy, 1993] : les résidus sont conçus de façon à répondre à des sous-ensembles de défauts différents.

Approche par table de signature

A partir des RRAs, on définit la **table de signature des défauts**. La signature théorique d'un défaut peut être envisagée comme la trace attendue du défaut sur les différentes RRAs qui modélisent le système [Patton & Chen, 1997] [Gertler, 1991] [Frank, 1996].

Étant donné un ensemble $R = \{RRA_1, \ldots, RRA_n\}$, de n RRAs et un ensemble $F = \{F_1, \ldots, F_m\}$ de m défauts, la signature du défaut F_j est donnée par le vecteur binaire $FS_j = (s_{1j}, \ldots, s_{nj})^T$, où :

- $s_{ij} = 1$ si des composants impliqués dans F_j sont impliqués dans le RRA_i
- $s_{ij} = 0$ sinon

Exemple

On considère le circuit suivant (Figure 1.17) composé de trois lampes liées en parallèle à une source de tension. On suppose que chacune des lampes possède une résistance interne.

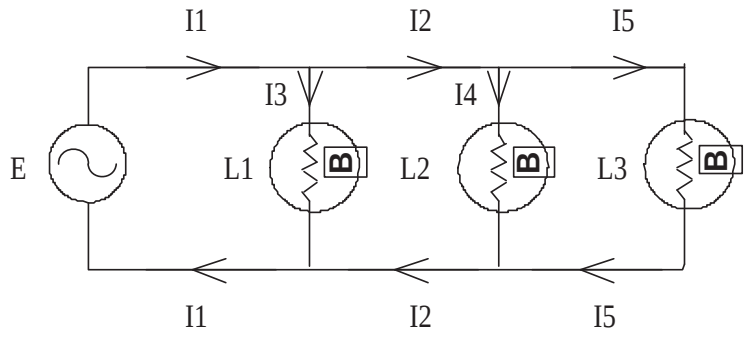


FIG. 1.17 – Circuit électrique composé de 3 lampes en parallèle

Les contraintes régissant ce système sont les suivantes :

$$
\begin{aligned}
&E_1(source) : \bar{I_1} = I_1 \\
&E_2(noeud1) : I_1 = I_2 + I_3 \\
&E_3(noeud2) : I_2 = I_4 + I_5 \\
&E_4(ampoule1) : L_1 = f_1(I_3) \\
&E_5(ampoule2) : L_2 = f_2(I_4) \\
&E_6(ampoule3) : L_3 = f_3(I_5) \\
&E_7(capteur2) : \bar{I_2} = I_2 \\
&E_8(capteur3) : \bar{L_1} = L_1 \\
&E_9(capteur4) : \bar{L_2} = L_2 \\
&E_{10}(capteur5) : \bar{L_3} = L_3
\end{aligned}
$$

où $E_i(composant)$ est la contrainte décrivant le comportement du *composant*, où $\bar{X}$ est la valeur mesurée de X et où L représente la quantité de lumière et I l'intensité.

Hypothèse : on suppose que les capteurs fonctionnent correctement, autrement dit que les valeurs mesurées correspondent aux valeurs réelles.
Graphiquement, la figure 1.18 représente les contraintes ainsi que le résultat de la génération des RRAs obtenues par la méthode des graphes bipartis (Figure 1.19).

On obtient ainsi la table de signature des défauts détaillée dans le Tableau 1.5.

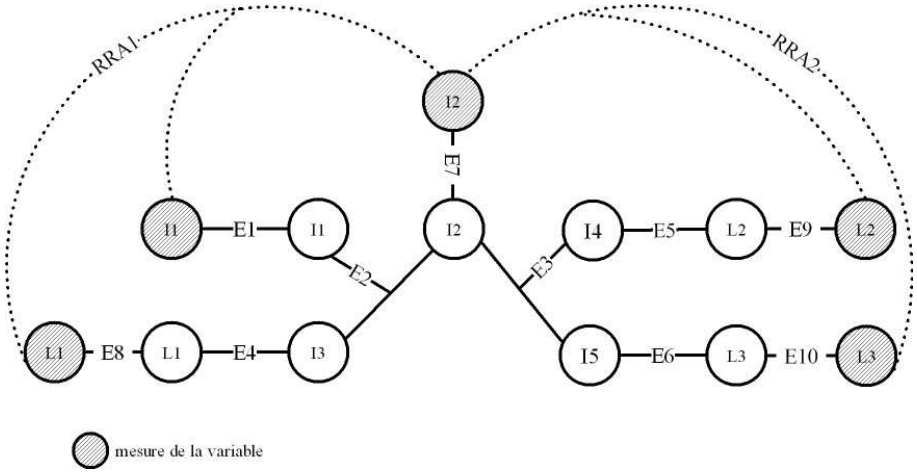


FIG. 1.18 – Représentation graphique des contraintes

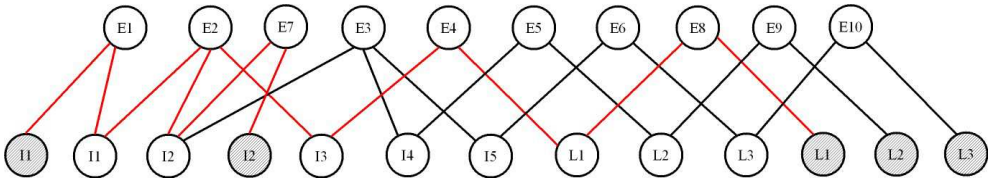


FIG. 1.19 – Détermination de la RRA_1 par la méthode des graphes bipartis

	Défaut Source	Défaut Noeud 1	Défaut Noeud 2	Défaut Ampoule 1	Défaut Ampoule 2	Défaut Ampoule 3
RRA_1	1	1	0	1	0	0
RRA_2	0	0	1	0	1	1

TAB. 1.5 – Table de signature des défauts

Où :

$$RRA_1 = \begin{cases} \hat{L}_1 = f_1\left(\bar{I}_1 - \bar{I}_2\right) \\ r_1 = \bar{L}_1 - \hat{L}_1 \end{cases}$$

Et

$$RRA_2 = \begin{cases} \hat{L}_3 = f_3(\bar{I}_2 - f_2^{-1}(\bar{L}_4)) \\ r_2 = \bar{L}_3 - \hat{L}_3 \end{cases}$$

avec $\bar{X}$ la valeur mesurée de X et $\hat{X}$ sa valeur estimée.

On suppose les observations suivantes :
- L'ampoule 1 est éteinte $L_1 = 0$
- L'ampoule 2 est éteinte $L_2 = 0$
- L'ampoule 3 est allumée $L_3 \neq 0$
- $I_2 = 20mA$
- $I_1 = 50mA$

Donc $r_1 \neq 0$ et $r_2 \neq 0$.

La table de signature des défauts, considérée ainsi, ne donne aucun diagnostic d'après l'approche FDI puisque aucun défaut n'a pour signature $[1, 1]^T$ en considérant que les défauts multiples ne sont pas pris en compte (pour cela une anticipation de ces défauts multiples aurait été nécessaire)

Le raisonnement DX

Pour la communauté DX [DeKleer & Williams, 1987] [Hamscher et al., 1992], un système peut être modélisé par un triplet $(SD, COMPS, OBS)$ avec :
- SD la description du système : c'est un ensemble contenant un ensemble de formules logiques constituées de prédicats du premier ordre avec égalité
- $COMPS$ un ensemble fini de labels représentant les composants du système
- OBS est un ensemble d'observations

Les éléments de $COMPS$ sont les éléments à diagnostiquer. Ils sont également présents dans SD et éventuellement dans OBS. L'ensemble SD inclut un prédicat unitaire noté **AN** qui signifie *anormal*. Pour un composant $c \in COMPS$, la notation $\neg AN(c)$ signifie que le composant c fonctionne correctement, autrement dit qu'il est dans un état normal. Par ailleurs, SD contient en général uniquement des relations décrivant le bon fonctionnement.
Dans la suite, des extensions de formalisme ont été proposées pour pouvoir tenir compte des modèles de dysfonctionnement [Struss & Dressler, 1989]. Ainsi, au lieu d'avoir simplement deux modes de fonctionnement (un mode normal modélisé et un mode anormal non modélisé), d'autres modes de dysfonctionnement peuvent s'ajouter au mode de fonctionnement normal.

La recherche de diagnostics par l'approche DX consiste en l'utilisation d'un solveur utilisant les données du modèle décrites précédemment.
Par ailleurs, pour faire le rapprochement avec l'approche FDI [Ploix et al., 2003] [Nyberg & Krysander, 2003], la recherche de conflits au sens de l'approche DX peut s'appuyer sur la recherche de RRAs, puis consiste à combiner les signatures de défauts pour rechercher des diagnostics minimaux. Pour générer ces diagnostics minimaux, on peut s'appuyer sur l'algorithme décrit dans [Reiter, 1987]. Les ensembles de diagnostics sont ensuite utilisés pour construire des arbres appelés H-S (*Hitting-Set*). Dans ces arbres, chaque chemin allant de la racine vers une feuille anotée par un "$\surd$" est un diagnostic minimal (Figure 1.20).
Pratiquement, un diagnostic *minimal* est le plus petit sous-ensemble d'explications expliquant à lui-seul les observations (des capteurs et des actionneurs). En conséquence, tout *sur-diagnostic* construit à partir des diagnostics minimaux est un diagnostic.

Exemple

On considère à nouveau l'exemple composé de trois lampes liées en parallèle à une source de tension (Figure 1.17). On suppose les observations suivantes :

- L_1 éteinte
- L_2 éteinte
- L_3 allumée
- $I_2 = 20mA$
- $I_1 = 50mA$.

A partir des RRAs construites dans le paragraphe sur le raisonnement FDI, on voit que le test 1 est faux et que le test 2 est faux (tableau 1.5).
On peut alors chercher les diagnostics minimaux en considérant les composants impliqués dans un RRAs comme étant à l'origine d'un R-conflit au sens du diagnostic logique. Les diagnostics provenant des RRAs sont $\{Source, Noeud1, Ampoule1\}$ pour le test 1 et $\{Noeud2, Ampoule2, Ampoule3\}$ pour le test 2.

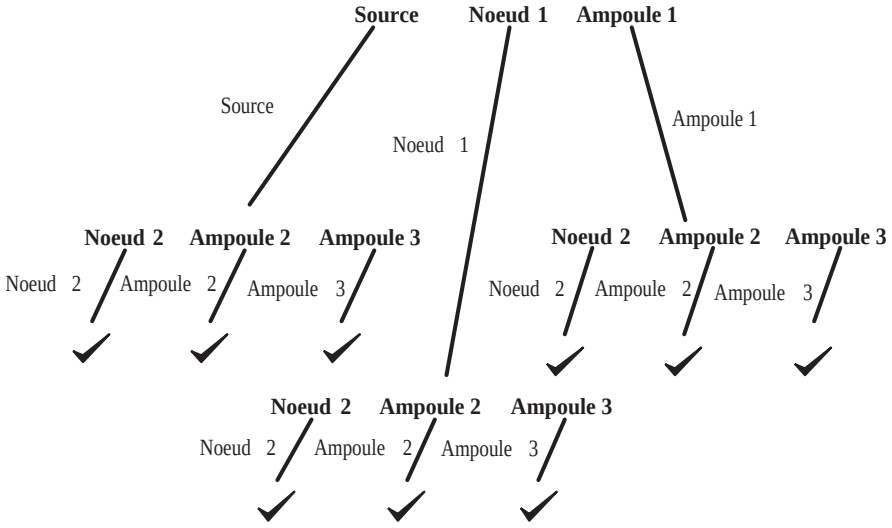


FIG. 1.20 – Méthode de l'arbre H-S

En recherchant les diagnostics minimaux par la méthode de l'arbre H-S (Figure 1.20), on obtient les diagnostics minimaux suivants :$\{Source, Noeud2\}$, $\{Source, Ampoule2\}$, $\{Source, Ampoule3\}$, $\{Noeud1, Noeud2\}$, $\{Noeud1, Ampoule2\}$, $\{Noeud1, Ampoule3\}$, $\{Ampoule1, Noeud2\}$, $\{Ampoule1, Ampoule2\}$, $\{Ampoule1, Ampoule3\}$.

Discussions

Dans les deux approches, le diagnostic est déclenché quand des inconsistences apparaissent entre le comportement modélisé (bon comportement) et les observations. La force de ces

approches réside dans le fait qu'elles peuvent être basées uniquement sur le comportement normal des composants : aucun modèle de mauvais comportement n'est nécessaire.
L'hypothèse forte des résultats de l'approche FDI est qu'un défaut se manifeste nécessairement en affectant les RRAs dans lesquelles il est impliqué.
En analysant la table de signature des défauts, il en résulte que chaque défaut impliqué dans une RRA de la table de signature des défauts est un défaut candidat, mais aussi que chaque composant impliqué dans une RRA satisfaite est implicitement disculpé. Ce résultat s'appuie donc sur une hypothèse d'exonération implicite dans l'approche FDI, qui n'existe pas dans l'approche DX [Cordier et al., 2000].

Par exemple, considérons à nouveau les tests issus de l'analyse par l'approche FDI (tableau 1.6)

	Défaut Source	Défaut Noeud 1	Défaut Noeud 2	Défaut Ampoule 1	Défaut Ampoule 2	Défaut Ampoule 3
RRA_1	1	1	0	1	0	0
RRA_2	0	0	1	0	1	1

Tab. 1.6 – Table de signature des défauts

Si on fait l'hypothèse que seul le test 1 est faux (ce qui sous entend que le test 2 est vrai), alors l'approche FDI donne comme diagnostic l'ensemble $\{Source, Noeud1, Ampoule1\}$ tout en disculpant l'ensemble $\{Noeud2, Ampoule2, Ampoule3\}$.

D'un autre côté, le fait de modéliser uniquement le bon comportement a ses limites. Dans l'exemple des trois lampes, si on considère les observations données dans l'exemple pour la partie DX, on remarque que :

- la table de signature de la communauté FDI ne donne aucun diagnostic, malgré l'apparition évidente de défauts
- les diagnostics minimaux de la communauté DX sont en partie cohérents, mais parmi ces diagnostics, on trouve des diagnostics physiquement impossibles. Par exemple $\{Source, Ampoule3\}$, signifierait que la source est en défaut (ne produit pas d'énergie) et la lampe aussi (en étant allumée), ce qui est physiquement impossible.

1.4.6 Conclusions

Les modèles comportementaux associés aux hypothèses de fonctionnement des composants est le lien fort qui va permettre de faire le lien entre les résultats de l'analyse de risque et les méthodes de diagnostic : d'un côté les hypothèses sur l'état du composant (bon ou mauvais) pour le diagnostic et de l'autre côté l'analyse des risques mettant en avant les relations causes

conséquences en termes de défaut et de dysfonctionnement.
D'un autre côté, le formalisme proposé aujourd'hui pour l'AMDEC ne permet pas à cette méthode de s'intégrer facilement aux méthodes du diagnostic. Un formalisme et un contenu adapté permettraient son intégration aux méthodes de diagnostic. On voit donc encore une fois la nécessité de modèles structurel, fonctionnel et comportemental commun à l'analyse des risques et le diagnostic.

1.5 Démarche de la thèse

"Proposer une approche intégrée analyse de risques/analyse diagnostique" : tel est l'objectif de cette thèse. Face à la pluralité des outils d'analyses de risque existants, nous avons opté pour l'analyse AMDEC qui permet de déterminer les relations de cause à effet de manière systématique. Les résultats de cette analyse AMDEC seront pris en compte en tant que connaissance experte du système. Ensuite, pour pouvoir prendre en compte les combinaisons de type conjonction, disjonction, etc. entre les événements, nous utiliserons la méthode de l'arbre de défaillance appliquée aux résultats de l'AMDEC.
Une fois obtenus, les résultats de cette analyse seront intégrés à l'analyse diagnostique de manière à l'affiner. Enfin, les résultats de l'analyse diagnostique combinés à ceux de l'analyse des risques vont nous permettre de pronostiquer les défaut et les défaillances du système.

Pour atteindre ces objectifs, les étapes suivantes seront introduites :

- le chapitre 2 propose une modélisation structurelle et fonctionnelle du système à analyser ainsi qu'un formalisme pour la connaissance experte issue de l'analyse AMDEC de manière à ce que ses résultats puissent s'intégrer dans une analyse diagnostique.
  D'autre part, une modélisation comportementale du système est également introduite de manière à :
  - distinguer les différents modes possibles des composants du système (mode correct, anormal, etc.)
  - ce que les résultats de l'analyse diagnostique puissent eux aussi s'intégrer à l'analyse AMDEC.
- dans la suite, le chapitre 3 propose des procédures de diagnostic, basés sur l'approche DX, intégrant les résultats de l'analyse AMDEC. Cette intégration apporte deux contributions :
  - elle permet l'élimination de diagnostics physiquement impossibles
  - elle permet de distinguer les défauts primaires parmi les diagnostics
- enfin, le chapitre 4 propose d'intégrer les résultats de l'analyse diagnostique à l'analyse AMDEC dans le but de pronostiquer les défauts et les défaillances du système en fonction des diagnostics.

Chapitre 2

Formalisation fonctionnelle et comportementale en vue d'une coopération entre analyse des risques et diagnostic

2.1 Introduction

Pour mettre en évidence les liens entre analyse de risques et diagnostic, il est nécessaire de rechercher les concepts de l'un et de l'autre et de créer ainsi des modèles et des outils permettant l'interactivité. Pour modéliser un système, plusieurs types de connaissances sont disponibles ; Chittaro et Kumar, dans [Kumar & Chittaro, 1998], définissent quatre différentes classes de connaissances :

- *Structurelle* : que contient le système physique et comment les éléments du système sont connectés ?
- *Comportementale* : comment se comporte chaque élément ?
- *Fonctionnelle* : quel est le rôle de chaque élément ?
- *Téléologique* : dans quel but chaque élément a-t-il été installé ?

Ces classes de connaissance coïncident avec la modélisation **FISE** [Flaus et al., 2006] que nous avons présenté au chapitre précédent :

- l'aspect **F**onctionnel de la modélisation FISE correspond à la classe de connaissance *fonctionnelle*
- les aspects **I**nteractionnel et **S**tructurel correspondent à la classe de connaissance *structurelle*
- l'**E**tat (i.e. les grandeurs décrivant le comportement des entités) est lié à la connaissance *comportementale*

Nous avons vu dans le chapitre précédent que la plupart des analyses de risques se basent sur une modélisation structurelle et fonctionnelle du système alors que le diagnostic repose essentiellement sur une modélisation structurelle et comportementale. Dans ce chapitre, nous allons donc développer un formalisme unique de modélisation permettant d'appréhender les différents niveaux de modélisation présentés ci-dessous et de faire le lien entre eux pour définir les liens possibles entre analyse de risques et diagnostic :

- une modélisation structurelle sur laquelle s'appuient le diagnostic et l'analyse des risques
- une modélisation comportementale sous la forme de contraintes ainsi qu'un formalisme adapté aux besoins du diagnostic
- une modélisation fonctionnelle et un formalisme permettant d'appréhender cette modélisation qui servira de base à l'analyse des risques. Cette modélisation fonctionnelle n'est pas uniquement une abstraction du modèle comportemental, mais peut apporter une connaissance experte supplémentaire au modèle comportemental issue de l'analyse AMDEC du système. Par exemple, un circuit électrique est généralement représenté suivant le point de vue électrique par son modèle de comportemental. Or si le point de vue thermique est important pour l'analyse des fautes, celui-ci n'est pris en compte que dans l'analyse AMDEC.

2.2 Modélisation par contraintes pour le diagnostic

Dans ce paragraphe, nous allons développer un formalisme pour la modélisation structurelle et comportementale qui permettra par la suite la mise en oeuvre d'outils de diagnostic d'une part, et de faire le lien avec l'analyse des risques d'autre part. En effet, pour pouvoir intégrer les résultats de l'analyse AMDEC à l'analyse diagnostique en tant que connaissance experte, il faut qu'un formalisme unique en termes de modélisation pour ces deux analyses soit développé, faute de quoi les notions communes (par exemple celle de défaut) ne seront pas compatibles entre les analyses et donc aucune intégration ne sera possible.

2.2.1 Définitions

L'analyse AMDEC et l'analyse diagnostique repose toutes les deux sur une modélisation structurelle du système physique à étudier. Ainsi, dans un premier temps, nous allons poser les bases de la modélisation structurelle et comportementale en identifiant les différents concepts couverts par ces modèles, à savoir :

- les ressources
- les modes de comportement des ressources

Ressource

Définition 1 (Système)**.** Dans notre cadre, nous appelons ***système***, un ensemble déterminé d'éléments interconnectés ou en interactions, les éléments interconnectés étant appelés ***ressources***.

Définition 2 (Ressource)**.** On appelle *ressources* d'un système, les moyens (matériels, humains, énergétiques) dont dispose ou peut disposer le système.

La notion de *ressource* est préférée à celle de *composant* habituellement utilisée pour un découpage structurel pour trois principales raisons :

- la notion de ressource est plus précise que celle de composant, considéré habituellement comme la plus petite partie d'un système qu'il est nécessaire et suffisant de considérer pour l'analyse du système [Villemeur, 1988].
  La notion de ressource dénote la notion d'*élément nécessaire* au bon fonctionnement du système, alors que la notion de composant dénote simplement la notion de découpage.
- la notion de ressource, plus connue dans le domaine des analyses fonctionnelles, sera reprise pour la modélisation fonctionnelle proposée ultérieurement
- nous souhaitons adopter une approche "macroscopique" du système à l'instar des analyses fonctionnelles

Notation 1. On note $\Pi = \{r_1, \ldots, r_p\}$ l'ensemble des ressources du système.

D'autre part, tout comme les approches de diagnostic présentées dans le chapitre précédent, nous définissons un certain nombre de variables qui vont permettre de décrire les comportements de chacune des ressources.

Notation 2. On note $\Omega = \{V_1, \ldots, V_n\}$ l'ensemble des variables du système.

A l'aide de ces variables, nous définissons l'ensemble des variables permettant la description du comportement d'une ressource r (Figure 2.1) par :

$$Rdom : \begin{array}{ccc} \Pi & \rightarrow & P(\Omega) \\ r & \mapsto & \{V_1, \ldots, V_s\} \end{array}$$

où $P(\Omega)$ est l'ensemble des parties de Ω.

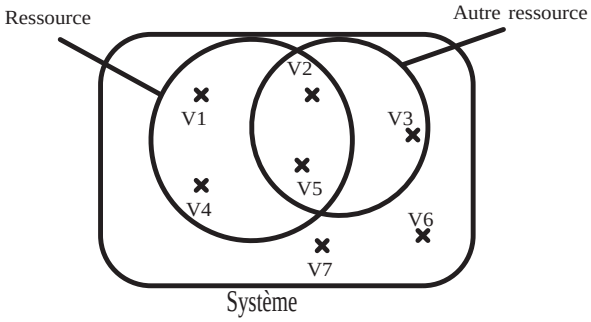


FIG. 2.1 – Représentation d'une ressource

Mode de comportement

Chaque ressource peut se comporter de différentes façons. Elle se trouve alors dans ce qu'on appelle un mode de comportement particulier.

Définition 3 (Modes de comportement d'une ressource)**.** Soit $r \in \Pi$ une ressource. Nous définissons l'ensemble des ***modes*** de r, noté $modes(r)$, par

$$modes(r) = \{CM, FM_1, \ldots, FM_m\}$$

où :
- CM (*Correct Mode*) correspond au mode normal
- FM_j (*Fault Mode j*) sont des modes de défauts correspondant à des défauts spécifiques de la ressource (idée que l'on retrouve dans [Struss & Dressler, 1989])

Cette notion de mode est similaire à celle développée dans [Struss & Dressler, 1989], où le mode d'un composant représente son état.

Probabilité et gravité de mode

Pour compléter cette notion de mode, on associe à chacun d'eux une probabilité d'occurrence, en considérant que les modes sont indépendants les uns des autres. On notera alors $P(CM(r))$ la probabilité que la ressource r soit dans son mode correct CM et $P(FM_i(r))$ la probabilité que la ressource soit dans son mode de défaut FM_i.
En conséquence, on a :

$$\forall r \in \Pi, P(CM(r)) + \sum_i (P(FM_i(r))) = 1$$

Cette notion de probabilité de mode, et plus particulièrement de mode de défaut, a un double intérêt :
- lors de la phase de diagnostic, il sera possible d'évaluer les probabilités d'occurrence de chacun des diagnostics
- lors de la phase de pronostic, il sera possible de choisir le diagnostic le plus probable pour l' "injecter" dans l'analyse des risques, dans le cas où plusieurs diagnostics potentiels sont identifiés pendant l'analyse diagnostique

Par ailleurs, pour faire le lien avec les analyses de risques, nous proposons ici d'associer à chaque mode de défaut $FM_i(r)$ d'une ressource r une gravité, de manière à quantifier les dommages que ce mode pourrait infliger au système. Cette gravité sera notée $G(FM_i(r))$ et sera évaluée sur une échelle arbitraire de 11 points $\{0, 1, 2, \ldots, 10\}$, avec $G = 0$ signifiant que le mode de défaut n'aura aucun impact en termes de gravité sur le système et $G = 10$ signifiant que le mode de défaut aura un impact important en termes de dommage sur le système.
Cette quantification du danger possible représenté par le mode de défaut nous sera utile par la suite, quand il s'agira de trier les risques possibles du système lors de la phase de pronostic.

Dans la section suivante, nous proposons une modélisation comportementale reposant sur la notion de mode. L'objectif est de définir un modèle comportemental du système pour que, lors d'une analyse diagnostique, non seulement les ressources a priori défaillantes soient détectées, mais aussi que les modes de défaut associés à ces ressources en défaut soient déterminés.
Le fait que le mode de défaut de la ressource soit localisé est nécessaire pour que les résultats de l'analyse diagnostique puissent être intégrés à ceux de l'analyse des risques, car, comme nous le verrons dans la suite, l'analyse AMDEC repose en partie sur la recherche de relations cause/effet impliquant des modes de défaut.

2.2.2 Modélisation comportementale

L'intérêt du formalisme pour la modélisation comportementale vient du fait que, comme annoncé précédemment, nous souhaitons effectuer un diagnostic sur notre système. En effet, l'objectif est de déterminer dans quels modes les différentes ressources sont susceptibles de se trouver, et, si certaines sont a priori dans un mode de défaut, à quelles conséquences cela peut conduire sur le système.

Définition 4 (Contrainte)**.** On appelle *contrainte*, une proposition pouvant être soit vraie, soit fausse :

- Si la relation logique est vraie, la contrainte est dite *satisfaite*
- Si la relation logique est fausse, la contrainte est dite *insatisfaite*

Le comportement d'une ressource r sera décrit par une ou plusieurs *contraintes*, notées $C_j(V_1, \ldots, V_k)$ basées sur l'ensemble de variables de $Rdom(r)$.
Ces variables peuvent être classées en 3 catégories :

- les variables caractérisant les actions provenant de l'environnement extérieur au système
- les variables internes à la ressource
- les variables caractérisant les actions sur l'environnement extérieur au système

Notation 3. L'ensemble des contraintes d'une ressource $r \in \Pi$ sera noté $Cons(r)$.

Dans cette partie, différents types de modèles de comportement sont présentés en vue d'une analyse diagnostique (détection et localisation). Seront décrits dans la suite de ce paragraphe :

- un modèle de *comportement correct*, constitué des contraintes décrivant le bon comportement des différentes ressources
- un modèle de *comportement physiquement impossible*, reposant sur des contraintes décrivant uniquement les comportements physiquement impossibles des ressources
- des modèles de *mauvais comportements* constitués des contraintes décrivant les comportements des ressources dans un mode de défaut particulier

Modèle de bon comportement

Le modèle de bon comportement d'une ressource r est décrit par plusieurs contraintes s'appuyant sur l'ensemble des variables $Rdom(r)$. Si ces contraintes ne sont pas satisfaites, alors la ressource n'est pas dans le mode correct (CM).

Étant donné que les variables décrivant le système sont habituellement définies sur des ensembles denses de valeurs, le principe de non-exonération [Cordier et al., 2000] ne permet pas de conclure que la ressource fonctionne correctement si la contrainte est satisfaite. En effet, pour conclure que la ressource est dans le mode correct, la contrainte doit être satisfaite pour chacune des valeurs possibles des variables, ce qui est rarement vérifiable.

Nous proposons ici une abstraction discrète des variables pour que, si la contrainte est satisfaite, la ressource est alors dans le mode associé à cette contrainte.
Par ailleurs, cette discrétisation des variables va nous permettre de mettre en évidence les intervalles de sécurité pour certaines variables, comme par exemple, scinder l'ensemble des valeurs possibles de la pression dans un conteneur en 3 zones : zones de pression trop faible (qualité du produit détérioré), zone de pression optimale, zone de pression de danger (risque d'explosion). Cette décomposition en intervalles de sécurité peut s'appuyer sur les seuils de sécurité donnés par les organes de mesures (Annexes : section 5.3 en page 179).

Pour discrétiser les variables, leur domaine de variation est divisé en zones ou intervalles. Ainsi, un découpage en n zones conduit à n valeurs discrètes possibles.

En conséquence, le modèle de comportement définit les jeux de valeurs possibles des variables impliquées dans les contraintes. Les contraintes obtenues sont dites *qualitatives*.

Ainsi, une contrainte n'est autre qu'un jeu de valeurs discrètes (ou tuple).

Par ailleurs, en fonction des découpages en zones des variables et donc des différentes valeurs possibles, il se peut qu'il y ait plusieurs contraintes qualitatives correspondant au bon comportement. Dans ce cas :

- si une contrainte qualitative est satisfaite, on peut donc conclure que le comportement de la ressource est normal pour ce point de fonctionnement particulier.

  Note :
  Ce modèle de bon comportement est celui utilisé dans l'approche DX du diagnostic. En utilisant ce formalisme, une contrainte caractérisant le bon comportement d'une ressource r est noté $ok(r) \rightarrow C(V_1, \ldots, V_k)$ vérifié, signifiant "la contrainte est vraie sous hypothèse que le mode de la ressource r est normal".

- s'il est possible de vérifier que toutes les contraintes qualitatives sont satisfaites et si elles le sont, alors on peut conclure que la ressource fonctionne correctement (dans le cas où on est capable de mesurer toutes les variables impliquées dans les contraintes)

Modèle de mauvais comportements

Pour compléter le modèle de bon comportement, on définit des modèles de comportement associés à des modes de défaut connus (FM) par le biais de contraintes. Ces modèles peuvent être utilisés par un algorithme de diagnostic (chapitre 3) de manière à mieux isoler un défaut. En effet, le mode de défaut sera suspecté si sa contrainte associée est satisfaite. Par ailleurs, une probabilité d'occurrence peut être attribuée à chaque mode de défaut ainsi qu'à chaque défaut pour permettre de trier les modes de défaut et les défauts à vérifier durant la procédure de diagnostic.

Modèle de comportement physiquement impossible

Enfin, pour étendre le modèle de bon comportement et de mauvais comportement, nous définissons ici un modèle de comportement physiquement possible. Pour un ensemble de valeurs possibles des variables de $Rdom(r)$ caractérisant une ressource r, le modèle de comportement physiquement possible donne l'ensemble des valeurs possibles des variables (on pourra noter que ce modèle de comportement inclut le modèle de bon comportement et de mauvais comportement).

Habituellement, lorsque l'on définit des contraintes de comportement d'une ressource, on sous entend que les domaines complets de valeurs des variables impliquées dans les contraintes peuvent être décrits. Cependant, il peut être intéressant, dans certains cas, de restreindre ces domaines de valeurs aux seuls jeux de valeurs "physiquements possibles". Ainsi, lors de la procédure de diagnostic, il sera possible d'éviter l'exploration de comportements physiquement impossibles de ressources et donc la présence de diagnostics physiquement impossibles (notion d'impossibilité physique développée également dans [Friedrich et al., 1990]). Si les contraintes liées au comportement physiquement possible sont satisfaites, la ressource est dite dans un mode physiquement possible.

En réalité, il se peut que dans certains cas, il soit plus facile de déterminer les comportements physiquement impossibles et leurs contraintes associées. Dans ce cas, la ressource est dite dans un mode *virtuel* physiquement impossible, noté (PIM).
En fait, étant donné qu'un tel comportement est impossible, la ressource n'est dans un aucun mode réel ; cependant, la notation PIM est utilisée pour repérer les jeux de valeurs physiquement impossibles.

Grâce à ces modèles de comportement complémentaires, le chapitre 3 proposera des procédures de diagnostic de manière à intégrer les résultats de l'analyse AMDEC dans l'analyse diagnostique.

2.2.3 Exemple

Considérons le circuit électrique en figure 2.2 :

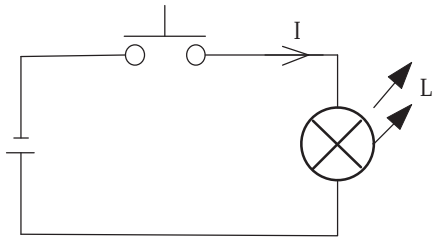


FIG. 2.2 – Circuit électrique comportant une lampe et un interrupteur

Le comportement de la lampe peut s'exprimer à l'aide des variables I, U et L, I étant l'intensité circulant dans le circuit, U la tension et L la quantité de lumière émanant de l'ampoule fonction de I.

L'intensité nécessaire pour alimenter l'ampoule est donnée par : $\begin{cases} I \in]-30mA, -1mA[\cup]1mA, 30mA[\\ I \in]-1mA, +1mA[\end{cases}$ définissant respectivement les intervalles d'intensité suffisantes et insuffisantes pour alimenter la lampe. On suppose qu'au delà de $30mA$ en valeur absolue, le filament rompt (donc la lampe n'éclaire pas) et qu'en dessous de $1mA$ en valeur absolue il n'y a pas assez de courant pour que la lampe s'éclaire.

On décide de discrétiser la variable I avec $dom(I) = \{0, 1\}$ où :
- "0" signifie "courant trop faible ou nul"
- "1" signifie "courant suffisant"

Pour discrétiser L, on choisit un autre type de découpage, à savoir la présence ou non de lumière émanant de la lampe.
On considère désormais L discrétisée avec $dom(L) = \{0, 1\}$ où :
- "0" signifie "absence de lumière"
- "1" signifie "présence de lumière"

Pour la discrétisation de U, la même idée est conservée que pour celle de L. On choisit de s'occuper de la présence (1) ou de l'absence (0) de tension dans le circuit. On a donc $dom(U) = \{0, 1\}$.

Formellement, on a donc $Rdom(lampe) = \{I, U, L\}$.
L'espace du domaine des variables de la lampe est représentée en figure 2.3.

L'ensemble des contraintes décrivant le comportement de la lampe, représenté en figure 2.4, peut être expliqué ainsi :
- Le cas $I = 0$, $U = 0$ et $L = 0$ signifie "l'intensité est insuffisante voire nulle, la tension est absente et il n'y a pas de lumière". Ce point correspond à la fois au mode correct et au mode de défaut (le filament peut être rompu, et sans courant ni tension impossible de

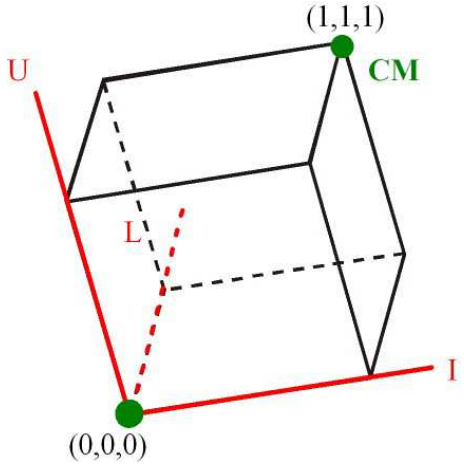


FIG. 2.3 – Espace du domaine des variables de la lampe

vérifier si la lampe fonctionne correctement ou non).

- Le cas $I = 1$, $U = 1$ et $L = 1$ signifie "l'intensité est suffisante, la tension est présente et il y a de la lumière". Ce point correspond au mode correct.

- Le cas $I = 0$, $U = 1$ et $L = 0$ signifie "l'intensité est insuffisante voire nulle, il y a une tension mais il n'y a pas de lumière". Ce point correspond à un mode de défaut. En effet, le fait qu'il y ait une tension mais pas de courant montre que la lampe ne laisse pas passer le courant, donc est dans un mode de défaut (le filament de la lampe pourrait être rompu par exemple).

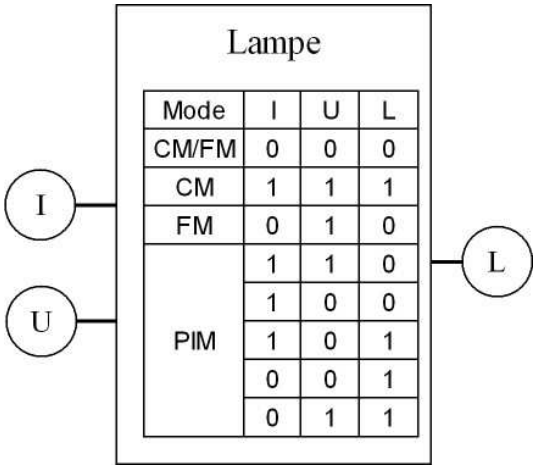
- Le cas $I = 1$, $U = 1$ et $L = 0$ signifie "l'intensité est suffisante, il y a une tension mais il n'y a de lumière". Ce point correspond à un mode physiquement impossible.
  En effet si la lampe était en défaut, il y aurait un court-circuit, donc pas de courant. Or, il y a du courant. Donc ce jeu de valeurs est impossible.

- Le cas $I = 1$, $U = 0$ et $L = 0$ signifie "l'intensité est suffisante, mais il n'y a ni tension ni lumière". Ce point correspond au mode physiquement impossible, car s'il y a du courant, il devrait y avoir forcément une tension.

- Le cas $I = 1$, $U = 0$ et $L = 1$ signifie "l'intensité est suffisante, mais il n'y a pas de tension et il y a de la lumière". Ce point correspond donc également au mode physiquement impossible pour les mêmes raisons que le cas précédent.

- Le cas $I = 0$, $U = 0$ et $L = 1$ signifie "l'intensité est insuffisante voire nulle, il n'y a aucune tension mais il y a de la lumière". Ce point correspond également au mode physiquement impossible. En effet, sans courant il ne peut évidemment pas y avoir de lumière.

– Le cas $I = 0$, $U = 1$ et $L = 1$ signifie "l'intensité est insuffisante voire nulle, mais il y a une tension et de la lumière". Ce point correspond encore une fois au mode physiquement impossible car sans courant, aucune lumière ne peut provenir de la lampe.



Mode	I	U	L
CM/FM	0	0	0
CM	1	1	1
FM	0	1	0
PIM	1	1	0
	1	0	0
	1	0	1
	0	0	1
	0	1	1

FIG. 2.4 – Ensemble des contraintes de la lampe

où FM correspond au mode de défaut : 'filament rompu"

2.2.4 Limites de l'approche

Cette modélisation qualitative des comportements des ressources présente un certain nombre de limites, provenant principalement de la discrétisation des variables :

– bien que cette discrétisation permette de représenter différentes contraintes et de les associer à des modes de comportement, seulement une partie du comportement est décrit par rapport aux contraintes quantitatives qui décrivent le comportement d'une entité pour un ensemble dense de valeurs.
– d'autre part, nous avons choisi de ne pas tenir compte de l'aspect dynamique du système. Cependant, cela n'est pas en soit un problème majeur pour la suite de nos travaux. En effet, les contraintes développées dans cette partie ne serviront qu'à affiner les résultats d'une analyse diagnostique réalisée au préalable (par exemple, en suivant l'approche FDI ou DX) pouvant prendre en compte des contraintes dynamiques et quantitatives. Cet affinage, basé sur ces contraintes qualitatives et statiques, permettra par la suite d'éliminer les diagnostics physiquement impossibles (tout en intégrant les résultats de l'analyse AMDEC) et de localiser précisément les modes de défauts des ressources a priori en défaut.

2.2.5 Lien modèle structurel/comportemental

Cette association entre mode de comportement et contraintes est le lien entre le modèle structurel et le modèle comportemental (figure 2.5).

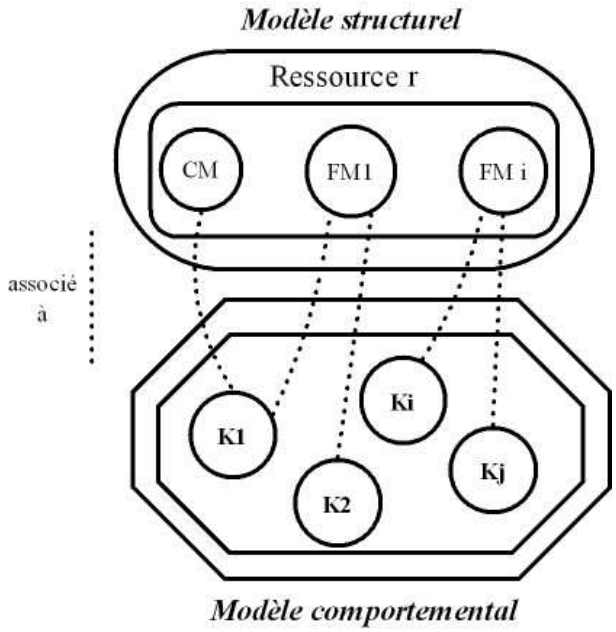


FIG. 2.5 – Lien entre le modèle structurel et comportemental

Ce lien est la base d'une approche diagnostic.
En effet, considérons une ressource r et son ensemble de contraintes associé $Cons(r)$: si une contrainte $K \in Cons(r)$ est vérifiée, alors la ressource est dans un des modes associés à cette contrainte. En particulier, pour le diagnostic, la contrainte vérifiée peut correspondre à un/des modes de défauts.
En conséquence, en utilisant un algorithme adapté (chapitre 3), il est possible de rechercher des contraintes et donc leurs modes de défaut associés pour diagnostiquer un système.

2.2.6 Conclusion

Dans cette partie ont été développés des outils pour une modélisation comportementale pour le diagnostic [Désinde et al., 2006*c*] afin que par la suite l'analyse des risques puisse y être intégrée :

- le modèle structurel est matérialisé par des ressources, découpage nécessaire pour une analyse de risque et pour une analyse diagnostique
- le modèle comportemental est basé :
  - sur une discrétisation des variables provenant soit des niveaux de sécurité des organes

de mesures (*Annexe 1*), soit du sens commun des ingénieurs
- l'établissement d'une liste de contraintes matérialisées par des tuples
- l'association à chaque contrainte d'un certain nombre de modes de comportements (mode correct, mode de défaut, mode physiquement impossible) caractérisant cette contrainte dans la perspective d'une analyse diagnostique

D'un point de vue du diagnostic, on voit donc qu'à partir des valeurs des variables, on est capable de déterminer dans quel mode est la ressource.

2.3 Modélisation fonctionnelle pour l'analyse AMDEC

Après un modèle comportemental pour le diagnostic, nous allons développer, dans ce paragraphe, un formalisme pour la modélisation fonctionnelle qui servira de support à une analyse des modes de défaillances, de leurs effets et de leur criticité (AMDEC). A la différence de [Ressencourt & Travé-Massuyès, 2006], le modèle fonctionnel que nous proposons n'est pas uniquement une abstraction du modèle comportemental mais provient des résultats de l'analyse AMDEC du système.
Un tel formalisme est proposé, puisque, originellement, l'AMDEC est un outil créé pour pouvoir être appliqué à n'importe quel système et doté d'une terminologie volontairement vague pour pouvoir être adaptée. Ainsi, dans cette section nous proposons un formalisme pour une modélisation fonctionnelle en vue d'une analyse AMDEC en détaillant les notions de mode de défaillance, cause, effet, etc. Ce formalisme permettra d'intégrer les résultats de l'analyse AMDEC à l'analyse diagnostique dans le chapitre 3 en tant que connaissance experte complémentaire au modèle comportemental du système.

2.3.1 Définitions

Dans un premier temps, nous allons reprendre et détailler les bases d'un modèle fonctionnel pour l'analyse AMDEC. Pour cela, nous allons définir les notions de :
- fonction
- défaillance et mode de défaillance d'une fonction
- mode de comportement d'une fonction

Fonction

La modélisation fonctionnelle d'un système passe tout d'abord par la définition des **fonctions** que le système doit remplir.

Définition 5 (Fonction)**.** Nous considérons ici que le comportement attendu d'une ressource (ou d'un groupe de ressources) conduit à considérer que cette ressource (ou ce groupe de ressources) à un rôle, une mission, que l'on appelle ici **fonction**.

Notation 4. Formellement, les fonctions seront libellées par un **verbe d'action** (ex :la fonction "ventiler")

On note $\Phi = \{f_1, \ldots, f_q\}$ l'ensemble des fonctions du système telles que *aucune fonction n'est la sous fonction d'une autre fonction.* Autrement dit, aucune fonction n'est incluse dans une autre fonction.

La liste des fonctions $\Phi = \{f_i\}$ est obtenue par un découpage fonctionnel (arbitraire) d'un système.

Nous considérons qu'une fonction $f \in \Phi$, pour se réaliser normalement (Figure 2.6), s'appuie :

- sur un certain nombre de ***ressources*** R_j
  Soit $R(f) = \{R_1, \ldots, R_p\}$ l'ensemble des ressources sur lesquelles s'appuie la fonction f.
- sur un certain nombre de fonctions f_k jouant le rôle de ***services***
  Soit $SF(f) = \{f_1, \ldots, f_r\} \in P(\Phi)$ l'ensemble des fonctions sur lesquelles s'appuie la fonction f. Ces fonctions sont aussi appelées *fonctions supports.*

D'autre part, nous considérons également qu'une fonction f peut servir de support (i.e. participer à un service) à une ou plusieurs fonctions.
On note $ACT(f) = \{f_j / f \in SF(f_j)\}$ l'ensemble des fonctions dont f est support.

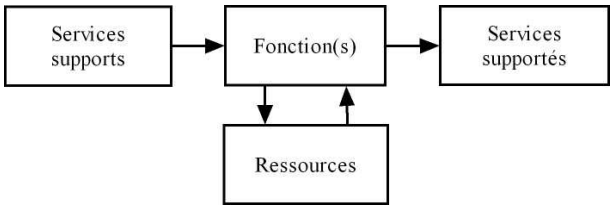


FIG. 2.6 – Représentation graphique d'une fonction

Comparée à la modélisation fonctionnelle proposée dans [Flaus et al., 2006], nous proposons ici :

- une description du comportement des ressources supports sous forme de contraintes qualitatives de manière à rester relativement simple par rapport à un modèle dynamique complet de bon fonctionnement et mauvais fonctionnement
- une approche plus approfondie du mauvais fonctionnement en utilisant l'analyse AMDEC
- une représentation abstraite des flux, vus sous leur angle fonctionnel, qui permettra de compléter les résultats de l'AMDEC

Pour résumer, une fonction :

- s'appuie sur un certain nombre de ressources $R(f)$ et/ou de fonctions $SF(f)$

– peut servir de support à une ou plusieurs autres fonctions $ACT(f)$

Modes d'une fonction

A l'instar des ressources pour lesquels ont été définis un certain nombre de modes de comportement, nous définissons dans ce modèle fonctionnel un certain nombre de modes pour chaque fonction.

Définition 6 (Mode d'une fonction)**.** Soit $f \in \Phi$ une fonction. Nous définissons l'ensemble des modes possibles d'une fonction, noté $modes(f)$, par $modes(f) = \{nf, fm_1, \ldots, fm_r\}$ où :
– nf (normal functioning) correspond au fonctionnement normal
– fm (failure mode) correspond à un mode de défaillance

Lorsqu'une fonction ne se réalise plus ou se réalise incorrectement, on parle de ***défaillance*** qui se définit comme "la cessation d'aptitude d'une (ou d'un groupe d') entité(s) à accomplir une fonction recquise".
Les défaillances de ces entités ont des effets sur les fonctions de celles-ci ; ces effets sont appelés "modes de défaillance" de la fonction et seront nommées par le nom de ces effets. Ainsi, un ***mode de défaillance*** se définit comme "la manifestation par laquelle la défaillance est observée" [Villemeur, 1988].

Définition 7 (Mode de défaillance)**.** Pour affiner la définition initiale d'un mode de défaillance, nous considérons ici qu'un **mode de défaillance** est toute altération de la fonction, en d'autre termes, toute façon qu'a la fonction attendue de ne pas se réaliser correctement.

Cette définition est compatible avec les différents types de mode de défaillance définies dans [Villemeur, 1988] et repris dans [Zwingelstein, 1995], comme par exemple :
– un fonctionnement prématuré (ou intempestif)
– l'absence de fonction à la sollicitation
– la perte soudaine de la fonction
– le maintien de la fonction sur ordre d'arrêt
– etc.

Sous-système fonctionnel

Pratiquement, lorsque l'on cherche à analyser les risques sur un système complexe, i.e. pourvu d'un grand nombre de ressources et/ou de fonctions, on découpe ce système en sous-systèmes fonctionnels, de manière à ce que l'analyse fonctionnelle et l'analyse des risques qui en découle soit le plus clair possible, l'idée étant que d'autres analystes puissent consulter les documents sans trop s'y perdre.

Définition 8 (Sous-système fonctionnel)**.** Nous appelons ***sous-système fonctionnel***, un ensemble composé (Figure 2.7) :

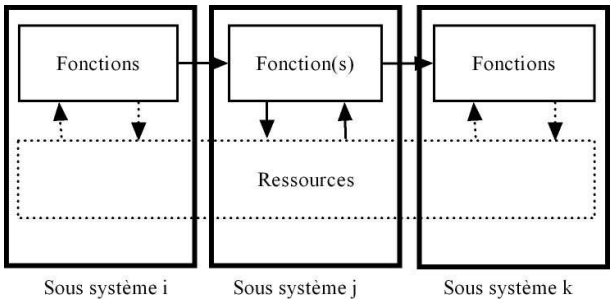


FIG. 2.7 – Découpage en sous-systèmes

– d'une ou plusieurs fonctions f_i
– de l'ensemble de leurs ressources $\bigcup R(f_i)$

Formellement, un sous-système fonctionnel Σ_k sera défini par le couple

$$\Sigma_k = \left(\{f_i\} \subset \Phi, \bigcup_i R(f_i) \right)$$

Conceptuellement, le découpage en sous-systèmes permet de représenter les parties du système qui jouent un rôle principal dans le fonctionnement du système global. Par exemple, si l'on considère un système de chauffage commandé à distance, on découpera un tel système en deux sous-systèmes : une partie chauffage et une partie commande.

Synthèse

Le modèle fonctionnel développé dans cette partie, ainsi que son formalisme, serviront de base à l'analyse AMDEC étendue proposée dans la partie suivante. Cette modélisation fonctionnelle repose en résumé :

– sur un découpage en fonctions du système
– sur les liens fonctions/ressources supports et fonctions/fonctions supports
– sur l'attribution d'une liste de modes possibles pour chaque fonction
– sur le regroupement de fonctions en sous-systèmes pour une meilleure compréhension du système

Pour affiner le modèle fonctionnel, on pourra se référer également à [Flaus et al., 2006] où la modélisation fonctionnelle proposée permet de cibler certaines fonctions particulières, comme les fonctions de sécurité, par exemple, en vue d'une analyse des risques.

2.3.2 Formalisation de l'analyse fonctionnelle

Une fois les bases de la modélisation fonctionnelle posées, nous proposons dans cette partie une méthode d'analyse des risques basé sur l'AMDEC [MIL-STD1629-A, 1983] avec l'ajout de

notions provenant de l'analyse par arbres de défaillances [Mortureux, 2002]. Cet ajout d'information provenant des arbres de défaillance permet d'intégrer les notions de simultanéité dans les relations de cause à effet dans le modèle fonctionnel que l'analyse AMDEC n'entreprend pas d'étudier.

Dans un premier temps, nous allons repréciser les notions de cause et d'effet introduites dans l'AMDEC originale pour les adapter au contexte de la modélisation fonctionnelle que nous avons proposée précédemment.
Cette partie propose donc un formalisme pour les notions de causes et d'effets développées par l'analyse AMDEC tout en intégrant cette notion de simultanéité.

Notion de cause

A l'origine, la notion de "cause" d'un mode de défaillance, selon l'analyse AMDEC originale, pouvait représenter n'importe quel phénomène, événement, défaut, défaillance, ou encore n'importe quel concept à l'origine d'un mode de défaillance. Pour affiner cette notion de cause, nous allons nous appuyer sur le formalisme proposé dans la section précédente de manière à définir les contours de cette notion de cause.
Dans le cadre du formalisme que nous présentons, nous imposons que la cause d'un mode de défaillance soit exprimée précisément en fonction des modes de défauts et des modes de défaillances des éléments du système.

Dans le paragraphe précédent, nous avons défini qu'une fonction f, pour se réaliser normalement, s'appuie :
- sur un certains nombre de ressources R_j
- sur un certains nombre de fonctions f_k, appelées aussi services.

En conséquence, un mode de défaut des ressources supports et/ou un mode de défaillance des fonctions supports sont susceptibles d'entraîner un mode de défaillance de la fonction dont elles sont le support.

En conclusion, les causes d'un mode de défaillance pourront donc être d'ordre :
- **structurel :** des modes de défaut de ressources $R_j \in R(f)$ supports à f
- **fonctionnel :** des modes de défaillance de fonctions $f_k \in SF(f)$ supports à f

Par ailleurs, le point à noter est que le mode de défaut d'une ressource ou le mode de défaillance d'une fonction support n'entraîne pas forcément un mode de défaillance. En effet, un mode de défaillance peut être causé par une combinaison de modes de défauts et de modes de défaillances.
Par exemple, considérons le mode de défaillance "Ne pas rouler droit" d'une voiture : le fait que la roue soit dans un mode de défaut "Sous gonflé" ne suffit pas à engendrer la crevaison, il faut également que la voiture soit dans le mode de défaillance "Rouler à grande vitesse".
Cette notion de combinaison sera donc intégrée dans l'analyse AMDEC étendue que nous pro-

posons ici.

Notion d'effet

De la même manière que la notion de cause, la notion d' "effet" d'un mode de défaillance pouvait représenter à l'origine n'importe quel concept étant la conséquence d'un mode de défaillance. Pour affiner cette notion d'effet, nous allons nous appuyer sur le formalisme proposé pour la modélisation fonctionnelle de manière à définir les contours de la notion d'effet.

Dans le paragraphe précédent, nous avons vu qu'une fonction pouvait servir de support à des fonctions ou services $f_j \in ACT(f)$.

Un mode de défaillance d'une fonction f peut entraîner :
- un mode de défaillance des fonctions $f_j \in ACT(f)$
- un mode de défaut de ressources $R_i \in R(f_j)$

Pour les mêmes raisons évoquées précédemment pour la notion de cause, si un mode de défaillance se produit, l'implication de mode de défaillance ou de mode de défaut en termes de conséquences n'est pas systématique ; il peut être nécessaire d'avoir une combinaison de modes de défaillance pour que les modes de défaillances ou modes de défaut résultants soient effectifs.

Soit $f \in \Phi$ une fonction du système. En conclusion, un mode de défaillance m_f pourra avoir des effets :
- **structurels** sur les ressources (celles qui sont les supports des fonctions dont f est support)
- **fonctionnels** sur les fonctions qui ont pour support la fonction f

Représentation graphique de l'analyse des risques

Pour permettre une meilleure vue d'ensemble des résultats de l'analyse AMDEC étendue, en particulier des combinaisons, nous proposons de représenter graphiquement ces résultats en reprenant la structure des arbres de défaillances ou arbres des causes (Figure 2.8). Une telle représentation permet d'avoir une vue d'ensemble des relations de causes à effets du système et rend l'étude de ces relations plus aisée que si elles étaient sous forme de tableau.
Etant donné que dans notre contexte, il n'y a pas d'événements indésirables étudiés (ou alors on peut considérer qu'il y en a plusieurs représentés sur un même graphique), le graphique obtenu n'est plus un arbre. On appellera donc cette représentation ***Graphe Causal de Dysfonctionnement (GCD)***.
Par ailleurs, les différentes combinaisons considérées dans ce graphe sont celles habituellement utilisées pour représenter un arbre de défaillance, à savoir [Clifton, 2005] :
- les portes ET : conjonctions de modes
- les portes OU : disjonctions de modes
- les portes NON : négation d'un mode

– les portes DELAI : délai temporel (fixé) d'un mode

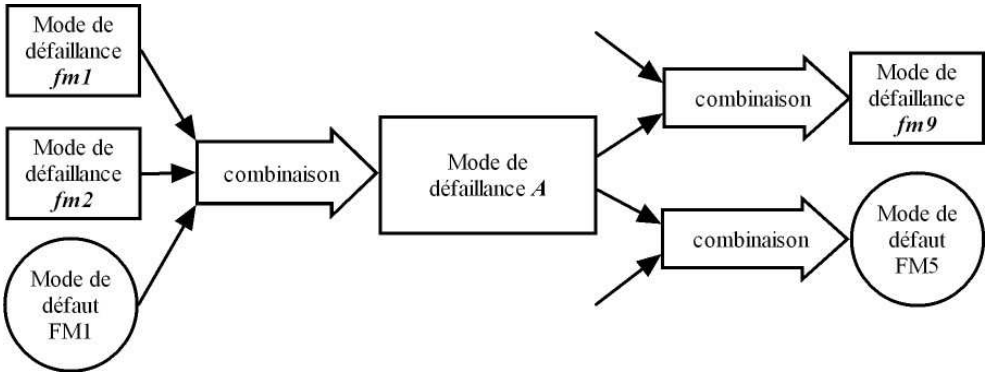


FIG. 2.8 – Représentation graphique des résultats de l'analyse AMDEC (tableau 2.1)

TAB. 2.1 – Tableau de l'analyse AMDEC représentée graphiquement en figure 2.8

Mode de défaillance	Causes	Effets
Mode de défaillance A	fm_1	fm_9
	fm_2	FM_5
	FM_1	

Graphiquement,
– un mode de défaut sera représenté par un cercle avec son intitulé à l'intérieur
– un mode de défaillance sera représenté par un rectangle avec son intitulé à l'intérieur

Concernant ce Graphe Causal de Dysfonctionnement, nous considérons qu'il est *non-cyclique*. En conséquence, il existe dans ce graphe des modes qui n'ont aucun antécédent (i.e. qui ne sont la conséquence d'aucun autre mode) : ces modes particuliers seront appelés *modes de base*.

A ce stade, les types de relations cause/effet suivantes ont été établies :
– combinaison (modes de défaillance) ⇒ mode de défaillance
– combinaison (modes de défaut) ⇒ mode de défaillance
– combinaison (modes de défaut, modes de défaillance) ⇒ mode de défaillance
– combinaison (modes de défaillance) ⇒ mode de défaut

Reste à étudier le type de relation manquant qui est la relation de cause à effet "combinaison (modes de défaut) ⇒ mode de défaut" et "combinaison (modes de défaut, modes de défaillance)

⇒ mode de défaut". Ces relations sont construites en effectuant une analyse des *flux de danger* du système entre les ressources.

Utilisation des flux de danger

Pour compléter les types de relations cause/effet introduites précédemment, il nous faut introduire la notion de cause à effet entre modes de défaut. Conceptuellement, cela signifie qu'une ressource dans un mode de défaut est capable d'engendrer par influence un mode de défaut d'une autre ressource.
Une telle relation cause/conséquence est importante à prendre en compte pour deux principales raisons :

- elle correspond aux dégradations générées par un élément lors de son passage en mode de défaut (appelée élément de scénario dans la méthode Scénarisk [Froquet, 2005])
- cette relation mode de défaut/mode de défaut n'existe pas à l'origine dans l'analyse AMDEC originale, concentrée sur l'analyse des causes et effets de modes de défaillance. La structuration du résultat de l'analyse AMDEC que nous avons entrepris fait apparaître ce point de façon naturelle.

Nous proposons donc de compléter l'analyse AMDEC originale en recherchant les relations de cause à effet entre modes de défaut. Pour cela, nous allons nous appuyer sur la recherche des flux de danger (section 1.3.2 en page 35) qui consiste à déterminer les impacts d'une ressource sur les autres ressources quand celle-ci est dans un mode de défaut (Figure 2.9).

Par exemple, considérons deux condensateurs C_1 et C_2, très proches l'un de l'autre, sur un circuit électronique. Un mode de défaut de C_1 pourrait être $FM_1(C_1)$ =*"Condensateur détruit"*. Cette explosion entraîne l'explosion du condensateur C_2, positionné à côté de C_1, qui se retrouve alors dans son mode de défaut $FM_1(C_2)$ =*"Condensateur explosé"*. Le flux de danger "explosion" a mis en évidence le lien de causalité entre les modes de défaut.

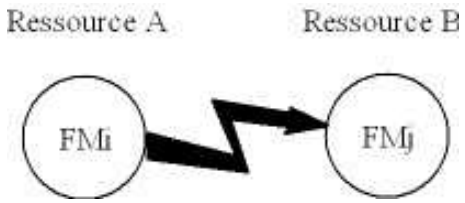


FIG. 2.9 – Flux de danger entre ressources

Tout comme les notions de cause et d'effet pour les modes de défaillance, l'implication de mode de défaut n'est pas systématique ; il peut être nécessaire d'avoir une combinaison de modes de défaut pour que les modes de défaut résultants soient effectifs.

Lien modèle fonctionnel/structurel

Par le biais de l'analyse AMDEC et par la modélisation fonctionnelle proposée dans cette section naît le lien entre le modèle structurel et fonctionnel (figure 2.10). Grâce à cette analyse, on peut mettre en évidence des relations de cause à effet :

- de fonction à ressource, un mode de défaillance entraînant un mode de défaut d'une ressource
- de ressource à fonction, un mode de défaut entraînant un mode de défaillance possible
- de fonction à fonction, un mode défaillance d'une fonction entraînant un mode de défaillance possible d'une autre fonction

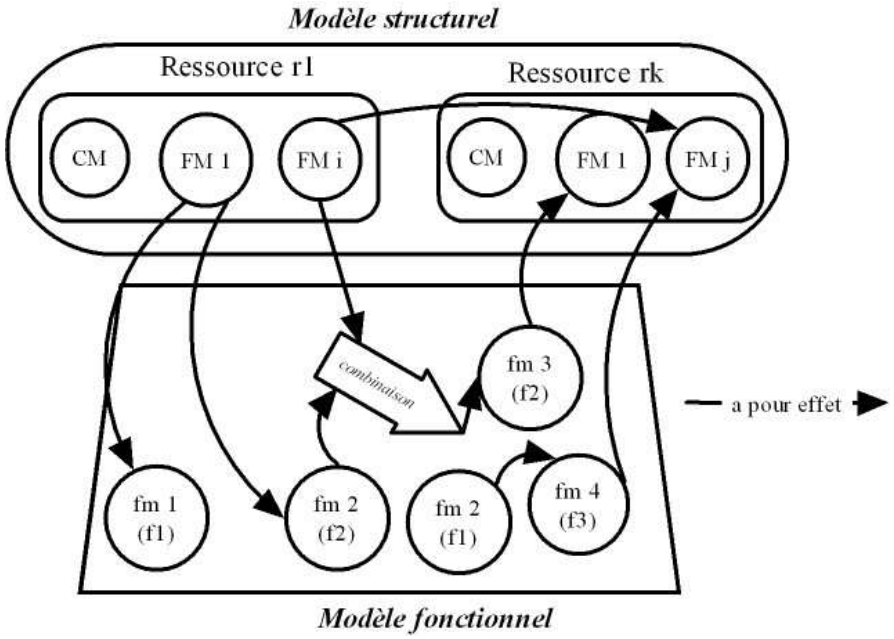


FIG. 2.10 – Lien entre le modèle structurel et fonctionnel

Conclusions

Grâce à l'extension de formalisme ainsi défini pour l'AMDEC à partir des notions de mode de défaut et de mode de défaillance, il est possible d'avoir une vue d'ensemble précise des relations de cause à effet du type :

- mode de défaillance à mode de défaillance
- mode de défaillance à mode de défaut
- mode de défaut à mode de défaillance
- mode de défaut à mode de défaut

Par ailleurs, l'ajout de combinaisons entre les différents modes a permit de prendre en compte la notion de simultaneité des modes dans les relations de cause à effet : d'où l'appellation d'analyse AMDEC ***étendue***.

2.3.3 Contraintes fonctionnelles

Dans la perspective d'intégrer les résultats de l'analyse AMDEC à l'analyse diagnostique, nous proposons dans cette partie de représenter le comportement d'une fonction à l'aide des contraintes qualitatives détaillées dans la section 2.2.2 en page 59. L'idée ici n'est pas d'effectuer un diagnostic fonctionnel, mais de valider pendant l'analyse diagnostique si l'altération des fonctions est cohérente ou non avec les diagnostics obtenus.

Construction des contraintes

Notation 5. On note $Fdom(f)$) l'ensemble des variables permettant la description du comportement de la fonction $f \in \Phi$.

Pour définir les contraintes représentant une fonction $f \in \Phi$, nous allons nous servir des contraintes de ses éléments supports, à savoir :
- les contraintes de ses ressources supports $R(f)$
- les contraintes de ses fonctions supports $SF(f)$

En effet, le fonctionnement de la fonction f est conditionné par le comportement de chacun de ses éléments supports. C'est donc leurs contraintes qui vont permettre de décrire le comportement de la fonction f.
Ainsi,

$$Fdom(f) \subseteq \left(\bigcup_i Rdom(r_i) \right) \cup \left(\bigcup_j Fdom(f_j) \right), r_i \in R(f), f_j \in SF(f)$$

A l'image des ressources, une contrainte est un tuple de $Fdom(f)$.

La construction des contraintes de la fonction f se déroule en 3 étapes :

1. la première étape consiste à construire l'ensemble des tuples possibles à partir des variables de $Fdom(f)$
2. l'étape suivante consiste à éliminer les contraintes contenant des jeux de valeurs "physiquement impossibles", autrement dits associées à des modes virtuels physiquement impossibles de fonctions.

3. enfin, dans cette dernière étape, à l'instar des ressources pour lesquelles on associe des modes de comportement aux contraintes, on associe également aux contraintes restantes les modes de fonctionnement correspondants.

Remarques :

- Pour pouvoir définir dans quel mode de fonctionnement se trouve une fonction, il est nécessaire de connaître les valeurs des variables impliquées dans les contraintes.
- La ou les contraintes où l'on reconnaît les tuples correspondant au bon comportement des éléments supports correspondent au mode de bon fonctionnement de la fonction
- Les contraintes où l'on reconnaît au moins un jeu de valeurs correspondant au mauvais comportement d'un des éléments supports correspondent à un mode de défaillance de la fonction
- Grâce à ces contraintes, il est possible de déterminer si un mode de défaillance est *détectable* ou non, la détection étant un des critères (à l'instar de la gravité et de l'occurrence) que l'on cherche à quantifier lors d'une analyse AMDEC

Grâce à cette modélisation comportementale et fonctionnelle, il nous sera possible de vérifier si un mode de défaillance, conséquence d'un diagnostic selon les résultats de l'analyse AMDEC, est cohérent ou non. En anticipant un peu sur le chapitre 3, on voit donc qu'un mode de défaillance incohérent avec un diagnostic révèle un diagnostic physiquement impossible. D'où l'affinage du diagnostic.

Exemple

On considère un système d'écoulement de l'eau de pluie composé d'un seul tuyau.
Le modèle comportemental du tuyau est donné en figure 2.11 ainsi que l'analyse fonctionnelle de ce système, où E_1 représente la présence d'eau $E_1 = Eau$ ou l'absence d'eau $E_1 = PasEau$ en amont du tuyau et où E_2 représente la présence d'eau $E_2 = Eau$ ou l'absence d'eau $E_2 = PasEau$ en sortie du tuyau.

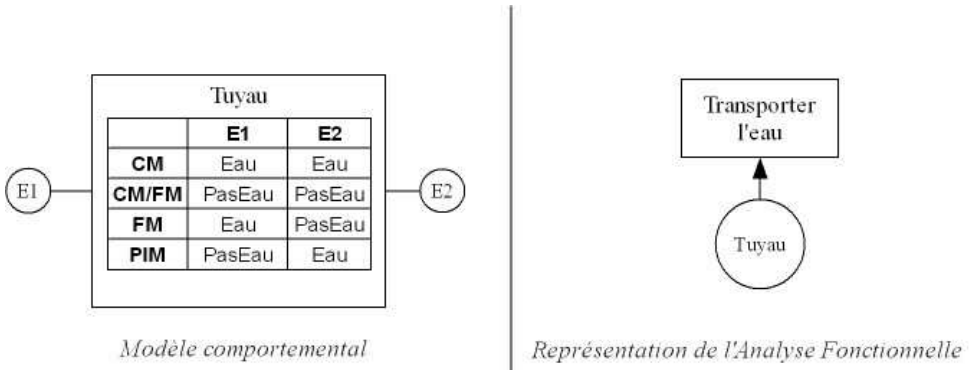


FIG. 2.11 – Modèle comportemental et analyse fonctionnelle du tuyau

La fonction "Transporter l'eau" s'appuie sur la ressource "tuyau". Dans ce cas, les contraintes fonctionnelles de cette fonction s'appuient sur le modèle comportemental du tuyau. On en déduit donc 3 contraintes fonctionnelles (figure 2.12) :

- $E_1 = Eau$ et $E_2 = Eau$ correspondant à la réalisation correcte de la fonction

- $E_1 = PasEau$ et $E_2 = PasEau$ correspondant au mode de défaillance fm_1 ="Ne pas transporter l'eau" ou au fonctionnement correct.
- $E_1 = Eau$ et $E_2 = PasEau$ correspondant au mode de défaillance fm_1 ="Ne pas transporter l'eau"

On remarque effectivement que :
- la contrainte liée au mode correct du tuyau est la seule contrainte correspondant au bon fonctionnement
- les contraintes contenant le mode de défaut du tuyau correspondant au mode de défaillance de la fonction

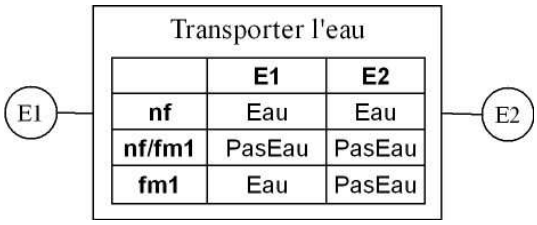


FIG. 2.12 – Contraintes de la fonction "Transporter l'eau"

2.3.4 Matrice de dysfonctionnement

La représentation graphique des relations de cause à effet pour un système complexe risque d'être difficile à lire et surtout à utiliser. Notre objectif étant d'utiliser de telles données, ces relations vont être synthétisées sous la forme d'une matrice, dite **matrice de dysfonctionnement**. Cette matrice, facilement manipulable informatiquement, nous permettra par la suite de développer une application logicielle.

Considérons un sous-système Σ_i contenant une seule fonction $f \in \Phi$, défini par :

$$\Sigma_i = (f \in \Phi, R(f) = \{r_1, \ldots, r_n\})$$

Les services (fonctions) dont f a besoin sont définis par $SF(f) = \{f_1, \ldots, f_p\}$.

On appelle matrice de dysfonctionnement (ou table de causes/effets) le tableau à double entrée :
- contenant en ligne :
  - ▶ les services nécessaires à f, $SF(f) = \{f_1, \ldots, f_p\}$ et leurs modes possibles respectifs
  - ▶ les ressources dont les modes de défaut sont les causes de modes de défauts des ressources de f, $R(f) = \{r_1, \ldots, r_n\}$
  - ▶ tout mode *intermédiaire virtuel* nécessaire pour simplifier les combinaisons de modes (par exemple, une conjonction de disjonction)
- en colonne :

- ▶ la fonction f et ses modes possibles respectifs
- ▶ les ressources supports à f, $R(f) = \{r_1, \ldots, r_n\}$ et leurs modes possibles respectifs

Le tableau reprend donc les résultats de l'analyse AMDEC étendue et celle de l'analyse des flux de danger(Figure 2.13) :

- les causes sont en lignes
- les effets sont en colonnes

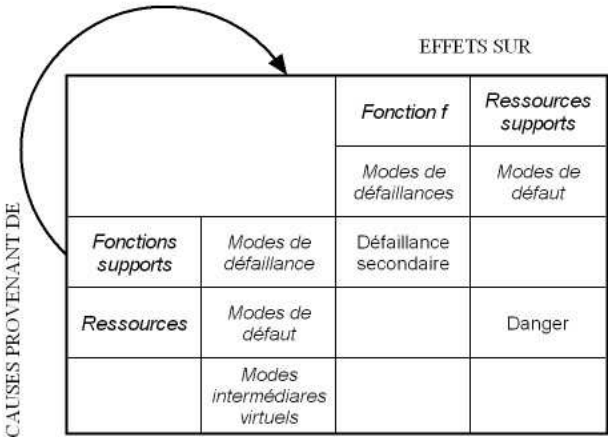


FIG. 2.13 – Synthèse de la matrice de dysfonctionnement

Parmi ces relations de causes à effets, on peut distinguer :

- celles qui conduisent à des *modes de défaillances secondaires* : ce sont des modes de défaillance de la fonction f provoquées par des modes de défaillance des fonctions supports à f, $SF(f)$
- celles qui conduisent à des *dangers* : ce sont des modes de défaut des ressources provoqués par d'autres ressources servant également de support à f

Notation 6. La matrice de dysfonctionnement se remplit ainsi (Figure 2.14) :

- si un effet j a pour origine une seule cause i, on mettra un "+" dans la case (i, j)
- si un effet j a pour origine p causes disjointes, on mettra un "+" dans les lignes (causes) correspondantes dans la colonne j
- si un effet j a pour origine n causes conjointes, on mettra un "×" dans les lignes (causes) correspondantes dans la colonne j
- si un effet j a pour origine la négation d'une cause i, on mettra "-" dans la cellule (i, j). Cette notation peut se combiner avec les conjonctions et disjonctions ou il suffira d'ajouter un "-" (un tiret) devant le symbole située dans la cellule de la cause concernée.
- si un effet j a pour origine une cause i avec un délai τ, on mettra ce délai τ dans la case (i, j)

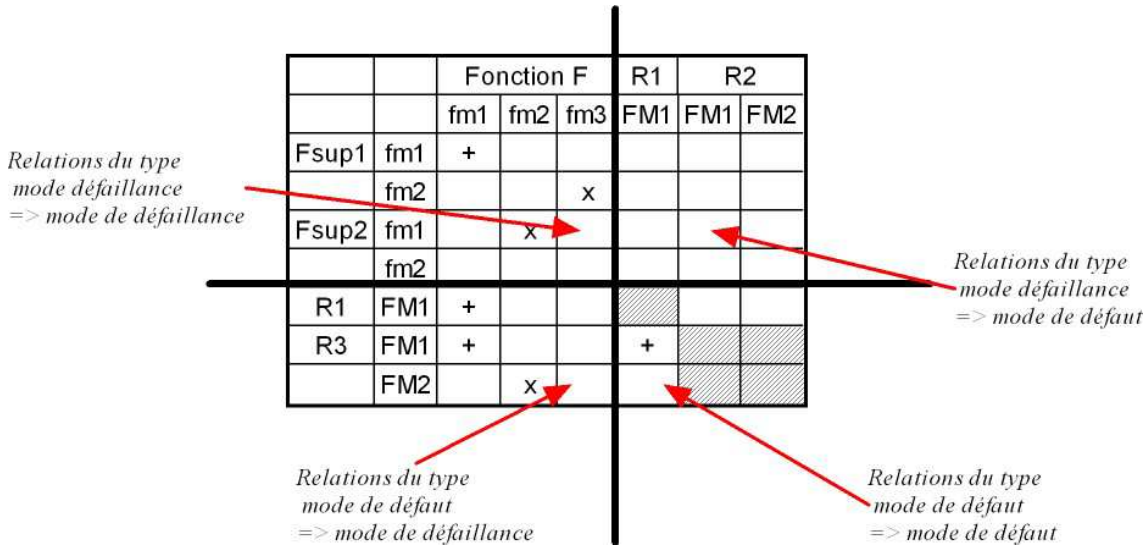


FIG. 2.14 – Matrice de dysfonctionnement issue de l'analyse AMDEC étendue

Avec cette matrice, il est donc possible de représenter entièrement le Graphe Causal de Dysfonctionnement, quelque soit la complexité des combinaisons en introduisant dans la matrice autant d'effets et de causes intermédiaires que nécessaire. De plus, le fait de devoir rechercher ces éléments intermédiaires et de les intégrer permet d'avoir une vue plus précise des enchaînements.

Synthèse

Cette représentation a plusieurs avantages :
- elle permet une vue d'ensemble des relations causes/effets pour faire en sorte d'être le plus complet possible
- elle est facilement implémentable dans une application logicielle
- elle est facile à compléter par des retours d'expérience ou en cas de modifications de l'installation, du procédé

2.3.5 Exemple

On considère le système suivant (Figure 2.15) représentant un rétroprojecteur.

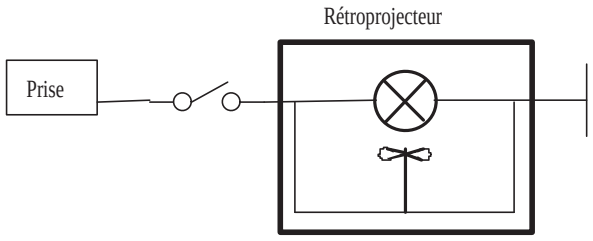


FIG. 2.15 – Rétroprojecteur simplifié

Le découpage structurel considéré est le suivant :
- La prise : $Prise$
- L'interrupteur : $Inter$
- L'ampoule : Amp
- Le ventilateur : $Vent$

Ce découpage est volontairement sommaire pour éviter de trop le complexifier : cet exemple a vocation d'illustration. Pour un exemple plus complexe, on pourra se reporter au chapitre 5 en page 141.

On a : $\Pi = \{Prise, Inter, Amp, Vent\}$.

Le découpage fonctionnel donne par exemple les fonctions suivantes :
- Alimenter en énergie (f_1)
- Eclairer (f_2)
- Ventiler (f_3)

Donc $\Phi = \{f_1, f_2, f_3\}$.

Le résultat de l'analyse fonctionnelle est représenté en figure 2.16 et se résume ainsi :
- la fonction "alimenter" s'appuie sur la prise et sur l'interrupteur et sert aux fonctions "ventiler" et "éclairer"
- la fonction ventiler a besoin de la fonction "alimenter" et s'appuie sur le ventilateur. Elle sert à la fonction "éclairer"
- la fonction "éclairer a besoin des fonctions "alimenter" et "ventiler" et s'appuie sur l'ampoule

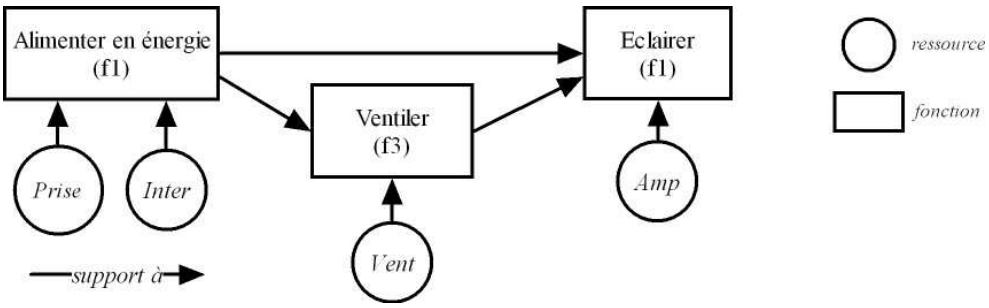


FIG. 2.16 – Représentation de l'analyse fonctionnelle du rétroprojecteur

Ainsi formellement :
- $SF(f_1) = \emptyset$, $R(f_1) = \{Prise, Inter\}$, $ACT(f_1) = \{f_2, f_3\}$
- $SF(f_2) = \{f_1\}$, $R(f_2) = \{Amp\}$, $ACT(f_2) = \emptyset$
- $SF(f_3) = \{f_1\}$, $R(f_3) = \{Vent\}$, $ACT(f_3) = \{f_2\}$

Enfin, on définit les modes de défaut de chacune des ressources (Tableau 2.2) et les modes de défaillance (Tableau 2.3) de chaque fonction qui vont être pris en compte lors de l'analyse AMDEC.

TAB. 2.2 – Modes de défaut des ressources du rétroprojecteur

Ressources	Libellé du mode de défaut	Notation
Prise	Connectique rompue	$FM_1(Prise)$
Interrupteur	Grippé en position OFF	$FM_1(Inter)$
	Grippé en position ON	$FM_2(Inter)$
Ventilateur	Balais usés	$FM_1(Vent)$
	Rupture du bobinage	$FM_2(Vent)$
	Balais coincés	$FM_3(Vent)$
Ampoule	Filament détruit	$FM_1(Amp)$
	Douille mal serrée	$FM_2(Amp)$

TAB. 2.3 – Modes de défaillance des fonctions du rétroprojecteur

Fonctions	Libellé du mode de défaillance	Notation
Alimenter (f_1)	Ne pas alimenter	$fm_1(f_1)$
	Surtension	$fm_2(f_1)$
Eclairer (f_2)	Ne pas éclairer	$fm_1(f_2)$
Ventiler (f_3)	Ne pas ventiler	$fm_1(f_3)$

L'analyse AMDEC conduit aux résultats représentés dans le tableau 2.4 qui sont graphiquement représentés dans la figure 2.17.

Fonctions	**Modes de défaillance**	**Causes**		**Effets**	
		Défaut	**Modes de défaillance**	**Défaut**	**Modes de défaillance**
Alimenter en énergie (f_1)	Ne pas alimenter $fm_1(f_1)$	Connectique rompue $FM_1(Prise)$			Ne pas éclairer $fm_1(f_2)$
Eclairer (f_2)	Ne pas éclairer $fm_1(f_2)$	Filament détruit $FM_1(Amp)$ Douille mal serrée $FM_2(Amp)$	Ne pas alimenter $fm_1(f_1)$		
Ventiler (f_3)	Ne pas ventiler $fm_1(f_3)$	Balais usés $FM_1(Vent)$ ou Rupture du bobinage $FM_2(Vent)$ ou Balais coincés $FM_3(Vent)$	Ne pas alimenter $fm_1(f_1)$	Filament détruit $FM_1(Amp)$	Ne pas éclairer $fm_1(f_2)$

TAB. 2.4 – Résultats de l'analyse AMDEC du rétroprojecteur

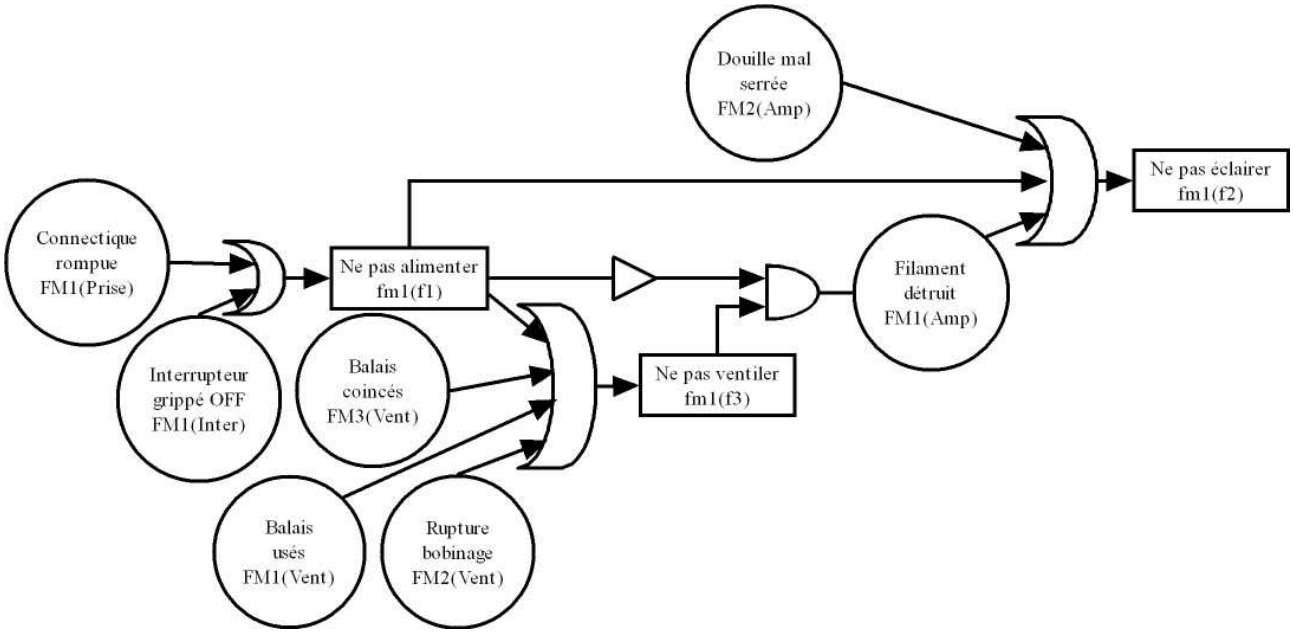


FIG. 2.17 – Représentation graphique d'un extrait des résultats de l'AMDEC du rétroprojecteur

Les combinaisons entre les modes sont représentés ici par des portes logiques à l'instar des arbres de défaillance. Par exemple, le mode de défaillance "ne pas ventiler" a pour cause soit le mode de défaillance "ne pas alimenter", soit le mode de défaut "balais coincés".

Ensuite, on établit les matrices de dysfonctionnement de chaque fonction :

– La fonction f_1 "Alimenter" (Figure 2.18) :

		Alimenter f1		***Prise***	***Inter***	
		fm1 Ne pas alimenter	fm2 Surtension	FM1 Connectique rompue	FM1 Grippé OFF	FM2 Grippé ON
Prise	FM1 Connectique rompue	+				
Inter	FM1 Grippé OFF	+				
	FM2 Grippé ON					

FIG. 2.18 – Matrice de dysfonctionnement de la fonction "Alimenter"

– La fonction f_2 "Eclairer" (Figure 2.19) :

		Eclairer f2	***Amp***	
		fm1 Ne pas éclairer	FM1 Filament détruit	FM2 Douille mal serrée
f1 Alimenter	fm1 Ne pas alimenter	+	-X	
f3 Ventiler	fm1 Ne pas ventiler	+	X	
Amp	FM1 Filament détruit	+		
	FM2 Douille mal serrée	+		

FIG. 2.19 – Matrice de dysfonctionnement de la fonction "Eclairer"

– La fonction f_3 "Ventiler" (Figure 2.20) :

		f3 Ventiler	***Vent***		
		fm1 Ne pas ventiler	FM1 Ventilateur coincé	FM2 Rupture du bobinage	FM3 Balais usés
f1 Alimenter	fm1 Ne pas alimenter	+			
Vent	FM1 Ventilateur coincé	+			
	FM2 Rupture du bobinage	+			
	FM3 Balais usés	+			

FIG. 2.20 – Matrice de dysfonctionnement de la fonction "Ventiler"

2.3.6 Conclusions

Dans ce chapitre, nous avons proposé un formalisme et une modélisation commune pour l'analyse des risques et le diagnostic. Pour cela, nous avons développé :

– un modèle structurel basé sur la notion de ressources pouvant être dans un certain nombre de modes possibles (mode normal, mode de défaut) et un mode virtuel physiquement possible

– un modèle comportemental basé sur une discrétisation des variables du système. Cette discrétisation nous a permis, en outre, de pouvoir définir un certain nombre de contraintes, qui, si elles sont vérifiées, assurent que la ressource est dans le mode considéré.

Le comportement d'une ressource reposant sur un certain nombre de variables, nous avons défini des contraintes de comportement comme étant des tuples de ces variables. En outre, chacun d'eux est associé à un ou plusieurs modes de comportement de la ressource. On retrouve alors l'idée développée par les approches de diagnostic où une contrainte associée à une ressource repose sur une ou plusieurs hypothèse(s) sur l'état de la ressource, à savoir ici son mode de comportement.

Par ailleurs, cette discrétisation des variables permet également :

- de représenter le comportement des ressources selon nos spécifications. Par exemple, en matière de prévention, les valeurs des variables discrètes peuvent représenter des seuils de sécurité
- de représenter le comportement des ressources qui n'ont pas de lois dynamiques, comme c'est le cas par exemple de l'interrupteur

– un modèle fonctionnel qui nous a permis d'étendre le formalisme de l'analyse AMDEC. Contrairement à l'analyse AMDEC originelle où derrière les notions de cause et d'effet de mode de défaillance étaient peu précises, nous avons choisi ici que seuls les modes de défaut et les modes de défaillance pouvaient être des causes et/ou des effets de mode de défaillance. De plus, nous avons ajouté des combinaisons de cause et d'effet pour en faire une analyse AMDEC étendue. Un formalisme sous forme de matrice de dysfonctionnement a aussi été développé.

Enfin, nous avons proposé une modélisation par contraintes du comportement des fonctions à l'image des ressources pour que les modes de défaillance puissent être validées ou non de manière à affiner les diagnostics.

A partir de cette représentation, le chapitre suivant développera une méthode intégrant les contraintes de bons comportements, de comportements physiquement impossibles ainsi que de mauvais comportements dans le but d'affiner les résultats de diagnostics. Par ailleurs, cette affinage se fera également en intégrant les résultats de l'analyse AMDEC en tant que connaissance experte lors de la procédure de diagnostic grâce à l'information provenant des relations de type cause/effet et des contraintes liées aux fonctions.

Chapitre 3

Intégration des résultats de l'AMDEC au diagnostic

3.1 Introduction

3.1.1 Problématique

Le principe du diagnostic est de donner une explication du comportement des ressources du système en fonction des valeurs relevées sur des capteurs et/ou provenant d'actionneurs. Ces explications sont obtenues en vérifiant les contraintes associées aux ressources. En utilisant les modèles que nous avons développés au chapitre 2, si une contrainte est vérifiée, alors l'explication est que la ressource est dans le mode particulier associé à cette contrainte.
Dans l'approche diagnostique dite DX, des modèles de bon comportement sont proposés, ainsi que des modèles de mauvais comportement [Struss & Dressler, 1989] [DeKleer & Williams, 1989] et des modèles de comportement physiquement impossible [Friedrich et al., 1990] de manière à affiner les résultats du diagnostic.

Dans ce chapitre, nous reprenons l'idée de modèles de bons et de mauvais comportements ainsi que la notion de comportement physiquement impossible que nous avons développés dans le chapitre 2 pour également affiner la recherche de diagnostics. Cependant, notre démarche présente un certain nombres de différences :

- nous proposons ici une méthode de raffinement de diagnostic basé sur une liste de diagnostics préalables. Il n'est donc pas nécessaire d'inclure les modèles de mauvais comportement et de comportement physiquement impossible lors de la recherche de diagnostics.
  Nous proposons donc une méthode d'affinage de diagnostic pouvant provenir de n'importe quelle analyse diagnostique préalable (DX ou FDI)
- par ailleurs, nous proposons d'intégrer les résultats de l'analyse AMDEC du système diagnostiqué en tant que connaissance experte supplémentaire dans le but :
  - d'éliminer les diagnostics impossibles en recherchant dans le Graphe Causal de Dysfonctionnement les modes de défaut et/ou les modes de défaillance incohérents

- d'améliorer le classement des diagnostics en localisant les modes de défauts primaires et secondaires dans chaque diagnostic

3.1.2 Méthode

Dans cette partie, une analyse diagnostique basée sur les 3 types de modélisation présentés au chapitre 2 (section 2.2.2, page 59) est proposée. En effet, ces 3 modèles permettent de distinguer les comportements possibles des comportements physiquement impossibles des ressources et, parmi les comportements physiquement possibles, de décrire également des comportements pour des types de défauts particuliers. Cette analyse diagnostique repose sur un algorithme basé sur une approche logique du diagnostic et étendue de manière à :

- éviter des diagnostics physiquement impossibles
- mieux localiser les défauts en déterminant les modes de défauts

Par ailleurs, pour affiner les résultats de l'analyse diagnostique, les résultats de l'analyse AMDEC seront intégrés lors de la phase d'affinage du diagnostic dans le but :

- d'aider à l'élimination des diagnostics physiquement impossibles
- d'améliorer le classement des diagnostics ; une analyse diagnostique conduit à une liste de défauts possibles qu'il est possible de classer du plus probable au moins probable. Nous proposons ici d'affiner cette hiérarchisation en utilisant les résultats de l'analyse AMDEC étendue.

3.2 Procédures de diagnostic

Hypothèse : Dans la suite de cette section nous supposons que les capteurs et les actionneurs sont parfaits, autrement dit que les valeurs réelles sont égales aux valeurs mesurées.

Nous proposons dans cette section un algorithme de diagnostic se déroulant en 4 étapes successives.
Dans un premier temps, une analyse diagnostique est réalisée, en considérant uniquement les modèles de bon comportement de chacune des ressources, pour que, comme expliqué dans le paragraphe précédent, n'importe quel résultat d'analyse diagnostique puisse être utilisé dans la suite.

Pour la recherche initiale de diagnostics, l'approche DX est privilégiée par rapport à l'approche FDI pour deux raisons :

- nous souhaitons déterminer s'il y a des défauts multiples, l'analyse de la table de signature des défauts stricto sensu selon la méthode FDI ne le permettant pas
- il n'y a pas d'exonération implicite dans la recherche de diagnostic dans l'approche DX, ce qui garantit une liste de diagnostic sans exonération.

Dans un deuxième temps, nous proposons d'affiner les diagnostics obtenus de manière à éliminer les diagnostics physiquement impossibles (s'il y en a) et préciser les modes de défauts des ressources a priori en défaut.
Il paraît alors logique de commencer par éliminer les diagnostics physiquement impossibles pour limiter la complexité de la recherche des modes de défaut.

Ainsi, on distinguera les étapes suivantes lors de l'analyse diagnostique :

- l'étape 1 qui consiste en la recherche de ressources a priori défaillantes en utilisant l'approche du diagnostic logique DX et la méthode de l'arbre HS de Reiter [Reiter, 1987]

- l'étape 2 qui consiste à compléter éventuellement les diagnostics minimaux en utilisant la connaissance experte issue de l'analyse AMDEC du système

- l'étape 3 qui consiste à exclure les diagnostics physiquement impossibles parmi la liste des diagnostics précédemment établie.
  Cet affinement est réalisé :
  - en utilisant les résultats de l'analyse AMDEC, plus précisément les relations de type cause/effet
  - en utilisant la propagation des valeurs issues des capteurs ou imposées par des actionneurs dans le modèle de comportements physiquement possibles, à l'instar de l'approche CSP (Constraint Satisfaction Problem) [Brand, 2004] [Fron, 1994]

- enfin, l'étape 4 qui utilise les modèles des modes de défauts pour isoler déterminer parmi la liste des diagnostics issus de l'étape précédente, leur mode de défaut correspondant.
  Dans le même temps, en utilisant les résultats de l'analyse AMDEC, les modes de défaut secondaires résultant d'un mode de défaut primaire sont recherchés de manière à améliorer le classement des diagnostics. En effet, un mode de défaut secondaire n'est que la conséquence d'un défaut primaire.

Dans les étapes 2, 3 et 4 présentées ci-dessus, les résultats de l'analyse AMDEC sous la forme de Graphe Causal de Dysfonctionnement sont utilisés. Etant donné que le modèle comportemental de notre système est purement statique, l'aspect temporel du Graphe Causal de Dysfonctionnement (donné en particulier par la porte DELAI) ne peut pas être pris en compte dans ce chapitre. On considérera donc dans la suite que les Graphes Causaux de Dysfonctionnement sont atemporels et donc que les relations de cause à effet sont instantanées.

3.2.1 Étape 1 : Détection des ressources en défaut

Pour cette phase de détection, nous proposons d'utiliser la recherche de tests [Staroswiecki et al., 2000] par exemple à l'aide de graphes bipartis à partir des modèles de bons comportements. Ce choix s'explique par le fait que, dans la suite, nous allons propager les valeurs des variables mesurées dans le modèle comportemental. Cette propagation risque d'être très complexe si ses contours ne sont pas définis. Ainsi, pour limiter la complexité de propagation

des valeurs, cette propagation se fera en parcourant les ressources impliquées dans chaque test.

On notera $\Delta = \{\delta_1, \ldots, \delta_p\}$ l'ensemble des tests établis pour l'analyse diagnostique, et $R(\delta_j)$ l'ensemble des ressources impliquées dans le test δ_j.

Ensuite, à partir des variables mesurées issues des capteurs et imposées par des actionneurs, ces tests sont évalués (vrai ou faux) et enfin, en utilisant la méthode de l'arbre HS de Reiter [Reiter, 1987], un ensemble de diagnosics, noté D est établi. Cet ensemble contient une liste des ressources en défaut expliquant les valeurs des variables mesurées.
Formellement, un ensemble de diagnostics minimaux sera noté $D = \{d_1, \ldots, d_n\}$ où les d_i sont les diagnostics, avec $d_i = \{r_f, \ldots, r_k\}$ l'ensemble des ressources suspectées.

Définition 9 (Degré de multiplicité d'un diagnostic)**.** Soit $d \in D$ un diagnostic.
On a $d = \{r_x, \ldots, r_z\}$.
On appelle degré de multiplicité de d la dimension de l'ensemble d, $card(d)$.

Les diagnostics obtenus peuvent être classés selon leur *degré de multiplicité*. Lors de la recherche des défauts secondaires (étape 4), le classement des diagnostics sera amélioré en modifiant le degré de multiplicité des diagnostics.

Une fois ces diagnostics obtenus, nous proposons dans la suite une procédure permettant de chercher si certains d'entre eux sont physiquement impossibles, et, dans ce cas, les supprimer de la liste des diagnostics.

Remarque :
Cette étape n'est pas détaillée car notre analyse s'appuie sur une liste de diagnostics déjà établis. Nous ne proposons pas de méthodes de détection, mais une méthode d'affinage de diagnostics à partir d'une liste de diagnostics préétablie.

3.2.2 Étape 2 : Complétion des diagnostics

Dans cette étape, nous considérons la connaissance experte apportée par l'analyse AMDEC du système. Le modèle comportemental du système peut ne pas contenir tous les modes de défaut possibles de chaque ressource, provenant par exemple d'une discrétisation trop "grossière" ou tout simplement parce que la connaissance pour modéliser ces modes de défaut n'est pas disponible.
Dans ce cas, il se peut que certains diagnostics établis à l'étape 1 soient incomplets. Pour les compléter, on recherche les modes de défaut conséquents ***valides*** (au sens booléen) de ces diagnostics dans le Graphe Causal de Dysfonctionnement.

Dans ce cas, les conséquences sont ajoutées au diagnostic étudié

Discussion

Cette étape n'est possible que si la connaissance apportée par l'analyse AMDEC du système est supplémentaire au modèle comportemental. En effet, si les modes de défauts présents dans les modèles fonctionnel et comportemental, cette étape n'aurait pas lieu d'être puisque les diagnostics minimaux seraient par définition suffisants pour expliquer les observations.

3.2.3 Étape 3 : Elimination des diagnostics physiquement impossibles

Parmi la liste des diagnostics établis précédemment, il se peut que certains de ces diagnostics soient impossibles. Dans [Friedrich et al., 1990], les modèles de comportement physiquement impossibles sont des jeux de valeurs n'ayant aucun sens physique. Ces modèles sont directement pris en compte lors de la recherche de diagnostics selon l'approche DX.

Nous proposons ici d'utiliser les modèles de comportements physiquement impossibles *a posteriori* pour éliminer les diagnostics physiquement impossibles provenant d'une analyse diagnostique non uniquement issue d'une approche DX (i.e. l'utilisation d'un solveur).
Par ailleurs, pour compléter l'approche de *Friedrich* dans laquelle des jeux de valeurs permettent d'identifier les comportements physiquement impossibles (ce que nous faisons également), nous proposons d'utiliser les résultats de l'analyse AMDEC pour rechercher les modes de défaillance conséquents aux diagnostics pour vérifier s'ils sont cohérents ou non.

Ainsi, l'élimination des diagnostics physiquement impossibles est réalisé en deux temps :

1. Dans un premier temps, les résultats de l'analyse AMDEC sont utilisés au titre de connaissance experte sur le système : pour chaque diagnostic, on recherche les modes de défaillance conséquents ***valides*** (au sens booléen) de ces diagnostics dans le Graphe Causal de Dysfonctionnement.
   On analyse ensuite les contraintes fonctionnelles liées aux modes de défaillance conséquents et on cherche à déterminer si ces contraintes sont vérifiées ou non selon les données disponibles par les capteurs et les actionneurs, supposés parfaits.
   SI un mode de défaillance conséquent n'est pas validé, alors le diagnostic est impossible.

   Dans ce cas, le diagnostic impossible est supprimé

2. Dans un deuxième temps, en utilisant le modèle de comportements physiquement possibles, chaque diagnostic est vérifié de manière à déterminer s'il est physiquement possible ou non. Cette vérification est réalisée en propageant les valeurs des variables mesurées et contrôlées dans le modèle de comportements physiquement possibles en réutilisant le "parcours" indiqué par les tests réalisés lors l'étape 1 (provenant de la recherche de RRAs par exemple).
   Nous proposons d'effectuer cette propagation en partant des contraintes contenant des variables mesurées ; cela permet de réduire immédiatement la complexité de la propagation.

L'algorithme permettant de réaliser l'étape 2 et 3 est donné ci-dessous et est représenté graphiquement par la figure 3.1. Les entrées de l'algorithme sont les suivantes :

- Un ensemble de tests, noté Δ
- Un ensemble de diagnostics à tester, noté D
- Un ensemble de valeurs connues noté V, issues des capteurs et des actionneurs
- Le Graphe Causal de Dysfonctionnement du système noté GCD, reprenant les résultats de l'AMDEC étendue du système diagnostiqué

Algorithme d'élimination des diagnostics physiquement impossibles

`Procedure` EliminationDiagPhysImp(TestSet Δ, DiagSet D, KnownValues[] V, Graph GCD)

Pour Chaque élément d de D du plus grand degré de multiplicité au plus faible **Faire**
Pour Chaque test δ de Δ **Faire**
Pour Chaque ressource r impliquée dans d **Faire**
Si les conséquences valides (en terme de mode de défaut) des défauts de r selon $GMDD$ ne sont pas dans d **Alors**
Ajouter les conséquences de d dans d
Si les conséquences valides (en terme de mode de défaillance) des défauts de r selon $GMDD$ ne sont pas cohérentes **Alors**
Supprimer d
Ensuite
Propager les valeurs des variables mesurées dans le modèle de comportement en faisant l'hypothèse de comportement physiquement impossible de r
Si les propagations ne conduisent qu'à des cas non absurdes **Alors**
Supprimer d
Recommencer avec un nouvel élément de D
FinSi
FinSi
FinChaque
FinChaque
FinChaque
FinProcédure

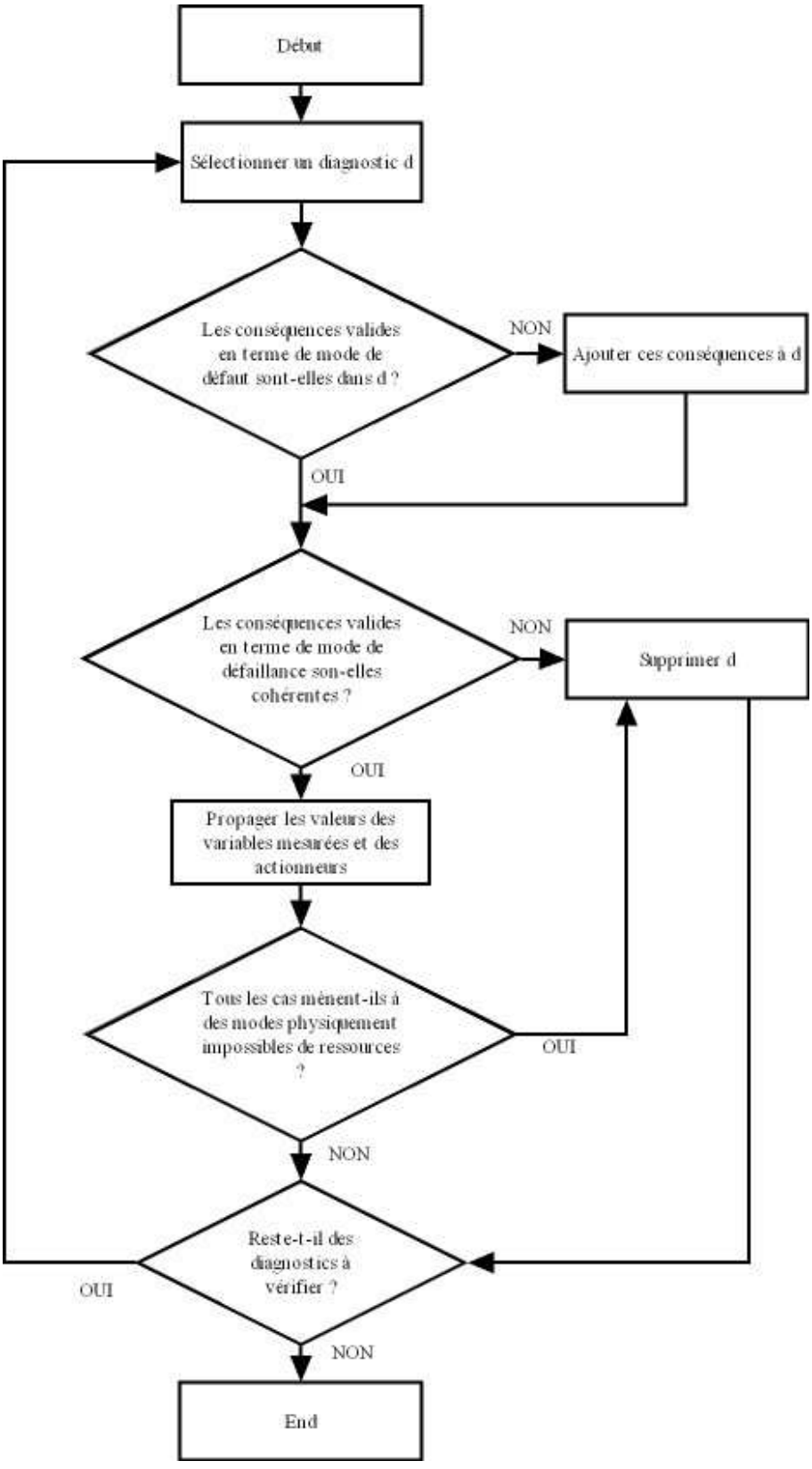


FIG. 3.1 – Algorithme d'élimination des diagnostics physiquement impossibles pour un test donné

Principe de l'algorithme / Discussion

L'algorithme analyse chaque diagnostic pour à la fois le compléter et vérifier s'il est physiquement possible ou non. Nous avons choisi de traiter les diagnostics par ordre de multiplicité, mais un quelconque ordre, même aléatoire serait possible.

Pour limiter la complexité de la propagation de valeurs dans le modèle de comportement, nous avons choisi de ne propager les valeurs qu'en utilisant les ressources impliquées dans les tests, et ce, pour chaque test. Par ailleurs, nous avons choisi de débuter la propagation en utilisant les variables mesurées. Bien sûr, il est également possible d'effectuer la propagation sans suivre le "chemin" indiqué par les tests, mais le fait de définir un "axe" de propagation permet de limiter la complexité.

L'hypothèse importante lors de la propagation est de considérer que seules les ressources impliquées dans les diagnostics testés peuvent se comporter de manière physiquement impossible, puisque c'est cela que l'on cherche à vérifier. En conséquence, lors de la propagation des valeurs, seules les contraintes associées aux modes de comportement physiquement possibles des diagnostics testés seront prises en compte et, d'autre part, seules les contraintes associées au bon et mauvais fonctionnement seront prises en compte pour les autres ressources. La complexité de l'algorithme se voit donc limiter une seconde fois.

3.2.4 Étape 4 : Identification des modes de défauts

Dans [Struss & Dressler, 1989] [DeKleer & Williams, 1989], des modèles de comportement spécifiques à des modes de défaut particuliers sont proposés. Ces modèles sont inclus directement lors de la recherche de diagnostics, au sens de l'approche DX du diagnostic, et sont pris en compte au même titre que les modèles de bon comportement. Ainsi, lors de la recherche de diagnostics, il est possible de vérifier immédiatement si un élément était dans un mode de défaut particulier.

Notre approche se base également sur des modèles de modes de défaut spécifique ; cependant, elle diffère en trois points :

- nos modèles de mauvais comportements sont utilisés *a posteriori*, après élimination des diagnostics physiquement impossibles. On réduit ainsi la complexité de recherche de modes de défaut particuliers en évitant le risque de rechercher des modes de défauts particuliers pour des diagnostics qui n'ont aucun sens physique
- nous affinons les degrés de multiplicité des diagnostics en recherchant les défauts secondaires. Les défauts primaires sont ceux qu'il faut avant tout réparer, car éliminer un défaut secondaire sans celui qui l'a engendré (i.e. le défaut primaire), c'est prendre le risque à coup sûr que ce défaut secondaire réapparaisse plus tard.

La méthode que nous proposons pour affiner encore une fois les diagnostics se déroule en 2 étapes :

1. dans un premier temps, nous proposons d'affiner le degré de multiplicité des diagnostics en recherchant si, parmi les diagnostics obtenus, certains de ses défauts ne sont que des *défauts secondaires* d'autres défauts présents dans le diagnostic. Cette recherche s'effectue en recherchant les causes de chacun des éléments impliqués dans le diagnostic dans le Graphe Causal de Dysfonctionnement.
   Si tel est le cas, nous proposons de réduire le degré de multiplicité du diagnostic d'autant que les défauts secondaires ont été repérés, considérant le fait que les défauts secondaires ne sont que des conséquences des défauts primaires.
   Ainsi un diagnostic avec un degré de multiplicité n comportant p défauts secondaires voit son degré de multiplicité devenir $n - p$.

2. dans un second temps, nous proposons de rechercher les modes de défaut de chaque ressource impliquée dans le diagnostic étudié en propageant les valeurs des variables mesurées dans les modèles de mode de défaut correspondants en supposant que chaque ressource autre que celle en défaut fonctionne correctement. Comme dans l'étape 3, cette propagation se fait pour chaque test pour limiter la complexité de propagation des valeurs.
   Cette hypothèse se justifie par le fait que si les autres ressources ne se comportaient pas correctement, alors le diagnostic testé serait incomplet, ce qu'il n'est pas puisque l'étape 2 complète les diagnostics incomplets. Cette vérification se fait dans l'ordre décroissant des degrés de multiplicité.

L'algorithme permettant de réaliser cette étape est donné ci-dessous et représenté en figure 3.2. Les entrées de l'algorithme sont les suivantes :

- Un ensemble de diagnostics à tester, noté D
- Un ensemble de valeurs connues noté V, issues des capteurs et des actionneurs
- Le Graphe Causal de Dysfonctionnement du système noté GCD, reprenant les résultats de l'AMDEC étendue du système diagnostiqué

Algorithme de localisation des modes de défaut pour un test donné

Procedure LocalisationModesDéfauts(DiagSet D, KnownValues[] V, Graph GCD)
Pour Chaque élément d de D du plus grand degré de multiplicité au plus faible **Faire**
Pour Chaque ressource r impliquée dans d **Faire**
Pour Chaque mode de défaut possible de r **Faire**
Propager les valeurs des variables impliquées dans le modèle de mauvais comportement en supposant les autres ressources comme se comportant correctement
Si les valeurs des mesures de capteurs et des actionneurs sont vérifiées **Alors**
Remplacer r par son mode de défaut m dans le diagnostic
RechercheDéfautSecondaires(m, GCD)
FinSi
FinChaque
FinChaque
FinChaque
FinProcédure

La procédure qui suit consiste à rechercher les défauts secondaires résultant de défauts primaires pour affiner les degrés de multiplicité des diagnostics. Elle est graphiquement représentée par la figure 3.3. Les entrées de l'algorithme sont les suivantes :

- Un mode de défaut à analyser, noté FM
- Le Graphe Causal de Dysfonctionnement du système noté GCD, reprenant les résultats de l'AMDEC étendue du système diagnostiqué

Algorithme de recherche des modes de défaut secondaires

Procedure RechercheDéfautsSecondaires(FaultMode FM, Graph GCD)
Propager le mode de défaut FM à travers le graphe GCD
Pour Chaque modes de défauts m conséquents valides et leur ressource associée r **Faire**
Si le mode de défaut m ou la ressource r est impliquée dans d **Alors**
Diminuer le degré de multiplicité de 1
FinSi
FinChaque
FinProcédure

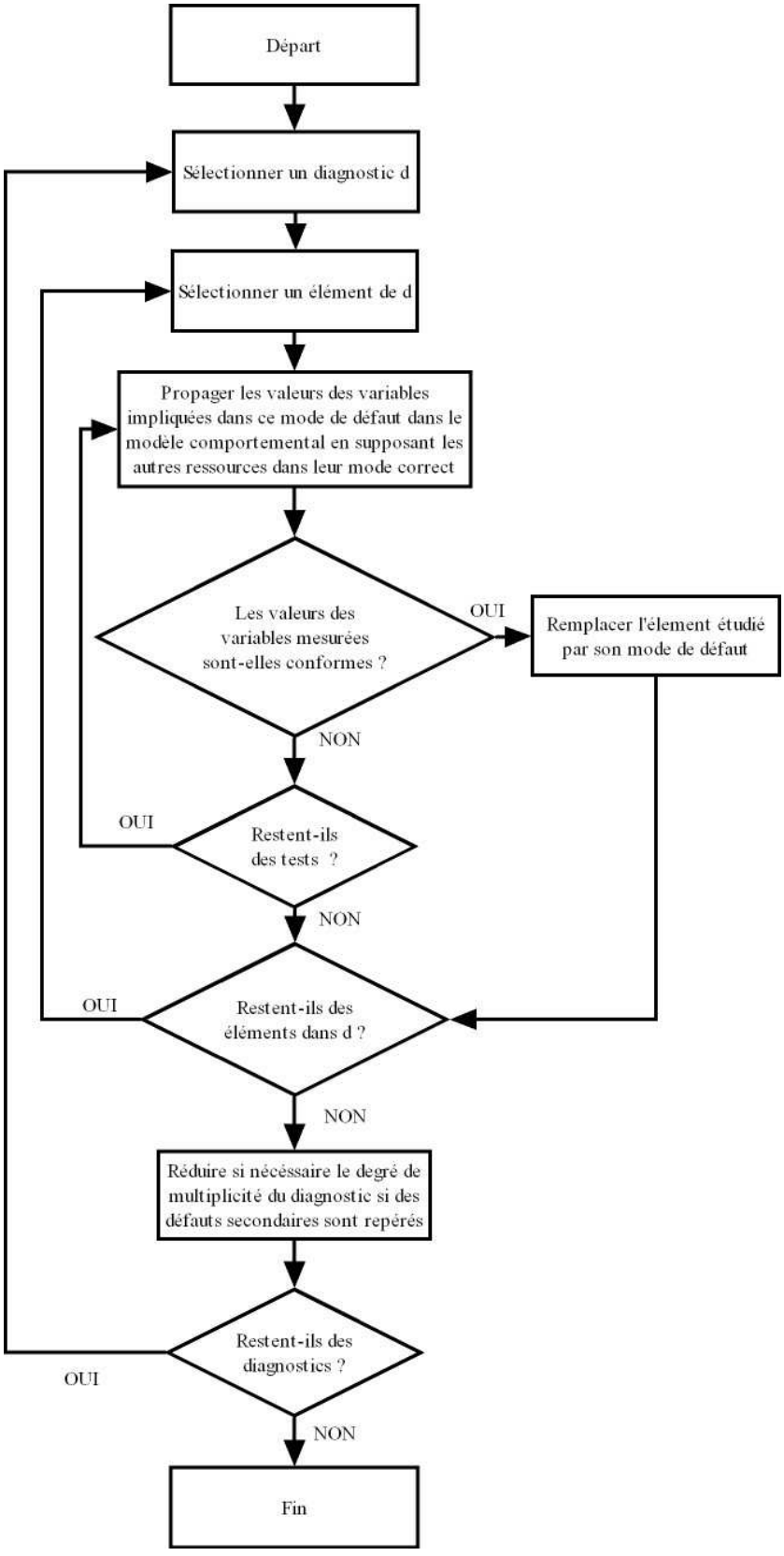


FIG. 3.2 – Algorithme de localisation des modes de défaut

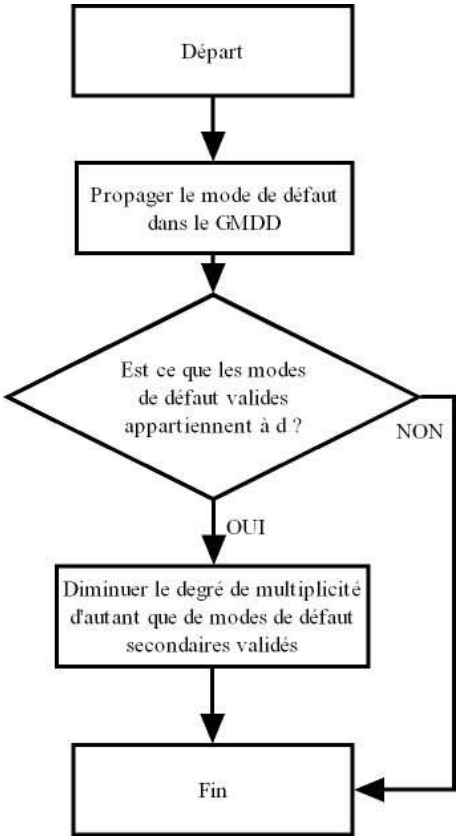


FIG. 3.3 – Algorithme de recherche des modes de défaut secondaires

Principe de l'algorithme / Discussion

L'algorithme analyse chaque diagnostic pour déterminer les modes de défauts de chacune des ressources impliquées dans le diagnostic. Nous avons choisi de traiter les diagnostic par ordre de multiplicité, mais un quelconque ordre, même aléatoire était possible.

Pour limiter la complexité de la propagation de valeurs dans le modèle de comportement, nous avons choisi de ne faire propager les valeurs qu'en utilisant les ressources impliquées dans les tests, et ce, pour chaque test. Par ailleurs, nous avons choisi de débuter la propagation en utilisant les variables mesurées. Bien sûr, il est également possible d'effectuer la propagation sans suivre le "chemin" indiqué par les tests, mais le fait de définir un "axe" de propagation permet de canaliser la complexité.

L'hypothèse importante lors de la propagation est de considérer que seuls les diagnostics testés peuvent avoir un mauvais comportement, puisque c'est cela que l'on cherche à vérifier. En conséquence, lors de la propagation des valeurs, seules les contraintes associées aux modes de défaut des diagnostics testés seront prises en compte et, d'autre part, seules les

contraintes associées au bon comportement seront prises en compte pour les autres ressources. La complexité de l'algorithme se voit donc limitée.

3.2.5 Exemple

Considérons à nouveau le rétroprojecteur du chapitre précédent représenté en figure 3.4.

On rappelle la liste des ressources issue du découpage structurel :
- La prise : $Prise$
- L'interrupteur : $Inter$
- L'ampoule : Amp
- Le ventilateur : $Vent$

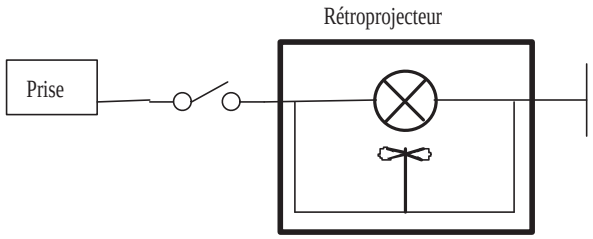


FIG. 3.4 – Schéma simplifié d'un rétroprojecteur

On considère les variables suivantes du système :
- P, caractérisant l'insertion de la prise de courant, avec $P = IN$ si la prise est insérée et $P = OUT$ si la prise ne l'est pas.
- U, représentant la présence ou non de tension entre l'alimentation et l'interrupteur, avec $U = 0$ en cas d'absence de tension et $U = 1$ en cas de présence
- A, représentant la position de l'interrupteur, avec $A = OFF$ si l'interrupteur est en position OFF et $A = ON$ s'il est en position ON
- I, représentant la présence ou non de courant entre la sortie de l'interrupteur et l'ampoule et entre l'interrupteur et le ventilateur, avec $I = 0$ en cas d'absence de courant et $I = 1$ en cas de présence
- L, représentant la présence ou non de lumière provenant de l'ampoule, avec $L = 0$ en cas d'absence de lumière et $L = 1$ en cas de présence
- V, représentant la présence ou non de ventilation (air propulsée) provenant du ventilateur, avec $V = 0$ en cas d'absence de lumière et $V = 1$ en cas de présence

Formellement, on a :
- $Cdom(Prise) = \{P, U\}$
- $Cdom(Inter) = \{U, I, A\}$
- $Cdom(Amp) = \{I, L\}$

– $Cdom(Vent) = \{I, V\}$

On rappelle les significations des modes de défaut de chaque ressource données dans l'exemple du chapitre 2 (Tableau 3.1).

TAB. 3.1 – Modes de défaut des ressources du rétroprojecteur

Ressources	Libellé du mode de défaut	Notations
Prise	Connectique rompue	$FM_1(Prise)$
Interrupteur	Grippé en position OFF	$FM_1(Inter)$
	Grippé en position ON	$FM_2(Inter)$
Ventilateur	Balais usés	$FM_1(Vent)$
	Rupture du bobinage	$FM_2(Vent)$
	Balais coincés	$FM_3(Vent)$
Ampoule	Filament détruit	$FM_1(Amp)$
	Douille mal serrée	$FM_2(Amp)$

Comme au chapitre précédent, ce découpage est volontairement sommaire pour éviter de trop le complexifier. Pour un exemple plus complexe, on pourra se reporter au chapitre 5 en page 141.

Les contraintes liées au comportement de chaque ressource sont représentées en figure 3.5.

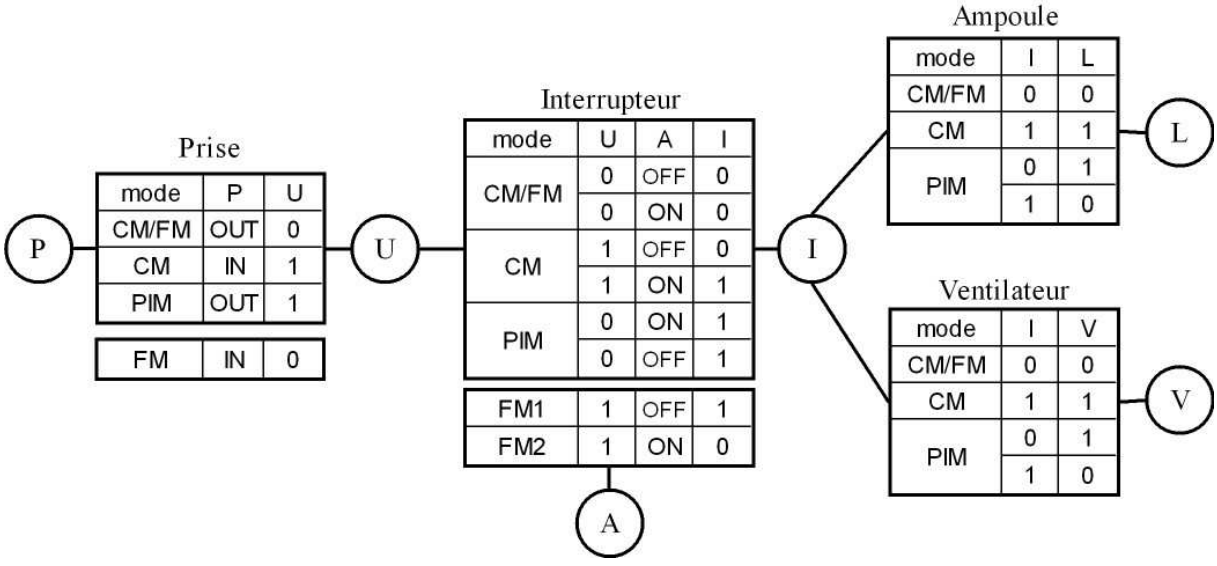


FIG. 3.5 – Représentation des contraintes de chaque ressource du rétroprojecteur

Pour terminer, les figures 3.7, 3.8 et 3.9 représentent respectivement les contraintes fonc-

tionnelles des fonctions f_1 (Alimenter), f_2 (Eclairer) et f_3 (Ventiler). Pour rappel, la figure 3.6 montre le résultat de l'analyse fonctionnelle du rétroprojecteur, analyse nécessaire pour la construction des contraintes fonctionnelles.

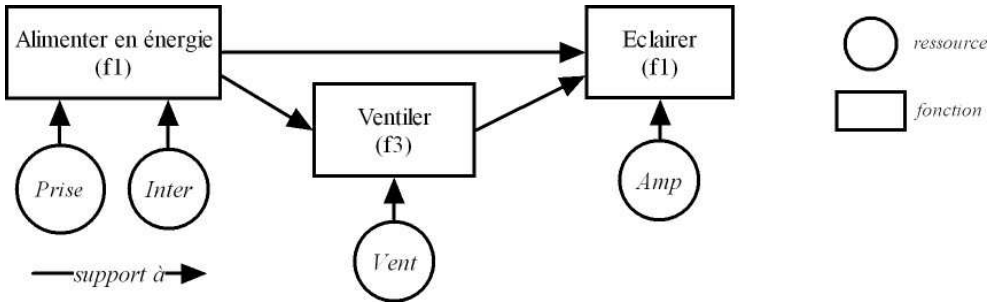


FIG. 3.6 – Représentation de l'analyse fonctionnelle du rétroprojecteur

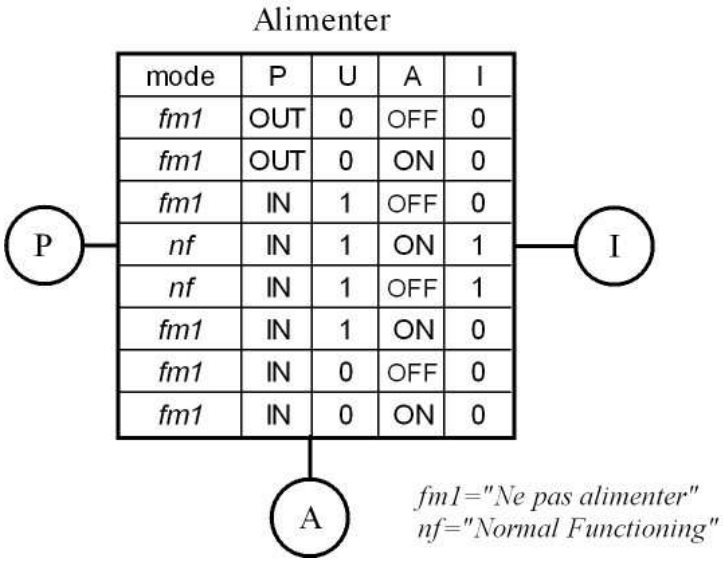
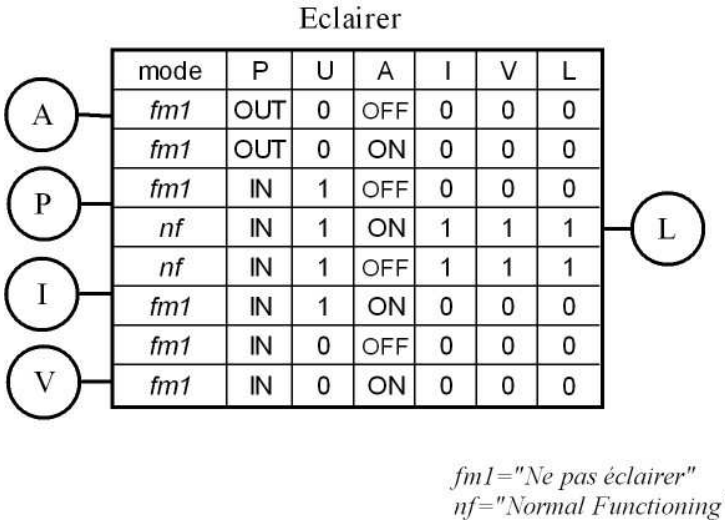
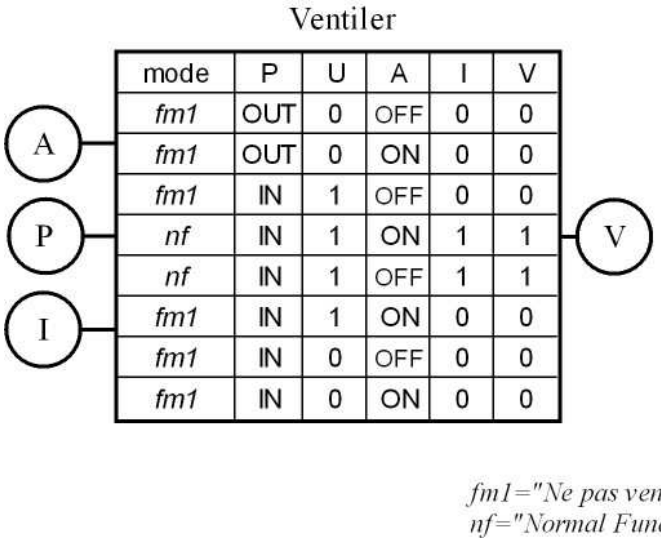


FIG. 3.7 – Contraintes fonctionnelles de la fonction f_1 "Alimenter"



mode	P	U	A	I	V	L
fm1	OUT	0	OFF	0	0	0
fm1	OUT	0	ON	0	0	0
fm1	IN	1	OFF	0	0	0
nf	IN	1	ON	1	1	1
nf	IN	1	OFF	1	1	1
fm1	IN	1	ON	0	0	0
fm1	IN	0	OFF	0	0	0
fm1	IN	0	ON	0	0	0

FIG. 3.8 – Contraintes fonctionnelles de la fonction f_2 "Eclairer"



mode	P	U	A	I	V
fm1	OUT	0	OFF	0	0
fm1	OUT	0	ON	0	0
fm1	IN	1	OFF	0	0
nf	IN	1	ON	1	1
nf	IN	1	OFF	1	1
fm1	IN	1	ON	0	0
fm1	IN	0	OFF	0	0
fm1	IN	0	ON	0	0

FIG. 3.9 – Contraintes fonctionnelles de la fonction f_3 "Ventiler"

On effectue la démarche suivante :
"Le rétroprojecteur est branché ($P = IN$) et allumé. On cherche à éteindre le rétroprojecteur ; pour cela on appuie sur l'interrupteur ($A = OFF$), mais le ventilateur tourne toujours ($V = 1$) et l'ampoule est toujours allumée ($L = 1$)".

Étape 1 : Détection des ressources en défaut

En utilisant les graphes bipartis [Staroswiecki et al., 2000] (Figure 3.10), 3 tests peuvent être construits à partir des contraintes précédentes (Fig. 3.2).

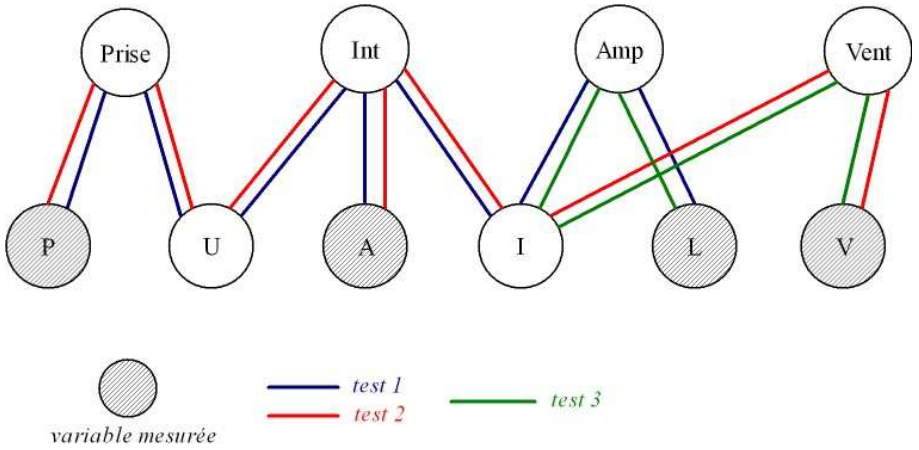


FIG. 3.10 – Recherche des tests par les graphes bipartis pour le rétroprojecteur

Parmi ces tests, les tests 1 et 2 sont faux alors que le test 3 est vrai.

	Prise	Inter	Amp	Vent
Test 1	x	x	x	
Test 2	x	x		x
Test 3			x	x

TAB. 3.2 – Tests établis en utilisant les graphes bipartis

Par suite, en utilisant la méthode de l'arbre HS [Reiter, 1987], on obtient l'ensemble des diagnostics minimaux suivants :

$$\{\{Prise\}, \{Inter\}, \{Amp, Vent\}\}$$

Étape 2 : Complétion des diagnostics

En propageant les modes de défaut possibles dans le Graphe Causal de Dysfonctionnement (figure 3.11) pour chacun des diagnostics, aucun mode de défaut conséquent n'est relevé. Les diagnostics sont donc tous complets.

Étape 3 : Elimination des diagnostics physiquement impossibles

Maintenant, en utilisant l'algorithme décrit dans la section précédente, les éventuels diagnostics physiquement impossibles vont être recherchés puis éliminés et les diagnostics incomplets complétés. Pour cela, deux méthodes sont à notre disposition :

- L'utilisation des résultats de l'AMDEC
- L'utilisation du modèle de comportements physiquement possibles par propagation des valeurs des variables mesurées

Utilisation des résultats de l'AMDEC

La figure 3.11 donne la représentation sous forme de Graphe Causal de Dysfonctionnement reprenant les résultats de l'AMDEC du rétroprojecteur.

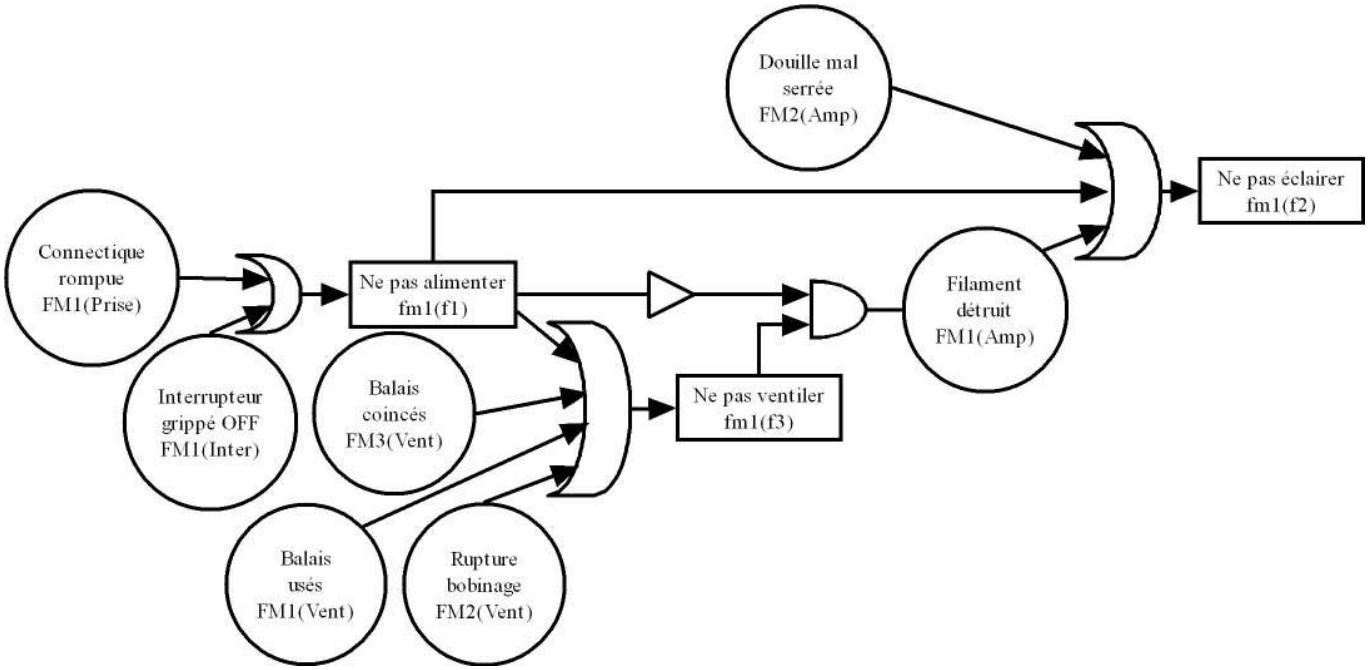


Fig. 3.11 – Représentation graphique des résultats de l'AMDEC du rétroprojecteur

Considérons le diagnostic suivant : $\{Prise\}$.

L'analyse du Graphe Causal de Dysfonctionnement nous montre que le mode de défaut de la prise $FM_1(Prise)$ "connectique rompue" entraîne le mode de défaillance $fm_1(f_2)$ "Ne pas éclairer". Or la lampe éclaire ($L = 1$). Ce qui est incohérent avec les contraintes fonctionnelles de la fonction "Eclairer" dont aucune associée à ce de mode de défaillance ne contient la valeur 1 pour la variable L (figure 3.8 en page 102).
Donc le mode de défaut $FM_1(Prise)$ n'est pas un diagnostic cohérent.

Considérons le diagnostic suivant : $\{Inter\}$.

Cette ressource possède deux modes de défauts, mais seul le mode de défaut FM_1 apparaît dans le Graphe Causal de Dysfonctionnement.
L'analyse du Graphe Causal de Dysfonctionnement nous montre que le mode de défaut de l'interrupteur $FM_1(Inter)$ "Grippé OFF" entraîne le mode de défaillance $fm_1(f_2)$ "Ne pas éclairer". Or la lampe éclaire par hypothèse ($L = 1$). Ce qui est incohérent avec les contraintes fonctionnelles de la fonction "Eclairer" dont aucune associée à ce de mode de défaillance ne contient la valeur 1 pour la variable L (figure 3.8 en page 102).

Cependant, comme il n'est pas possible de vérifier que $FM_2(Inter)$ est également incohérent, il n'est pas possible de conclure quant à l'incohérence de ce diagnostic.

Considérons le diagnostic suivant : $\{Amp, Vent\}$.

L'analyse du Graphe Causal de Dysfonctionnement nous montre que les modes de défaut du ventilateur $FM_i(Vent)$ entraînent le mode de défaut $FM_1(Amp)$ (présent dans la liste des diagnostics, donc ce diagnostic est complet) qui lui-même entraîne le mode de défaillance $fm_1(f_2)$ "Ne pas éclairer". Or la lampe éclaire par hypothèse ($L = 1$). Ce qui est incohérent avec les contraintes fonctionnelles de la fonction "Eclairer" dont aucune contrainte associée à ce de mode de défaillance ne contient la valeur 1 pour la variable L (figure 3.8 en page 102).
Donc ce diagnostic, bien que complet, est incohérent.

L'ensemble des diagnostics devient donc :

$$\{\{Inter\}\}$$

Utilisation du modèle de comportements physiquement possibles

Considérons maintenant le diagnostic restant $\{Inter\}$ et propageons les valeurs des variables mesurées pour déterminer si ce diagnostic est également physiquement impossible ou non. La figure 3.12 montre que si l'on considère l'interrupteur comme un diagnostic physiquement impossible, la prise doit être considérée comme un diagnostic également. Or le diagnostic $\{Inter\}$ est un singleton et ne contient pas l'élément $\{Prise\}$. Donc cette hypothèse est fausse.
$\{Inter\}$ est donc un diagnostic physiquement possible.

Étape 4 : Identification des modes de défauts

A cette étape, les modes de défaut de chaque diagnostic sont testés de manière à affiner la localisation des défauts. Le seul diagnostic restant est l'interrupteur $\{Inter\}$.

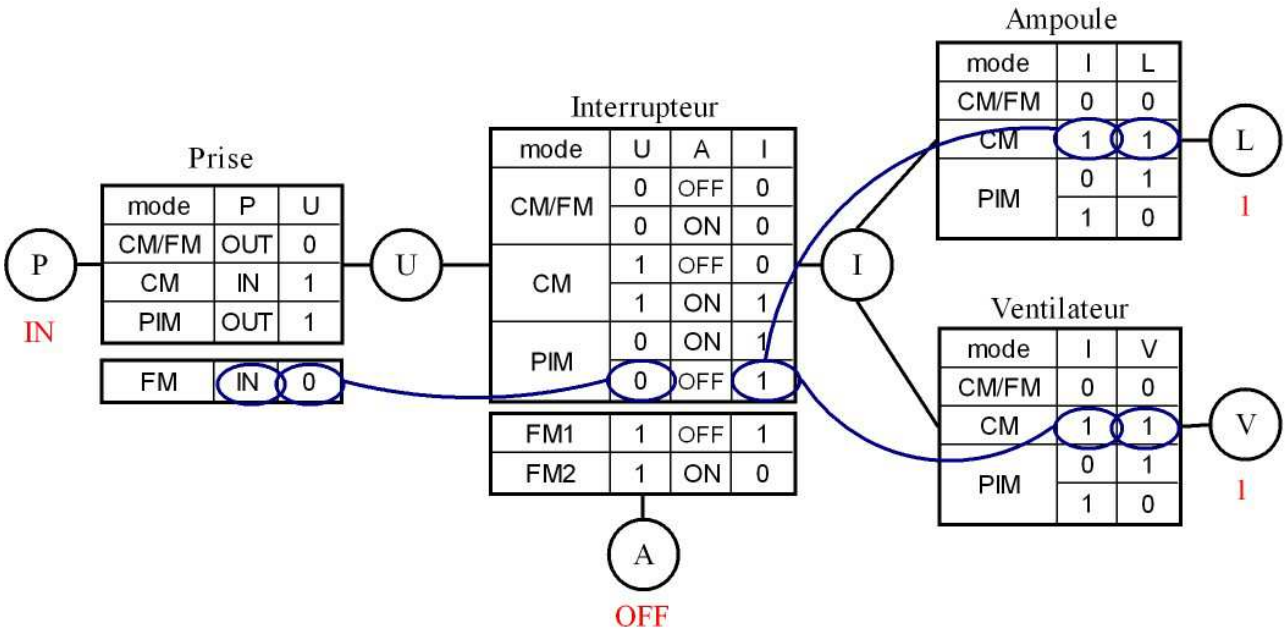


FIG. 3.12 – Représentation graphique de la propagation de valeurs dans l'étape 3 montrant que le diagnostic $\{inter\}$ n'est pas physiquement impossible

La propagation des valeurs dans les modèles de mauvais comportement (figure 3.13) nous conduit à conclure que c'est le mode de défaut $FM_1(Inter)$ qui explique les valeurs des variables mesurées. Explicitement, le mode de défaut de l'interrupteur est donc "Grippé en position ON".

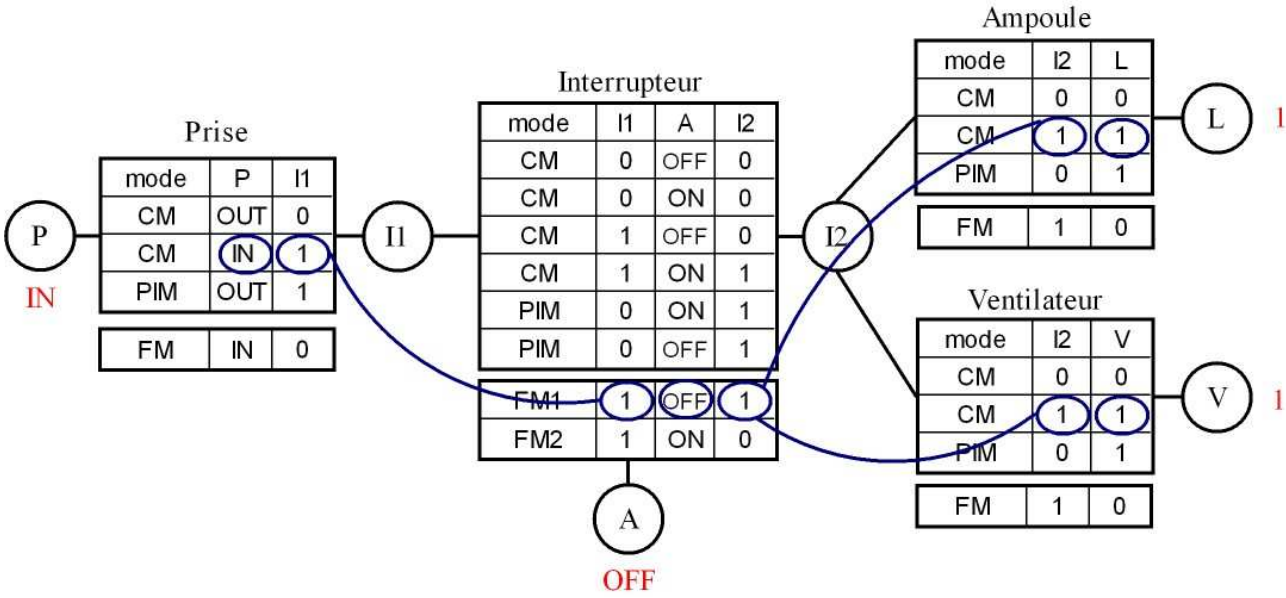


FIG. 3.13 – Représentation graphique de la propagation de valeurs dans l'étape 3

En conclusion, l'origine de la panne vient de l'interrupteur, avec comme mode de défaut "Grippé en position ON".
Grâce à la modélisation adoptée pour ce système, l'opérateur chargé de la maintenance connaît précisément l'origine du défaut, avec son mode de défaut particulier, alors qu'au départ, 3 diagnostics différents lui étaient proposés, sans modes de défaut ciblés.

3.2.6 Analyse des procédures de diagnostic et des résultats

Complexité des algorithmes

La complexité des algorithmes décrits dans les étapes 2 et 3 pourraient engendrer une explosion combinatoire lors de la propagation des valeurs dans les modèles de comportement.
Dans les exemples traités, il n'y a pas d'explosion combinatoire puisque lors de la propagation des valeurs, plusieurs paramètres restrictifs entrent en compte :

- les parcours effectués dans les différentes contraintes sont ceux indiqués par les tests établis dans l'étape 1 (Détection). Ainsi, pour un test incluant n ressources, ces mêmes n ressources seront traversées lors de la propagation des valeurs
- par ailleurs, les ressources traversées possèdent des hypothèses sur leurs modes. Ainsi :
  - dans l'étape 3 (Elimination des diagnostics physiquement impossibles), la ressource testée est supposée dans un mode physiquement impossible
  - dans l'étape 4 (Localisation des modes de défaut et affinage des degrés de multiplicité), la ressource testée est dans un mode de défaut particulier et les autres ressources supposées dans leur mode correct

Evaluation de la complexité :

Soient d diagnostics portant au plus sur r ressources et t tests.
Soit f le nombre maximum de contraintes liées au mauvais comportement pour une seule ressource.
Soit c le nombre maximum de contraintes liées au bon comportement pour une seule ressource.
Soit p le nombre maximum de contraintes liées au comportement physiquement impossible pour une seule ressource.

L'étape 3 (Elimination des diagnostics physiquement impossibles) comporte une boucle sur le nombre de diagnostics, une sur le nombre de tests et une sur chaque ressource impliquée dans le diagnostic étudié et teste chaque mode physiquement impossible de la ressource en propageant les valeurs des variables mesurées et des actionneurs, les autres ressources pouvant être dans n'importe quel mode.
La complexité de l'étape 2 est donc de $d \times t \times r \times p \times (f + c + p)$.

L'étape 4 (Localisation des modes de défaut et affinage des degrés de multiplicité) comporte une boucle sur le nombre de diagnostics, une sur le nombre de tests et une sur chaque ressource impliquée dans le diagnostic étudié et teste chaque mode de défaut de la ressource

en propageant les valeurs des variables mesurées et des actionneurs, les autres ressources étant supposées dans leur mode correct.
La complexité de l'étape 2 est donc de $d \times t \times r \times f \times c$.

Résultats des procédures

Grâce à aux différentes procédures proposées dans ce chapitre, il est possible d'éviter des diagnostics physiquement impossibles du fait de la prise en compte de ce mode lors de la recherche des défauts et grâce à l'intégration des résultats de l'AMDEC étendue dans ces procédures. Par ailleurs, les résultats de l'AMDEC nous ont également permis de compléter les diagnostics incomplets en recherchant les conséquences valides de chaque diagnostic en terme de mode de défaut dans le Graphe Causal de Dysfonctionnement.
Par ailleurs, la diversité des modes de défauts des ressources issus des modèles de mauvais comportement nous a permis de mieux localiser le type de défaut qui est à l'origine des valeurs des variables mesurées.

Pour appliquer ces algorithmes, une approche logicielle a été mise en oeuvre et détaillée en annexe (5.3) en page 181.

3.3 Tests de bon fonctionnement

L'utilisation de lois quantitatives ou quantitatives pour la modélisation du comportement d'un système permet uniquement de détecter des défauts si ces lois ne sont pas vérifiées. En effet, si ces mêmes lois sont validées, aucune conclusion ne peut être avancée quant au bon fonctionnement du système.
Dans cette partie, nous proposons de définir une méthode permettant de réaliser des tests de bon fonctionnement **pour une seule ressource**. Pouvoir réaliser de tels tests montre un certain intérêt :

- il est possible ainsi de vérifier immédiatement qu'une ressource n'est pas en défaut
- pour des ressources en position critique (i.e. vitale pour le bon fonctionnement du système, car, par exemple, non redondante) dans le système, il est possible de s'assurer qu'elles ne dysfonctionnent pas sans pour autant lancer toute la procédure de diagnostic qui peut être lourde et longue suivant la complexité du système

3.3.1 Principe

Tout comme la section précédente, nous faisons ici l'hypothèse du fonctionnement parfait des capteurs et des actionneurs.
L'impossibilité de conclure au bon fonctionnement d'une ressource provient de l'aspect quantitatif des contraintes associées aux ressources. Pour pouvoir conclure que la loi est vérifiée, il faudrait la vérifier pour toutes les valeurs possibles de chacune des variables, ce qui est impossible du fait de la densité de leur domaine de valeurs.

Or, dans le chapitre 2, nous avons proposé une modélisation qualitative des comportement des ressources, basée sur la discrétisation des variables et donc plusieurs contraintes peuvent être associées au bon comportement de la ressource. Du fait de cette discrétisation des variables, on peut affirmer que :

- si certaines de ces contraintes sont vérifiées, alors la ressource est dit vraisemblablement dans son mode correct à ces points de fonctionnement
- si toutes les contraintes sont vérifiées, alors la ressource est dans mode correct à tous les points de fonctionnement. Elle est donc compatible avec une hypothèse de "bon fonctionnement"

S'il est possible de stimuler le comportement d'une ressource à **différents points de fonctionnement**, on est alors en mesure de tester si une ressource fonctionne correctement ou non. En l'occurrence, vérifier tous les points de fonctionnement correspondant au mode correct permet de s'assurer que la ressource fonctionne correctement.

Soit $r \in \Pi$ une ressource et $Cons_{CM}(r) \subset Cons(r)$ l'ensemble des contraintes associées au mode correct CM de la ressource r.

- Si, $\forall k \in Cons_{CM}(r)$, k valide, alors la ressource est dite "en bon fonctionnement global"
- Si, $\exists H \subset Cons_{CM}(r), \forall k \in H$, k valide, alors la ressource est dite en en bon fonctionnement aux points de fonctionnement associées aux contraintes de H

3.3.2 Limites

Ce principe est difficilement intégrable aux algorithmes de diagnostic que nous avons proposés précédemment, du fait de qu'il est vraiment complexe de pouvoir agir sur les valeurs des variables lors de leur propagation dans le modèle.
Cependant, le fait de pouvoir tester le bon fonctionnement d'un composant dans les systèmes simples est possible car les systèmes simples sont composés de peu de composants ; un tel test est donc envisageable. Par exemple, les détecteurs de fumée possèdent leur propre test de bon fonctionnement.

3.3.3 Exemple

Prenons le cas de l'ampoule (extraite de l'exemple du rétroprojecteur), modélisée selon nos spécifications (figure 3.14), à savoir ici : "*nous voulons qu'elle éclaire*"

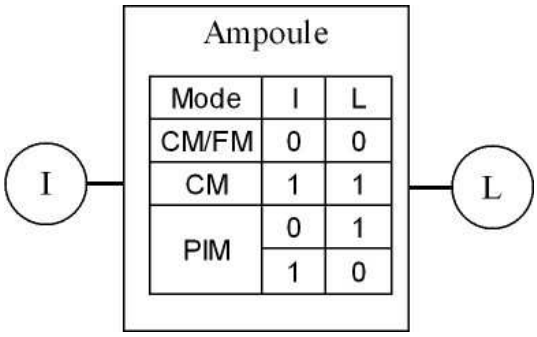


FIG. 3.14 – Contraintes de comportement de l'ampoule

En stimulant le système, on est capable de vérifier les points de fonctionnement correspondant au mode correct CM.
Pour cela, chacune des contraintes doit être vérifiée. Il faut qu'en stimulant le système (ici I) :

- En mettant I à 0, on observe que L vaille 0, autrement dit en ne mettant la lampe en présence d'aucun courant, la lumière ne doit pas apparaître
- En mettant I à 1, on observe que L vaille 1, autrement dit en mettant la lampe en présence de courant, on doit observer de la lumière

Si chaque contrainte est vérifiée, alors la lampe fonctionne correctement (répond à nos spécifications).
Ainsi, s'il faut s'assurer que certaines ressources du système ne dysfonctionnent pas, cette modélisation qualitative par discrétisation des variables permet d'effectuer ce test de bon fonctionnement, que les méthodes de diagnostic utilisant des valeurs continues de variables ne peuvent pas, faute de pouvoir vérifier les lois quantitatives du fait de la densité des domaines de valeurs des variables.

3.4 Conclusions

Dans le chapitre 2 nous avons proposé une modélisation comportementale reposant sur la discrétisation des variables du système. A partir de cette discrétisation, trois modèles de comportement ont été proposés pour décrire le comportement de chaque ressource :

- un modèle de comportement physiquement possible incluant :
  - un modèle de bon comportement, associé au mode correct
  - des modèles de mauvais comportements, associés aux modes de défauts des ressources
- un modèle de comportement physiquement impossible

S'appuyant alors sur les contraintes de ces différents modèles de comportement et sur leurs modes associés et en intégrant les résultats de l'analyse AMDEC au titre de connaissance experte du système sous forme d'un Graphe Causal de Dysfonctionnement, des procédures de

diagnostic ont été proposées dans le but d'affiner les procédures actuelles de diagnostic [Désinde et al., 2006*b*] :

- en complétant les diagnostics incomplets par la recherche des modes de défauts conséquents dans le Graphe Causal de Dysfonctionnement
- en éliminant les diagnostics physiquement impossibles grâce au modèle de comportement physiquement possible et aux relations causes/effets de l'AMDEC permettant de vérifier la cohérence des modes de défaillance résultants.
- en localisant le mode de défaut de la ressource en défaut
- en affinant le degré de multiplicité des diagnostics en recherchant les défauts secondaires

Pour approfondir la relation entre AMDEC et analyse diagnostique, nous proposons dans le chapitre suivant une méthodologie pour pronostiquer les modes de défaut et les modes défaillances futures du système à partir de diagnostics minimaux [Reiter, 1987].

Chapitre 4

Pronostic de défaillances et mise en sûreté de l'installation

4.1 Introduction

4.1.1 Problématique

La finalité de ce chapitre est de donner les outils pour permettre de déterminer les risques par pronostic de défaillances et de défauts durant le fonctionnement d'un procédé, d'une installation. Une prédiction en ligne permet de contrôler à tout instant une installation de manière à pouvoir prendre les mesures préventives et/ou correctives qui s'imposent ; ceci dans le but de sauvegarder, dans la mesure du possible, les produits finis, l'installation, le personnel et les bâtiments, en fonction des situations à risque possibles.
Une telle approche est un supplément aux méthodes d'analyse des risques existantes qui sont essentiellement des analyses a priori en vue de la mise en place de systèmes de protection et de prévention.
Ce chapitre propose donc également une approche intégrée analyse des risques / analyse diagnostique. Nous proposons dans cette partie d'injecter les résultats de l'analyse diagnostique dans ceux de l'analyse AMDEC pour développer une méthode de prédiction de défauts et de défaillances et une méthode de mise en sûreté.

4.1.2 Principe

Dans ce chapitre, nous allons nous appuyer sur les résultats de l'analyse AMDEC et sur les résultats d'une analyse diagnostique donnant les diagnostics minimaux [Reiter, 1987] ainsi que les modes des ressources a priori en défaut [Struss & Dressler, 1989].

1. ***De pronostic*** : pour cela, nous allons nous appuyer sur l'analyse fonctionnelle décrite dans le chapitre précédent ainsi que sur l'analyse AMDEC étendue. A partir des résultats de l'AMDEC étendue et du Graphe Causal de Dysfonctionnement, nous proposons de de

probabiliser et de temporiser de manière à rendre le graphe dynamique. Ainsi, à partir des diagnostics précédemment établis, les pronostics pourront alors être déterminés avec comme données leur probabilité d'occurrence et le laps de temps avant occurrence (si occurrence il y a).

2. ***De mise en sûreté*** : à partir des pronostics retenus, nous proposons de construire un nouveau Graphe Causal de Dysfonctionnement pour analyser et déterminer les points à vérifier et à surveiller pour limiter les risques, voire les éliminer.

Ce chapitre propose donc d'une part une méthode de prédiction des modes de défaillances et des modes de défauts et, d'autre part, une méthode de mise en sûreté à partir de résultats de l'analyse AMDEC étendue développée dans le chapitre 2 précédent et s'appuyant sur un algorithme de diagnostic en ligne donnant les diagnostics minimaux [Reiter, 1987] et les modes des ressources en défaut [Struss & Dressler, 1989]. Tout cela en s'appuyant sur l'analyse d'un arbre de défaillances temporisé et probabilisé.

4.2 Positionnement du problème

4.2.1 Approches existantes

Pour prédire les modes de défaillances et les modes de défauts, nous allons nous appuyer sur un Graphe Causal de Dysfonctionnement temporisé prenant en compte l'état du système, les défauts observés et l'évolution des probabilités de chacun des événements du graphe.

L'ajout de la notion temporelle dans le Graphe Causal de Dysfonctionnement peut se faire sous deux formes :

- la première est de considérer que les lois de probabilité sont variantes dans le temps, à l'instar de [Cabarbaye & Ngom, 2001]. Malheureusement, ces lois étant continues, plus l'arbre de défaillance sera complexe, plus le calcul des probabilités des différents éléments de l'arbre le seront, sans oublier les difficultés de calcul en fonction des portes traversées.
- la seconde est d'ajouter des portes de manière à introduire une notion temporelle. On obtient alors des arbres de défaillance dynamiques (DFT : Dynamic Fault Tree) [Dugan et al., 1992] [Meshkat et al., 2002]. L'une des principales limites des arbres de défaillance traditionnels vient de l'impossibilité de modéliser des séquences. Les systèmes pourvus de telles séquences sont habituellement représentés sous la forme de chaînes de Markov [Brémaud, 1999]. Pour parer à cela, les arbres de défaillances dynamiques ont été créés, de manière à ce que les arbres de défaillances puissent modéliser des séquences. Ainsi, différentes portes logiques ont été développées [Boyd, 1991].

Notre objectif est d'évaluer les probabilités de chaque événement au cours du temps en tenant compte des relations de cause à conséquence. Bien que les portes logiques développées pour les arbres de défaillance dynamiques (DFT) apportent une notion temporelle en permettant de prendre en compte les séquences d'événements, ces arbres de défaillances dynamiques ne nous

permettent pas de répondre totalement à notre objectif puisque nous souhaitons introduire des probabilités variant dans le temps, sans pour autant avoir des calculs complexes à résoudre.

Une autre manière de probabiliser les résultats de l'AMDEC pourrait être aussi l'utilisation des chaînes de Markov.
Considérons le résultat d'une analyse AMDEC et représentons sous forme de chaînes de Markov les relations de cause à effet (figure 4.1).

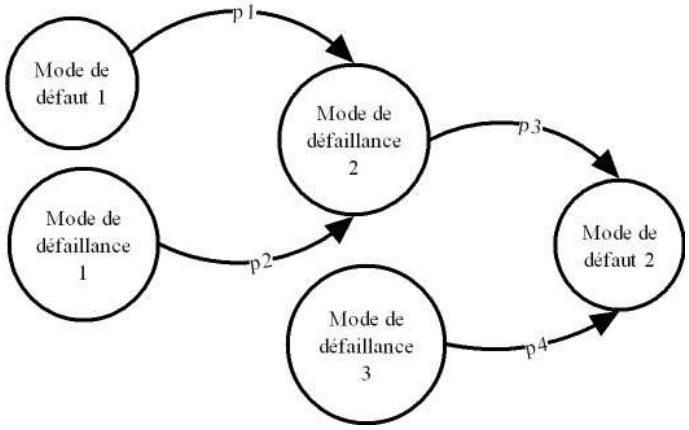


FIG. 4.1 – Résultats d'une analyse AMDEC sous forme de chaîne de Markov

On peut attribuer à chaque transition une probabilité, de manière à pouvoir observer dans le temps l'évolution du système. Cependant :

- on constate que les notions de portes logiques sont perdues par rapport à l'arbre de défaillance : on perd donc de l'information. Cette représentation permet uniquement de représenter les disjonctions d'événements et non les conjonctions, les portes retard, etc.
- il est nécessaire pour cette modélisation de générer des états spécifiques de notre installation alors que nous cherchons simplement à représenter des séquences de cause à effet à partir d'une analyse AMDEC et à calculer des probabilités d'occurrence des modes de défaut et de défaillance.

L'une des finalités du graphe de Markov est de pouvoir établir les probabilités de se retrouver dans un état particulier (ici un mode) à une date précise. En l'occurrence, ici, l'événement indésirable aura une probabilité tendant vers 1 quand t tendra vers l'infini puisque c'est un état absorbant et que la représentation des relations cause/conséquence de l'AMDEC sous forme de graphe de Markov correspond à un système non réparable.
Notre but est d'**évaluer les probabilités de chaque événement au cours du temps** en tenant compte des relations de cause à conséquence et des portes logiques et **non pas de déterminer l'état le plus probable** du système au cours du temps.

4.2.2 Principe

Notre objectif est le suivant : déterminer à chaque instant t les probabilités de chacun des modes du Graphe Causal de Dysfonctionnement à partir :

- des probabilités de départ des *modes de base*, (i.e. situés aux extrémités du graphe) sans utiliser de lois de probabilité continues qui complexifieraient de façon importante le calcul des probabilités des modes conséquents
- des relations de cause à effet entre les différents modes

Ainsi (figure 4.2),

- d'une part, nous avons un Graphe Causal de Dysfonctionnement statique
- d'autre part, nous avons des probabilités évoluant dans le temps, déterminées à partir des relations de cause à effet

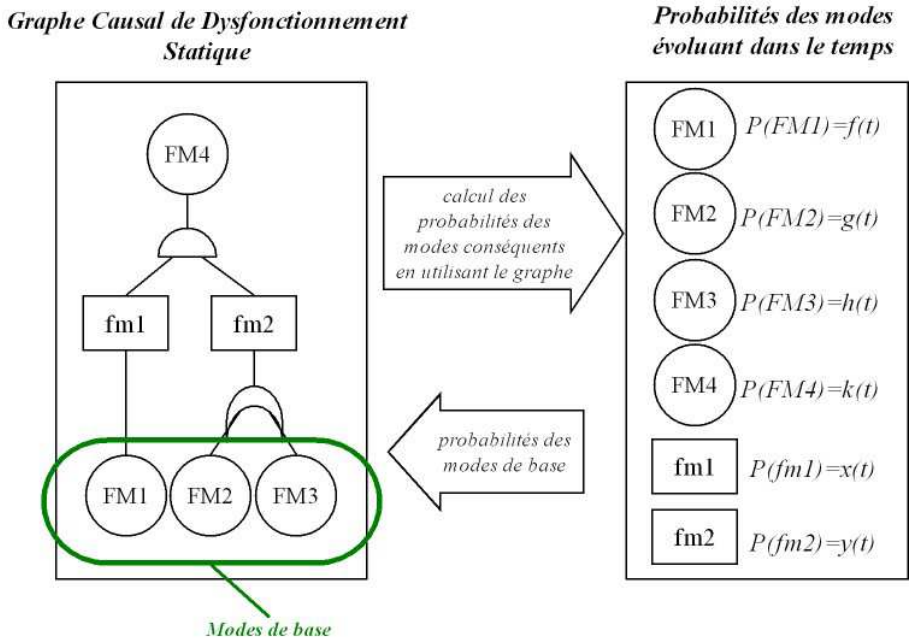


FIG. 4.2 – Interactions probabilités / Graphe Causal de Dysfonctionnement

4.2.3 Hypothèses et méthode de calcul des probabilités

L'hypothèse habituelle prise en compte dans les arbres de défaillance est celle de l'**indépendance des événements** à la base de l'arbre. Nous considérons donc cette même hypothèse pour les modes de base du Graphe Causal de Dysfonctionnement.
Pour rappel, les différentes combinaisons utilisées dans le graphe sont :

- les disjonctions, matérialisées par l'opérateur booléen $+$
- les conjonctions, matérialisées par l'opérateur booléen $\times$
- les négations, matérialisées par l'opérateur booléen $^-$

– les délais, non représentés par des opérateurs booléens, mais n'ayant aucune incidence sur le calcul des probabilités

Coupes minimales

Les éléments du Graphe Causal de Dysfonctionnement sont, pour certains, dépendants d'autres éléments du graphe par le biais de relations de cause à effet via différentes combinaisons possibles. Calculer leurs probabilités dans ce contexte peut donc s'avérer complexe. Pour lever cette complexité, nous allons calculer, pour chacun des modes, leurs *coupes minimales* [Schneeweiss, 1999].

Définition 10 (Coupe d'un mode)**.** Soit m un mode de défaillance ou de défaut du Graphe Causal de Dysfonctionnement.
Une *coupe* du mode m, est un ensemble de modes entraînant ce mode m. On parle aussi parfois de "chemin".

Définition 11 (Coupe minimale d'un mode)**.** Soit m un mode de défaillance ou de défaut du Graphe Causal de Dysfonctionnement.
Une *coupe minimale* du mode m, est la plus petite combinaison de modes entraînant le mode m.

La recherche des coupes minimales d'un mode m se fait à partir d'une transformation du Graphe Causal de Dysfonctionnement en une expression booléenne. Dans la suite, on exprimera toutes les coupes des modes comme par des disjonctions de conjonctions de modes de bases, ce qui est possible car un mode de base n'a pas d'antécédents.

Définition 12 (Ordre d'une coupe minimale d'un mode)**.** Soit m un mode de défaillance ou de défaut du Graphe Causal de Dysfonctionnement.
L'ordre d'une *coupe minimale* du mode m, est le nombre de modes combinés qui figurent dans cette coupe.
Par exemple, si $m = m_1 + m_2.m_3 + m_2.m_4$ est l'expression booléenne du mode m, elle comporte 3 coupes minimales, dont une d'ordre 1 et deux d'ordre 2.

Chaque mode m du Graphe Causal de Dysfonctionnement possède un nombre fini de coupes minimales :

– les coupes minimales d'ordre 1 représentent (si elles existent) les simples modes qui entraînent le mode m
– les coupes minimales d'ordre 2 représentent (si elles existent) les couples de modes qui, se produisant en même temps, entraînent le mode m
– etc.

Probabilités des modes

Grâce au calcul des coupes minimales d'un mode m, nous allons pouvoir déterminer sa probabilité à partir de sa coupe. Soit $m = c_1 + \ldots + c_n$ l'expression booléenne de m, où les c_j sont les coupes minimales.
On a alors :

$$P(m) = P(c_1 + \ldots + c_n)$$

D'après la formule de Poincaré, on a :

$$P(m) = \sum_{k=1}^{n} \left((-1)^{k-1} \sum_{1 \leqslant i_1 < i_2 < \ldots < i_k < n} P(c_{i_1} \times c_{i_2} \times \ldots \times c_{i_k}) \right)$$

Autrement dit,

$$P(m) = \sum_i P(c_i) - \sum_{(i,j)/1 \leqslant i < j \leqslant n} P(c_i \times c_j) + \sum_{(i,j,k)/1 \leqslant i < j < k \leqslant n} P(c_i \times c_j \times c_k) - \ldots + (-1)^n P(c1 \times \ldots \times c_n)$$

Ainsi, la probabilité d'un mode m est donnée par la probabilité de ses coupes minimales.

Exemple : conséquence directe

Considérons deux événements A et B tels que B est la conséquence de A (figure 4.3).
Soient $P(A)$ et $P(B)$ les probabilités respectives de A et B.
Soient a et b les variables booléennes représentant les événements A et B, notations habituellement utilisée dans les arbres de défaillance.

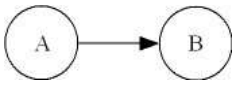


FIG. 4.3 – A implique B

On peut écrire $b = a$ (une seule coupe minimale d'ordre 1).

On peut donc en conclure que si deux événements sont conséquents, on a : $\begin{cases} P(A) = P(B) \\ b = a \end{cases}$

Exemple : conjonction d'événements

Considérons trois événements A, B et C tels que C est la conséquence de A et B (figure 4.4) avec A et B indépendants puisqu'il n'existe aucune relation de cause à effet entre ces deux événements.
Soient $P(A)$, $P(B)$ et $P(C)$ les probabilités respectives de A, B et C.
Soient a, b et c les variables booléennes représentant les événements A, B et C.

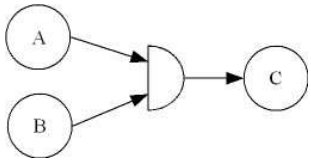


FIG. 4.4 – A et B implique C

On a $c = a.b$ (une seule coupe minimale d'ordre 2).
On peut donc en conclure que si un événement est la conséquence de la conjonction de deux autres, on a : $\begin{cases} P(C) = P(A).P(B) \\ c = a.b \end{cases}$

Exemple : conjonction d'événements avec dépendance

Considérons trois événements A, B et C tels que C est la conséquence de A et B avec B conséquence de A (figure 4.5).
Soient $P(A)$, $P(B)$ et $P(C)$ les probabilités respectives de A, B et C.
Soient a, b et c les variables booléennes représentant les événements A, B et C.

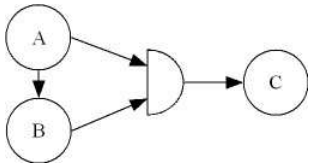


FIG. 4.5 – A implique B avec A et B implique C

On a : $c = a.b = a$ (une seule coupe minimale d'ordre 1)
On peut donc en conclure que : $\begin{cases} P(C) = P(A) \\ c = a \end{cases}$

Conclusions

On voit ici que même si les événements sont dépendants (i.e. il existe une relation de cause à effet entre eux), la probabilité des modes reste calculable en recherchant **les coupes minimales associées** à chacun d'eux.
Ainsi, dans la suite, nous considérerons l'**indépendance de chacun des modes de base** et calculerons les probabilités des événements dépendants en utilisant les méthodes décrites dans la suite.

4.3 Probabilités d'occurrence et temporisation du Graphe Causal de Dysfonctionnement

Dans cette section, nous considérons un Graphe Causal de Dysfonctionnement établi à partir des résultats de l'analyse AMDEC étendue.
Pour parer à la complexité des lois de probabilité continues, nous proposons dans cette partie de définir des lois de probabilités constantes par morceaux, appelées FPE (*Fonction de Probabilité par Episode*). Elles seront associées à chaque mode du Graphe Causal de Dysfonctionnement pour leur définir une probabilité en fonction du temps. Ensuite, nous développerons les différentes portes logiques qui seront utilisées.

4.3.1 Notion de FPE : Fonction de Probabilité Par Episode

Nous proposons ici de définir la *probabilité* d'occurrence des modes par *périodes (intervalles de temps)*. Cette information a pour but de déterminer, à chaque date t de vie du système, la probabilité de chaque événement.

Définition 13 (FPE)**.** Soit m un mode (de défaillance ou de défaut).
On appelle **Fonction de Probabilité par Episode** du mode m, et on note $FPE(m)$, la fonction constante par morceaux qui définit la probabilité d'occurrence du mode m en fonction d'intervalles de temps :

$$FPE(m) = ((p_1, \Delta t_1), \ldots, (p_n, \Delta t_n))$$

tels que $\forall t \in \Delta t_i, P(t) = p_i$ avec $\Delta t_i = [t_i^-, t_i^+[$

Graphiquement, une FPE correspond à la figure 4.6.

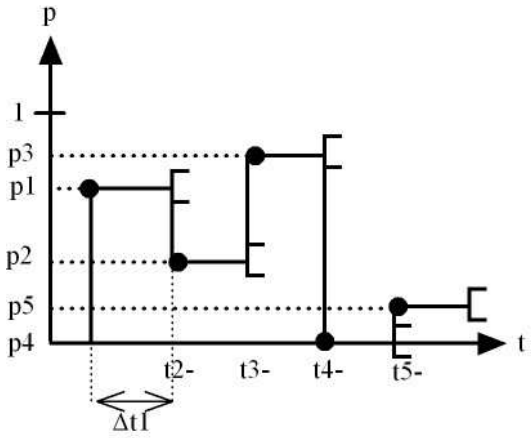


FIG. 4.6 – Représentation graphique d'une FPE

Notation 7. On notera par ailleurs $T(m) = \{t_1, t_2, \ldots, t_i, t_j, \ldots\}$ l'ensemble ordonné des dates des périodes de $FPE(m)$, tel que $\forall i < j, t_i < t_j$.

Remarques :

- $\forall i, t_i^+ = t_{i+1}^-$
- $t_{app} = \min_{\Delta t_i} t_i^-$ tel que $p_i \neq 0$ est appelé "date d'apparition".

Ainsi le Graphe Causal de Dysfonctionnement devient pourvu de FPE comme représenté en figure 4.7.

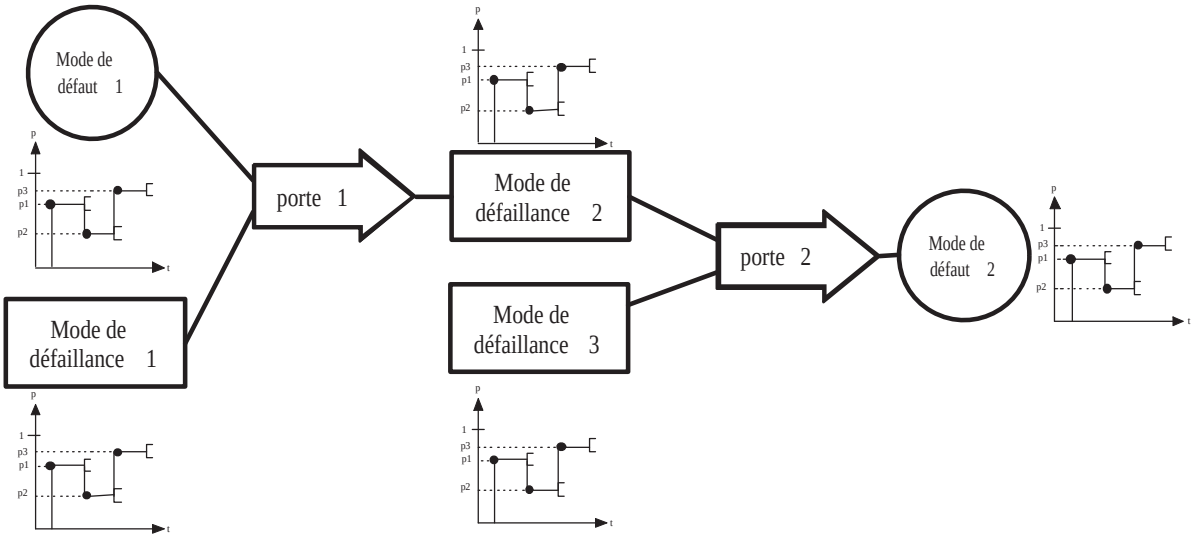


FIG. 4.7 – Arbre de défaillance pourvu de FPE

Les données de départs sont les FPE des *modes de base*, les autres FPE étant déduites en traversant les portes logiques.

Remarque :

On distingue deux types de FPE :

- Les FPE dont la probabilité du mode m est non nulle ($p \neq 0$) après écoulement de la dernière période (figure 4.8)

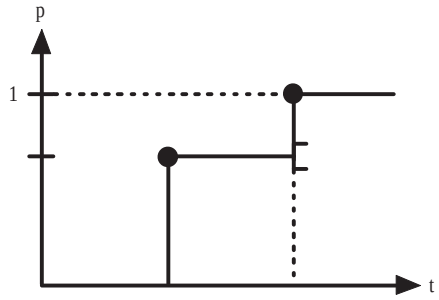


FIG. 4.8 – Date de danger

$t_{danger} = \max_{\Delta t_i} t_i^-$ est alors appelé "date de danger"

- Les FPE dont la probabilité du mode m est à 0 après écoulement de la dernière période (figure 4.9)

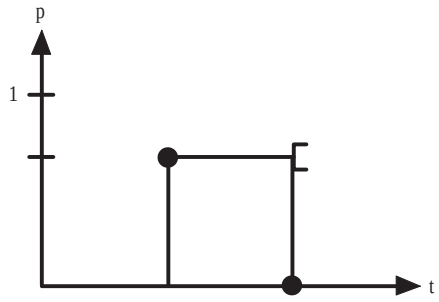


FIG. 4.9 – Date de sécurité

$t_{securite} = \max_{\Delta t_i} t_i^-$ est alors appelé "date de sécurité"

4.3.2 Combinaisons logiques utilisées

Seuls les modes de base du graphe sont pourvus de FPE ; les FPE des événements intermédiaires sont déduites calculant les probabilités des coupes minimales. Ainsi, dans cette section, nous allons détailler les différentes combinaisons qui vont être utilisées dans le graphe et les

modes de calcul qui en découlent sur les FPE.
Dans le Graphe Causal Dysfonctionnement, nous allons utiliser les combinaisons suivantes :

- les conjonctions
- les disjonctions
- les négations
- les délais, qui ne seront pas prise en compte lors des calculs de probabilité, mais qui seront utilisés lors de la phase de pronostic et de mise en sûreté

Les disjonctions

Les disjonctions d'événements sont habituellement représentés par la porte logique OU. On considère n modes indépendants (de défaut ou de défaillance) en amont d'une porte OU et leur FPE associées (figure 4.10).

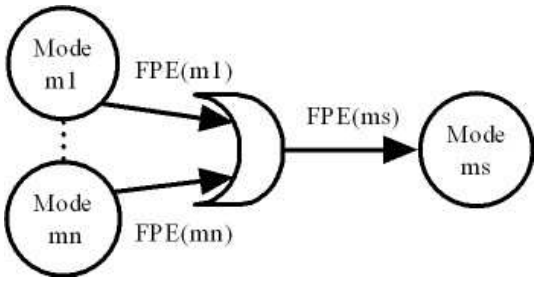


FIG. 4.10 – Porte OU

On a : $m_s = m_1 + \ldots + m_n$ au sens booléen. Il n'y a donc que des coupes minimales d'ordre 1. La FPE du mode en aval (m_s) de la porte est donc définie par :

- Son ensemble ordonné des dates donné par : $T(m_s) = \bigcup_{m_k \in \{m_1,\ldots,m_n\}} T(m_k)$

  Les périodes sont donc définies par : $\begin{cases} t_1^- = \min T(m_s) \\ t_2^- = \min(T(m_s) - \{t_1^-\}) \\ \vdots \end{cases}$

- Ses probabilités d'occurrences sont définies par : $\forall t \in \Delta t_i, P(m_s) = P\left(\sum_{k=1}^{n} m_k\right)$ que l'on pourra calculer en utilisant la formule de Poincaré par exemple.

Graphiquement, ces résultats sont résumés par la figure 4.12 (page 125) où deux modes seulement sont considérés en amont de la porte.

Les conjonctions

Les conjonctions d'événements sont habituellement représentées par la porte logique ET. On considère n modes indépendants (de défaut ou de défaillance) en amont d'une porte ET et leur FPE associées (figure 4.11).

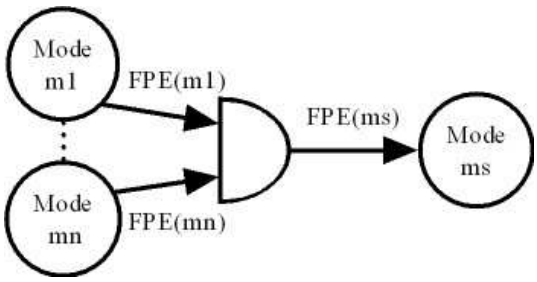


FIG. 4.11 – Porte ET

On a : $m_s = m_1 \ldots . m_n$ au sens booléen. Il n'y a que des coupes minimales d'ordre 1.
La FPE du mode en aval (m_s) de la porte est donc définie par :

- Son ensemble ordonné des dates donné par : $T(m_s) = \bigcup_{m_k \in \{m_1,\ldots,m_n\}} T(m_k)$

  Les périodes sont donc définies par : $\begin{cases} t_1^- = \min T(m_s) \\ t_2^- = \min(T(m_s) - \{t_1^-\}) \\ \vdots \end{cases}$
- Ses probabilités d'occurrences sont définies par : $\forall t \in \Delta t_i, P(m_s) = \prod_i P(m_i)$

Graphiquement, ces résultats sont résumés par la figure 4.13 (page 125) où deux modes seulement sont considérés en amont de la porte.

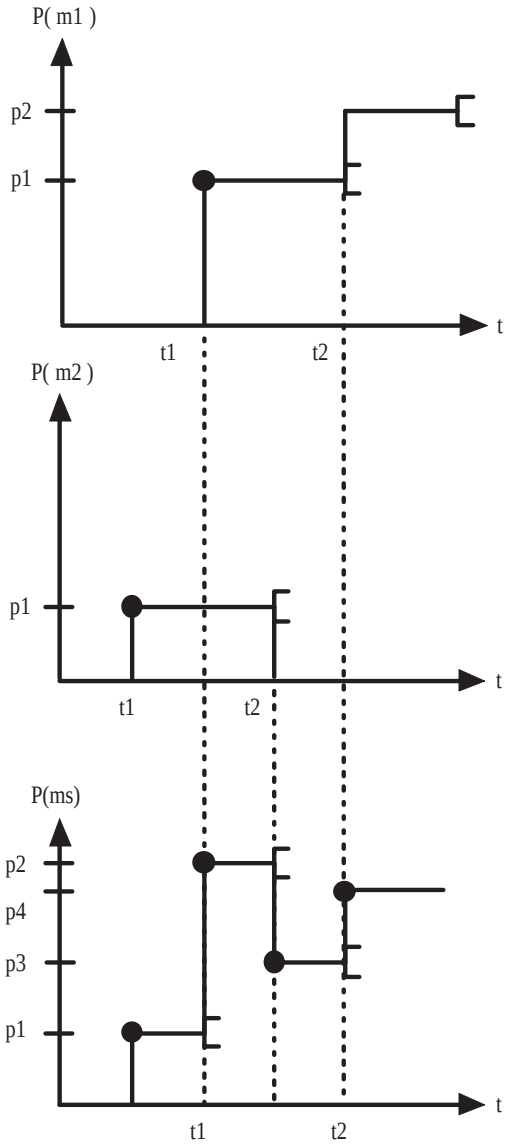


FIG. 4.12 – Calcul d'une FPE en sortie d'une porte OU

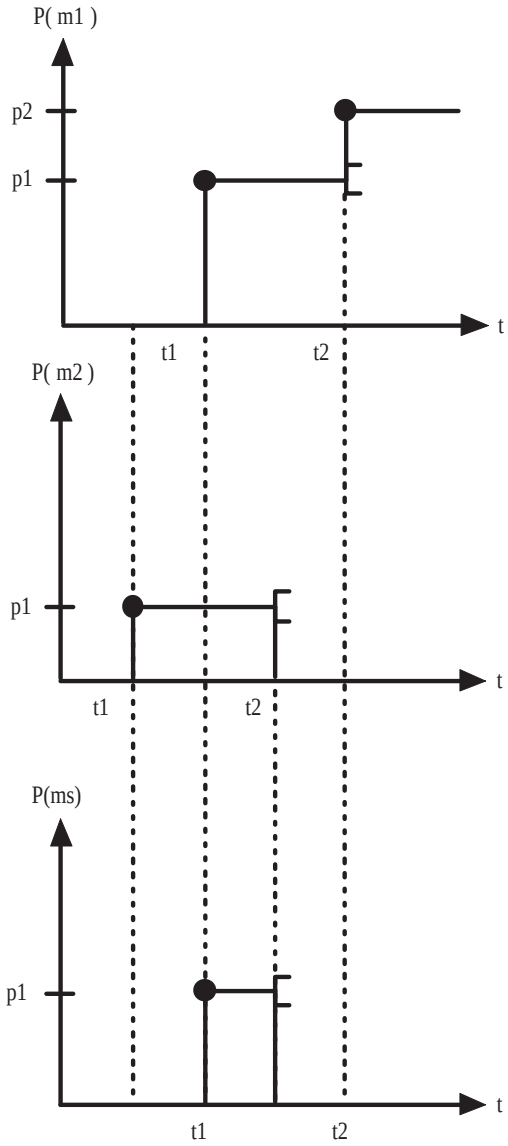


FIG. 4.13 – Calcul d'une FPE en sortie d'une porte ET

La négation

La négation est habituellement représentée par la porte logique NON (figure 4.14).

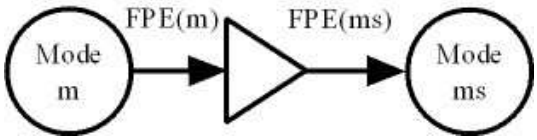


FIG. 4.14 – Porte NON

On a : $m_s = \overline{m}$ au sens booléen.
En sortie d'une porte NON, la FPE du mode m_s est définie par :

- $T(m_s) = T(m)$: les périodes restent donc inchangées
- $\forall t \in \Delta t_i, P(m_s) = 1 - P(m)$

4.3.3 Exemple

Considérons à nouveau le rétroprojecteur (figure 4.15).

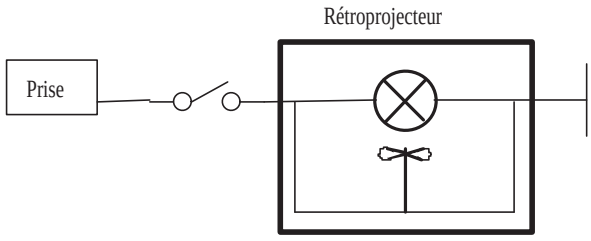


FIG. 4.15 – Schéma simplifié d'un rétroprojecteur

L'analyse AMDEC effectuée sur ce système a permis de mettre au point le Graphe Causal de Dysfonctionnement du système représenté en figure 4.16.

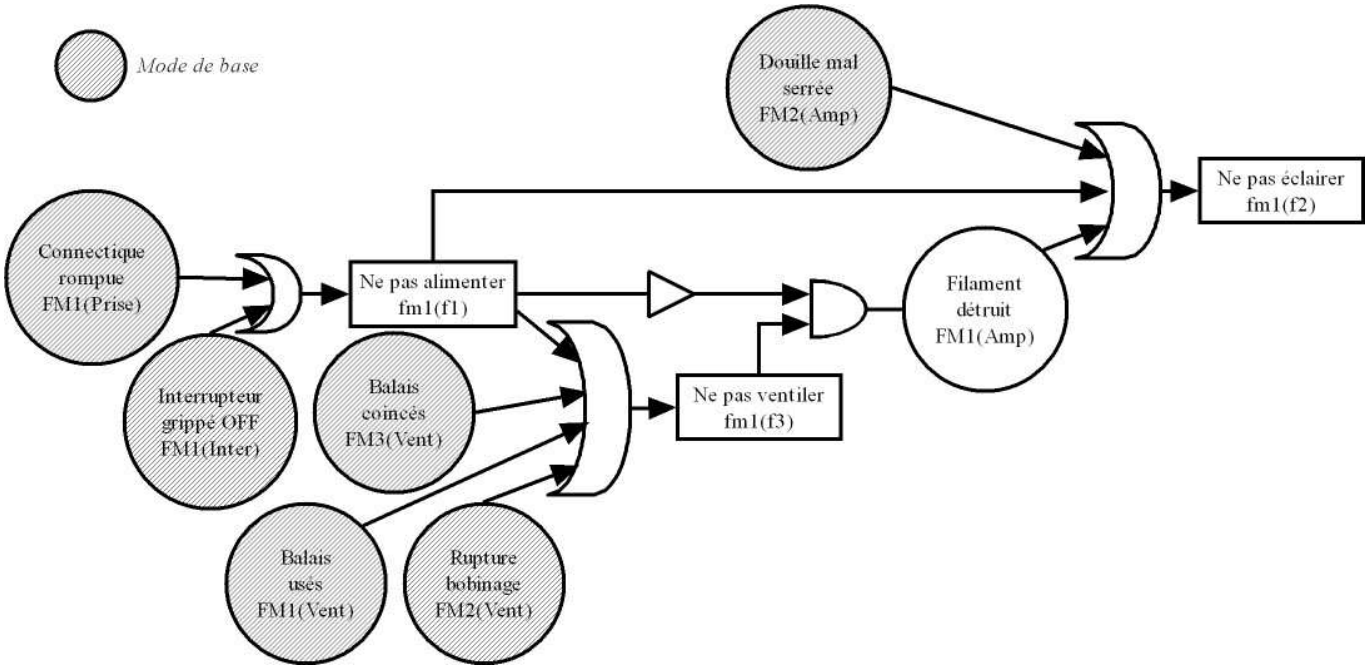


FIG. 4.16 – Extrait du Graphe Causal de Dysfonctionnement du rétroprojecteur

On considère par ailleurs les FPE des modes de base du graphe, représentées respectivement par les figures 4.17, 4.18, 4.19 et 4.20 :

– La FPE du mode de défaut $FM_3(Vent)$ "Balais coincés" est donnée par :

$$FPE(balais - coinces) = \begin{pmatrix} (0, [0, 1000h[) \\ (0.3, [1000h, 3000h[) \\ (0.7, [3000h, 5000h[) \\ (0.8, [5000h, +\infty[) \end{pmatrix}$$

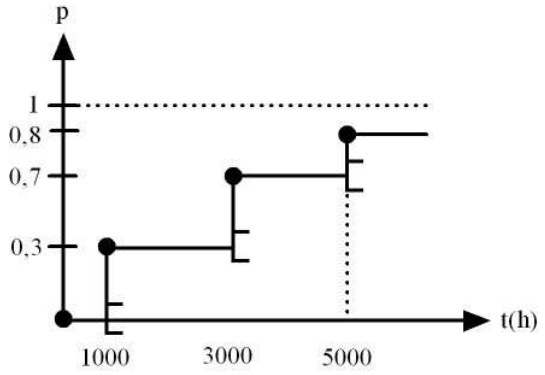


FIG. 4.17 – FPE du mode de défaut $FM_3(Vent)$ "Balais coincés"

– La FPE du mode de défaut $FM_1(Vent)$ "Balais usés" est donnée par :

$$FPE(balais - uses) = \begin{pmatrix} (0, [0, 5000h[) \\ (0.1, [5000h, +\infty[) \end{pmatrix}$$

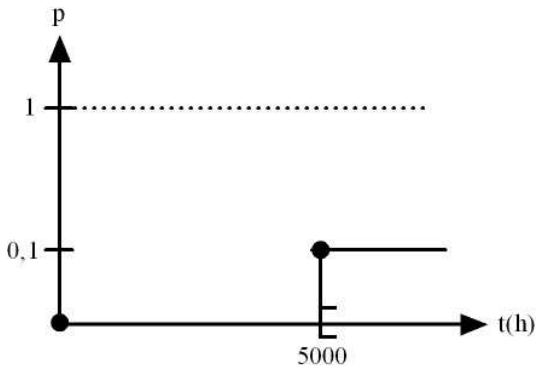


FIG. 4.18 – FPE du mode de défaut $FM_1(Vent)$ "Balais usés"

– La FPE du mode de défaut $FM_2(Vent)$ "Rupture Bobinage" est donnée par :

$$FPE(rupture - bobinage) = \begin{pmatrix} (0, [0, 1000h[) \\ (0.05, [1000h, 2000h[) \\ (0.1, [2000h, +\infty[) \end{pmatrix}$$

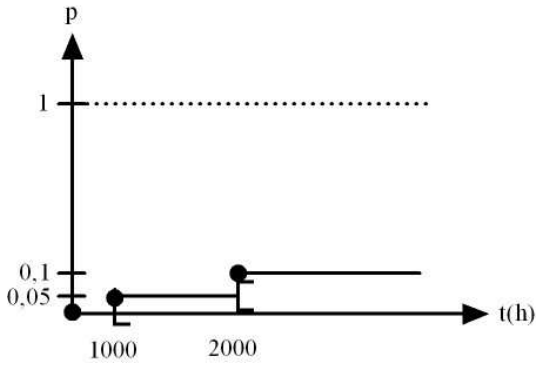


FIG. 4.19 – FPE du mode de défaut $FM_2(Vent)$ "Rupture Bobinage"

– La FPE du mode de défaut $FM1(Prise)$ "Connectique rompue" est donnée par :

$$FPE(connectique - rompue) = \begin{pmatrix} (0, [0, 10000h[) \\ (0.03, [10000h, 50000h[) \\ (0.04, [50000h, +\infty[) \end{pmatrix}$$

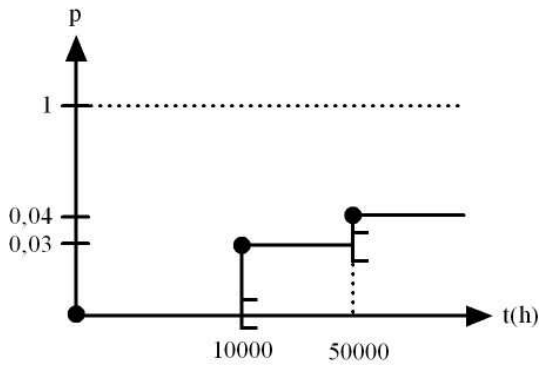


FIG. 4.20 – FPE du mode de défaut $FM1(Prise)$ "Connectique rompue"

La coupe minimale du mode de défaillance $fm_1(f_1)$ "Ne pas alimenter" est définie par $fm_1(f_1) = FM_1(Prise)$ au sens booléen.
Ensuite, la coupe minimale en sortie de la porte NON est définie par $\overline{fm_1(f_1)}$.
En conséquence, la FPE en sortie de la porte NON est définie par (figure 4.21) :

$$FPE = \begin{pmatrix} (1, [0, 10000h[) \\ (0.97, [10000h, 50000h[) \\ (0.96, [50000h, +\infty[) \end{pmatrix}$$

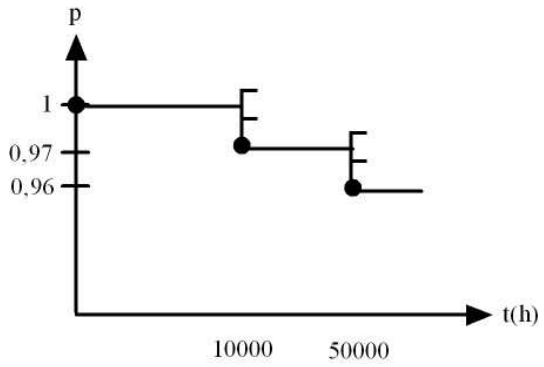


FIG. 4.21 – FPE en sortie de la porte NON

Détermination de la FPE du mode de défaillance $fm_1(f_3)$ "Ne pas ventiler"

La coupe minimale du mode de défaillance $fm_1(f_3)$ "Ne pas ventiler' est définie par $fm_1(f_3) = FM_1(Vent) + FM_2(Vent) + FM_3(Vent) + FM_1(Prise)$.

Le calcul de cette FPE est donné en figure en utilisant la formule de Poincaré 4.22 : $P(fm_1(f_3)) = P(FM_1(Vent)) + P(FM_2(Vent)) + P(FM_3(Vent)) +$

$P(FM_1(Prise)) - P(FM_1(Vent).FM_2(Vent)) - P(FM_1(Vent).FM_3(Vent)) - P(FM_1(Vent).FM_1(Prise)) - P(FM_2(Vent).FM_3(Vent)) - P(FM_2(Vent).FM_1(Prise)) - P(FM_3(Vent).FM_1(Prise)) + P(FM_1(Vent).FM_2(Vent).FM_3(Vent)) + P(FM_1(Vent).FM_2(Vent).FM_1(Prise)) + P(FM_2(Vent).FM_3(Vent).FM_1(Prise)) - P(FM_1(Vent).FM_2(Vent).FM_3(Vent).FM_1(Prise))$

Ainsi, par exemple, à la date $t = 1000$, $P(fm_1(f_3)) = 0 + 0.05 + 0 + 0.3 - 0.05 \times 0.3 = 0.335$

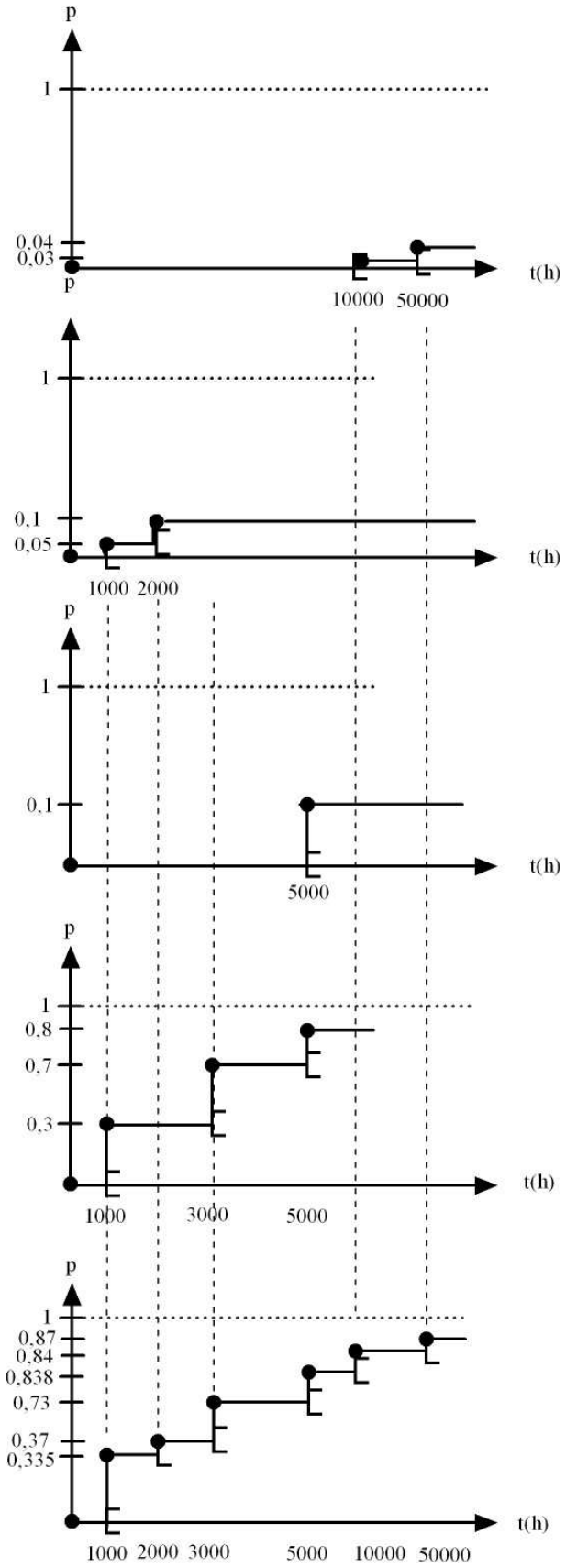


FIG. 4.22 – Calcul de la FPE du mode de défaillance $fm_1(f_3)$ "Ne pas Ventiler"

On obtient ainsi :

$$FPE(\text{“}NePasVentiler''\text{)} = \begin{pmatrix} (0, [0, 1000h[) \\ (0.335, [1000h, 2000h[) \\ (0.37, [2000h, 3000h[) \\ (0.73, [3000h, 5000h[) \\ (0.838, [5000h, +10000h[) \\ (0.84, [10000h, +50000h[) \\ (0.87, [50000h, +\infty[) \end{pmatrix}$$

Détermination de la FPE du mode de défaut $FM_1(Amp)$ “Filament Détruit”

La coupe minimale associée au mode de défaut $FM_1(Amp)$ “Filament Détruit”, après réduction de la coupe minimale, est définie par $FM_1(Amp) = FM_1(Prise) + FM_1(Vent) + FM_2(Vent) + FM_3(Vent)$ (qui correspond à la même coupe minimale que l'événement $fm_1(f_3)$: “Ne pas Ventiler”)

Ainsi, la FPE du mode de défaut $FM_1(Amp)$ “Filament Détruit” est définie par :

$$FPE(FM_1(Amp)) = \begin{pmatrix} (0, [0, 1000h[) \\ (0.335, [1000h, 2000h[) \\ (0.37, [2000h, 3000h[) \\ (0.73, [3000h, 5000h[) \\ (0.838, [5000h, +10000h[) \\ (0.84, [10000h, +50000h[) \\ (0.87, [50000h, +\infty[) \end{pmatrix}$$

4.4 Pronostic et recherche des risques

A partir du Graphe Causal de Dysfonctionnement formalisé d'après les éléments précédemment détaillés et à partir des données de l'état du système, nous proposons dans cette partie une méthode pour prédire les défaillances et les défauts éventuels en temps réel, ainsi que les mesures à prendre pour limiter ces risques, voire les éliminer.

La figure 4.23 représente les données disponibles du système. A partir de ces données, on est en mesure de diagnostiquer les éventuels défauts du système en suivant les méthodes détaillées dans le chapitre 3. A partir des résultats de cette analyse diagnostique nous proposons une méthode permettant de déterminer les défauts futurs et les défaillances futures en partant de chaque diagnostic et en précisant pour chacun des pronostics leur probabilité et leurs délais d'apparition en utilisant le Graphe Causal de Dysfonctionnement.

Suite à ce pronostic, nous proposons ensuite de construire un nouveau graphe (privé des défauts et des défauts et défaillances détectés) dont l'analyse va nous permettre de mettre en évidence les points du système à surveiller pour limiter les dangers, voire les éliminer, bien sûr quand cela est possible. Par ailleurs, si la gravité de chacun des modes de défaut et de défaillance des ressources du système est disponible, il sera possible d'évaluer ces risques en calculant leur *criticité* car leur probabilité est calculable via leur FPE.

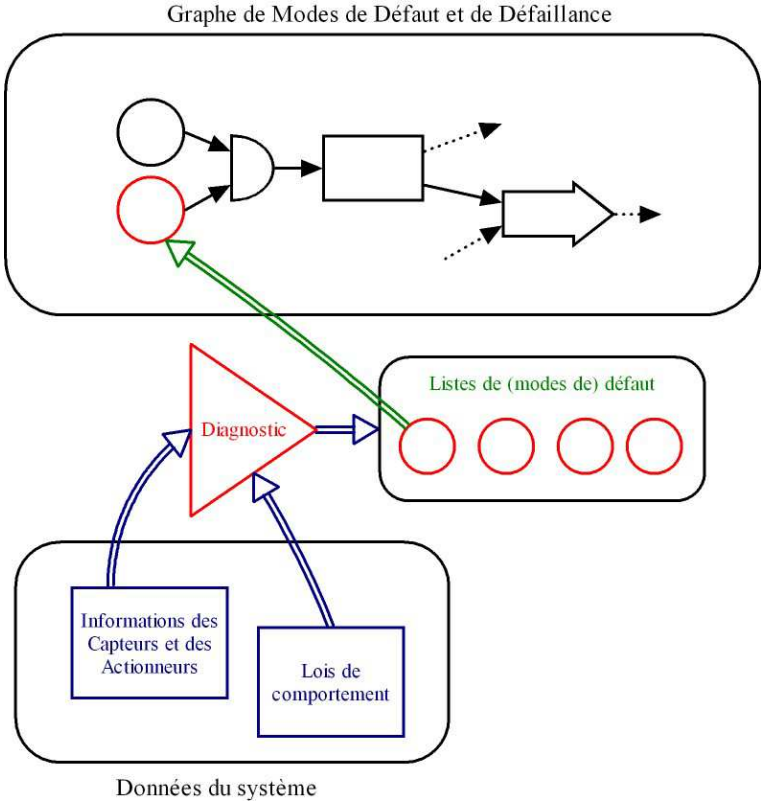


FIG. 4.23 – Données disponibles du système

4.4.1 Méthodologie de pronostic

La méthodologie de pronostic que nous proposons dans cette section repose sur deux étapes successives et exploite les résultats de l'analyse diagnostique effectuée selon les procédures détaillées dans le chapitre 3. Ces étapes successives sont les suivantes :

1. dans un premier temps, les FPE des modes de défauts diagnostiqués sont modifiés. En effet, étant détectés et localisés, leurs probabilités d'occurrence passent à 1 à partir de la date à laquelle le diagnostic a été effectué. Les FPE des modes conséquents risquent d'être donc à leur tour modifiées également
2. dans un deuxième temps, les ressources apparaissant comme étant en défaut sont repérées dans le Graphe Causal de Dysfonctionnement et toutes leurs conséquences valides (au sens booléen) sont déterminées. Cette recherche va nous permettre de présager des

défauts et défaillances futurs du système.

Soit $DiagSet$ l'ensemble des diagnostics $DiagSet = \{d_1, \ldots, d_n\}$ avec $d_i = \{m_1, \ldots, m_p\}$ où m est un mode de défaut.

Étape 1 : Mise à jour du Graphe Causal de Dysfonctionnement

La première étape avant le pronostic proprement dit est la mise à jour des FPE des éléments du Graphe Causal de Dysfonctionnement en fonction des diagnostics. Pour chaque diagnostic, un certain nombre de ressources sont soupçonnées d'être dans un certain mode de défaut donné par le diagnostic.
Ces modes de défaut présents dans le graphe sont pourvus d'une FPE comme chacun des modes. Or, comme ces modes de défaut ont été détectés, ils voient leur probabilité d'occurrence passer à 1 à la date t_{diag} où le diagnostic a été effectué (Figure 4.24).

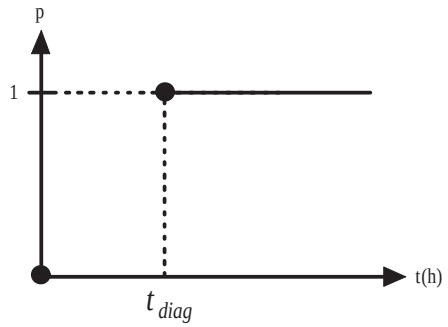


Fig. 4.24 – FPE d'un mode détecté par l'analyse diagnostique

Les FPE de chacun des modes détectés doivent alors être mises à jour et cela pour chacun des diagnostics, les FPE des autres événements changeant alors suivant les lois des différentes portes du graphe.

Étape 2 : Prédiction des modes de défaut conséquents et des risques

Dans cette étape, nous proposons une méthode permettant, en fonction des modes de défaut diagnostiqués, de déterminer à partir du Graphe Causal de Dysfonctionnement les différents modes susceptibles d'être les causes des diagnostics. Pour cela, nous proposons d'analyser le graphe pour en rechercher les modes valides lors de la propagation des modes de défaut diagnostiqués.

Cas particuliers :

- si le mode à propager est un mode de base, aucune précaution à prendre
- si le mode à propager est en aval d'une porte ET. On sait par déduction que tous les événements en amont de cette porte ET sont valides : ce cas de figure ne pose pas de problèmes.
- par contre, si le mode à propager est à l'aval d'une porte OU, il est impossible de déterminer lequel des modes amonts est valide. Pour savoir s'il est possible de propager ce mode, il faut calculer les expressions booléennes de chacun des modes avals. Si ces expressions contiennent un des modes en amont de la porte OU, alors il est impossible de conclure et de propager.

Pour pronostiquer les défauts et les défaillances futures, nous proposons de repérer dans le Graphe Causal de Dysfonctionnement les modes de défaut diagnostiqués et de propager ces modes de défauts dans le graphe :

- à chaque porte OU rencontrée, on relève le ou les modes conséquents, qui sont des **prédictions** (figure 4.25) et on continue de propager

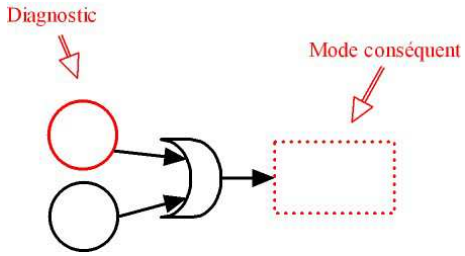


FIG. 4.25 – Mode prédit

- à chaque porte ET rencontrée, on relève le ou les modes en aval qui sont des **risques** (figure 4.26) et on arrête la propagation

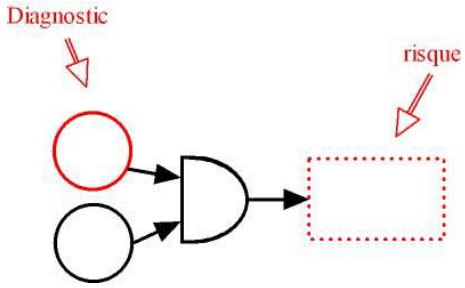


FIG. 4.26 – Risque

- à chaque porte DELAI rencontrée, on poursuit la propagation
- à chaque porte NON rencontrée : si la coupe minimale de l'élément en amont a pour valeur "faux", alors on poursuit la propagation, sinon on arrête la propagation.

La recherche de pronostics est résumée dans l'algorithme ci-après.

Algorithme de prédiction des modes de défaut/défaillance et des risques

Procédure Pronostic(Mode m, Graphe GCD)
PourChaque diagnostic d de $DiagSet$ **Faire**
Repérer les positions des modes appartenant à d dans $GMDD$
PourChaque mode m appartenant à d **Faire**
Boucler
PourChaque porte logique rencontrée **Faire**
Si la porte est une porte OU **Alors**
Ajouter les modes à la liste des modes prédits et continuer la propagation
Sinon Si la porte est une porte ET **Alors**
Ajouter les modes à la liste des risques et arrêter la propagation
Sinon Si la porte est une porte DELAI **Alors**
Poursuivre la propagation
Sinon Si la porte est une porte NON **Alors**
Si la coupe minimale de l'élément en amont a pour valeur "faux", alors on poursuit la propagation, sinon on l'arrête
FinSi
FinChaque
BouclerTantQue on peut encore propager
FinChaque
FinChaque
FinProcédure

4.4.2 Evaluation des risques

En analysant chaque nouvel arbre de défaillance correspondant à chaque diagnostic, les FPE des différents risques établis précédemment permettent d'estimer :

- la probabilité d'occurrence des risques en temps réel
- le délai avant occurrence des risques en temps réel

On peut alors évaluer les risques calculant leur criticité par la formule :

$$criticite = gravite \times probabilite$$

La gravité est une donnée du système et la probabilité est fournie en temps réel par la FPE. Ainsi, il est possible d'évaluer les risques en temps réel à partir d'un défaut localisé.

4.5 Mise en sûreté de l'installation

Dans cette partie, à partir des risques établis précédemment, nous proposons une méthode permettant de rechercher les points à surveiller pour limiter, voire éliminer ces défauts et ces

risques.
Dans un premier temps, une fois les risques déterminés, nous proposons de les classer par ordre de criticité ou de probabilité d'occurrence.

Pour déterminer les points du système à surveiller pour limiter les risques, voire les éliminer, nous proposons de créer un nouveau Graphe Causal de Dysfonctionnement, correspondant au Graphe Causal de Dysfonctionnement de départ auquel ont été ôtés les modes diagnostiqués. Pour cela, on met à jour les formules logiques de chacun des modes en considérant chaque mode diagnostiqué avec la valeur booléenne "vraie".

Pour déterminer les points à surveiller pour limiter les risques, nous proposons d'analyser chacune des formules logiques associées à ces risques. Nous rappelons qu'une telle formule est une disjonction de conjonction. Ainsi, les éléments à surveiller sont ceux parmi la coupe minimale du risque qui a la probabilité la plus élevée.

4.6 Exemple

Dans cet exemple, nous reprenons l'étude du rétroprojecteur dont les FPE ont été définies à la section 4.3.3 en page 126. La figure 4.27 donne le Graphe Causal de Dysfonctionnement du système.

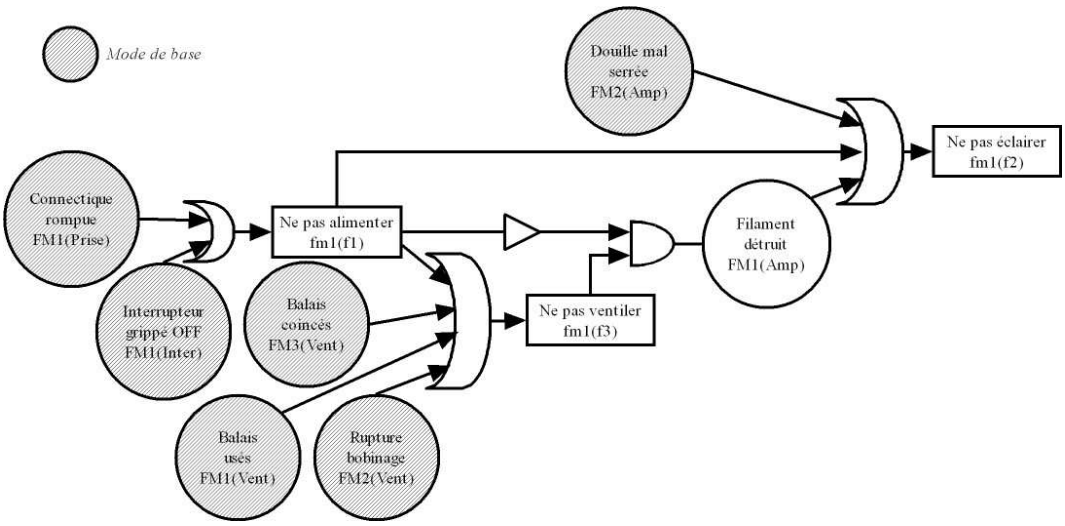


FIG. 4.27 – Extrait du Graphe Causal de Dysfonctionnement du rétroprojecteur

4.6.1 Diagnostic et mise à jour de l'arbre des défaillances

Supposons que l'analyse diagnostique réalisée sur ce système à la date $t_{diag} = 2000h$ conduise au résultat suivant : le *ventilateur* est a priori défaillant, avec comme mode de défaut détecté :

$FM_3(Vent)$ "balais coincés".
Étant donné ce diagnostic, la FPE du mode "balais coincés" se voit donc modifiée par :

$$FPE(BalaisCoinces) = (1, [t_{diag} = 2000h, +\infty[)$$

Ainsi,

$$FPE(NePasVentiler) = (1, [t_{diag} = 2000h, +\infty[)$$

(conséquence directe du mode de défaut dans le Graphe Causal de Dysfonctionnement).
On a par ailleurs l'événement $FPE(BalaisCoinces)$ qui qui devient "vrai" au sens booléen.

Ainsi, $FM_1(Amp) = \overline{FM_1(Prise)}$
En conséquence, la FPE associée du mode $FM_1(Amp)$ "Filament Détruit" est définie par celle de $\overline{FM_1(Prise)}$ autrement dit par :

$$\begin{pmatrix} (1, [0, 10000h[) \\ (0.7, [10000h, 50000h[) \\ (0.6, [50000h, +\infty[) \end{pmatrix}$$

4.6.2 Pronostic et mise en sûreté

Le nouveau graphe, privé des modes détectés est donné en figure 4.28.

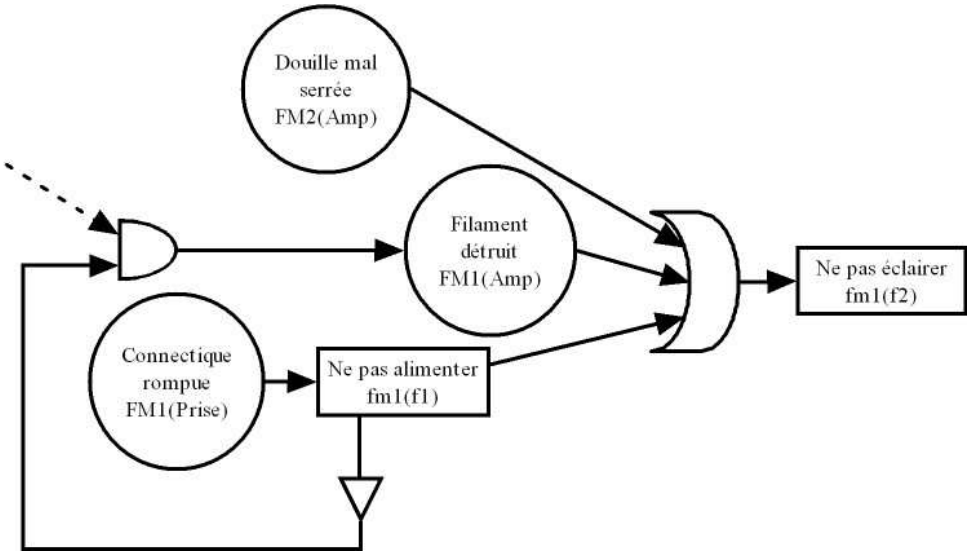


FIG. 4.28 – Nouveau Graphe Causal de Dysfonctionnement du rétroprojecteur

L'analyse du graphe à partir de l'algorithme de pronostic détaillé dans la section 4.4.1 en page 133 nous conduit à :

– l'ensemble des modes prédits : $\{fm_1(f_3)$: "Ne pas Ventiler"$\}$

– l'ensemble des risques : $\{FM_1(Amp)$: "Filament détruit"$\}$

L'analyse de la FPE du mode $FM_1(Amp)$: "Filament Détruit" nous annonce que cet événement ne se réalisera pas avant $20min$, mais qu'après ce délai, le danger est réellement imminent.

Pour éviter que le filament ne soit détruit, et donc au final que le rétroprojecteur n'éclaire plus, l'algorithme de mise en sûreté décrit en section 4.5 (page 136) nous indique qu'il faut donc faire en sorte que la connectique "se rompe", autrement dit, il suffit de débrancher le rétroprojecteur avant $20min$. Dans ce cas, le filament ne grillera pas (figure 4.29).

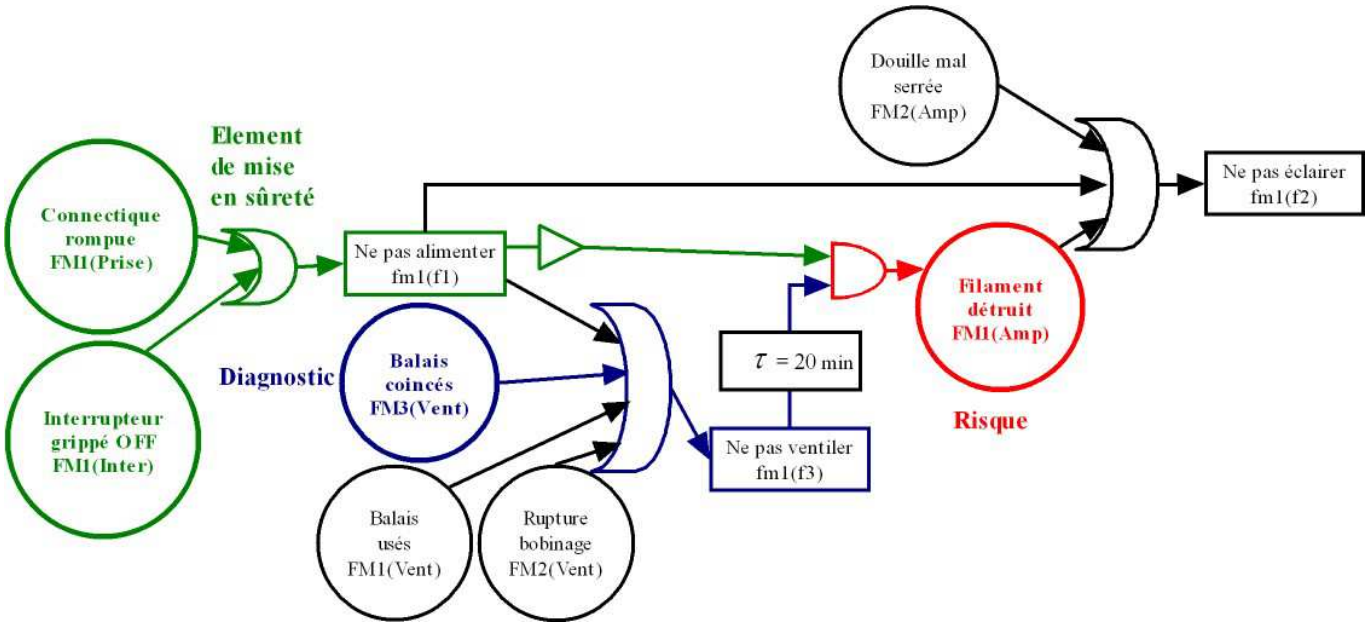


FIG. 4.29 – Mise en sûreté du rétroprojecteur

4.7 Conclusion

Dans ce chapitre nous avons proposé un formalisme pour probabiliser les Graphes de Causaux de Dysfonctionnement [Désinde et al., 2006*a*] en introduisant la notion de FPE (Fonction de Probabilité par Episode). Associées à chaque événement, ces FPE nous permettent de déterminer à chaque instant de vie du système, les probabilités des événements de l'arbre ainsi que leurs délais d'occurrence. Ces FPE ont été proposées car la prise en compte de lois de probabilité continues pour représenter la probabilités des événements résultant de conjonctions ou disjonctions est très complexe.
Dans la suite, grâce à cette probabilisation du Graphe Causal de Dysfonctionnement, en partant du résultat d'une analyse diagnostique donnant les diagnostics minimaux [Reiter, 1987] et les modes des ressources en défaut [Struss & Dressler, 1989], une méthodologie de pronostic de

défaillances et de défauts futurs en temps réel a été proposée. Cette méthodologie repose sur l'analyse du Graphe Causal de Dysfonctionnement, dans lequel sont "propagés" les modes de défaut détectés. Ainsi, il est possible d'établir la liste des risques potentiels du système, leur probabilité et leur délai d'occurrence.
Enfin, pour faire face à ces risques, un algorithme de mise en sûreté a été proposé, analysant le Graphe Causal de Dysfonctionnement pour en tirer les points à surveiller ou les actions à entreprendre pour limiter ces risques, voire les éliminer.

Chapitre 5

Application à un procédé exothermique industriel

5.1 Introduction

Dans ce chapitre nous allons considérer un procédé de fabrication industrielle d'un produit chimique, dont l'installation est représentée en figure 5.5 en page 143. Cette application est tirée d'une installation de la société *Rhône-Poulenc*.

La nomenclature de représentation de l'installation est la suivante :
– la figure 5.1 représente les différents types de vannes utilisées

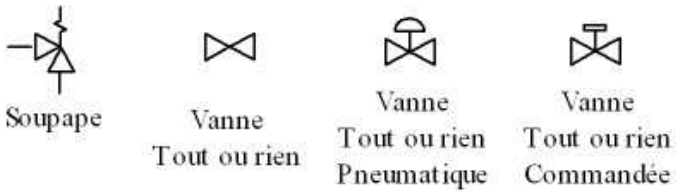


FIG. 5.1 – Liste des vannes utilisées dans l'application

– la figure 5.2 représente les différents types de capteurs/actionneurs utilisés

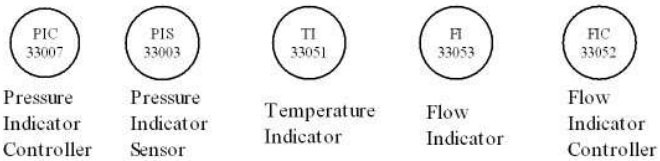


FIG. 5.2 – Liste des capteurs utilisés dans l'application

– la figure 5.3 représente les différents types de connexions utilisées

FIG. 5.3 – Liste des connexions utilisées dans l'application

– la figure 5.4 représente les composants restants

FIG. 5.4 – Liste des autres composants utilisés dans l'application

– la numérotation des différents composants de l'installation a été entièrement reprise par rapport à l'installation originale. La nomenclature utilisée est celle d'un découpage en différentes sections : par exemple, tous les composants se rattachant à la section 33 seront étiquetés 33???.

Le principe de fonctionnement de l'installation est le suivant :

→ les vannes $XV26036$, $XV22307$ et $XV24308$ servent à introduire dans le réacteur $R33030$ respectivement les additifs 1, 2 et 3.

→ la fabrication du produit fini est le lieu d'une réaction chimique exothermique à fort pouvoir calorifique.
Pour fournir un produit fini conforme aux normes de qualités ainsi que pour éviter l'emballement de la réaction, le réactif est propulsé par la pompe $P33040$ vers l'échangeur $E33040$, où il sera en contact indirect avec de l'eau froide de manière à abaisser sa température.
L'arrivée d'eau froide dans le système est assurée par la vanne $XYSV33027$.
Un système d'asservissement permet de réguler la quantité d'eau froide arrivant dans l'échangeur pour ajuster la température du milieu réactionnel, via le capteur de température $TI33051$ et l'actionneur $FIC33052$ de commande de la vanne $CV33053$:
  – Si la quantité d'eau froide est trop faible, la réaction risque de s'emballer et d'augmenter la pression dans le réacteur
  – Si la quantité d'eau froide est trop forte, le produit fini ne répondra pas aux normes de qualité et sera donc perdu

→ Pour éviter un emballement de la réaction, un système d'évacuation d'urgence du produit est prévu. Le capteur de débit d'eau froide $FI33053$ commande la fermeture de la vanne $XV33041$ pour protéger l'échangeur et l'ouverture de la vanne de purge $XV33021$ si le débit d'eau froide maximal permis par l'installation a été atteint.

→ Pour réguler la pression, un système d'asservissement est monté sur le réacteur. Le capteur de pression $PIS33003$ et l'actionneur $PIC33007$:

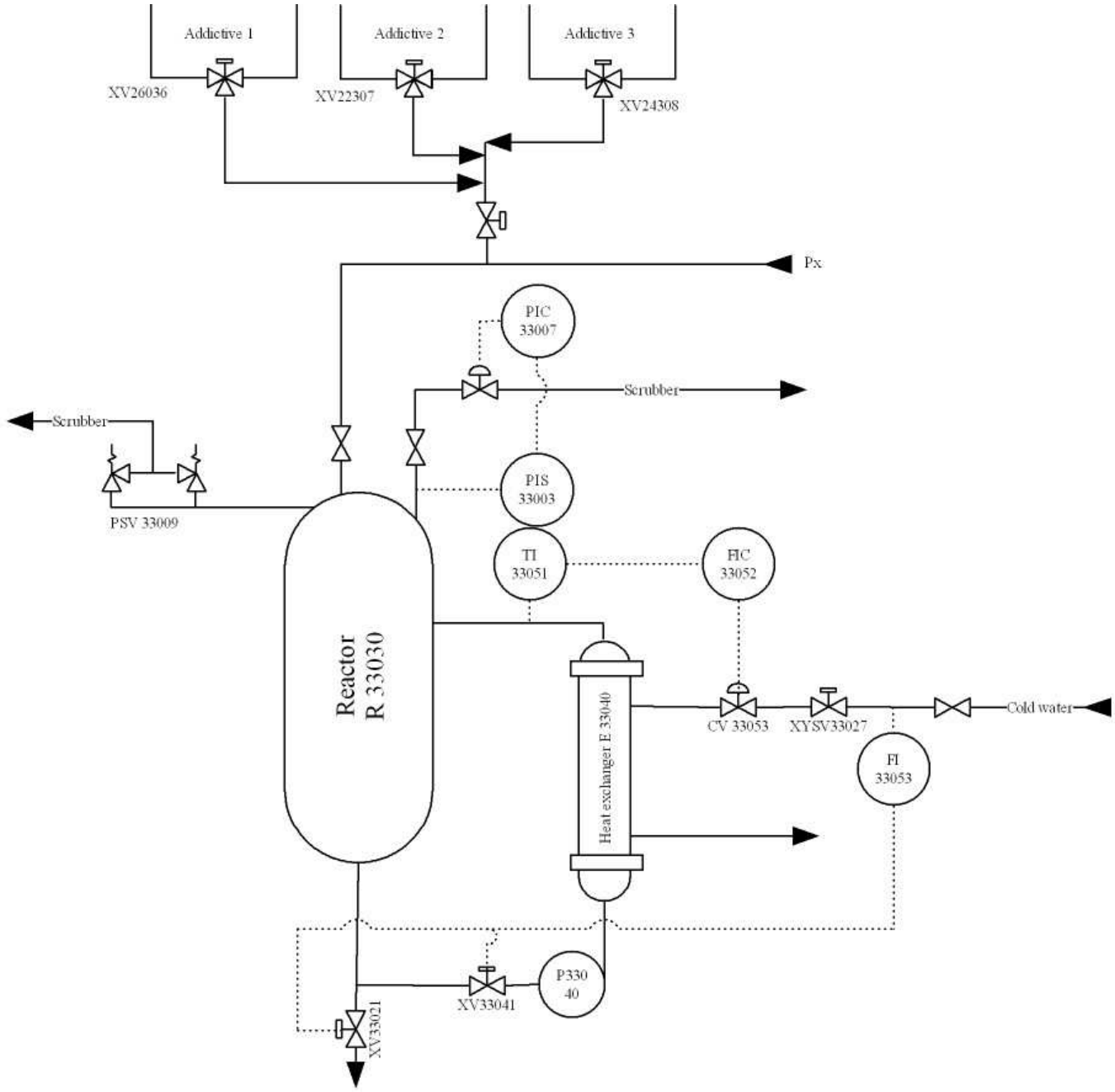


FIG. 5.5 – Schéma du procédé de fabrication exothermique

– Si la pression est trop forte, la vanne associée à l'actionneur $PIC33007$ s'ouvre pour éviter l'explosion du réacteur
– Si la pression est trop faible, le produit fini ne répondra pas aux normes de qualité et sera donc perdu

On peut aussi noter la présence d'une soupape de sécurité mécanique $PSV33009$.

5.2 Analyse de l'application

Dans cette section, un cas concret d'évaluation des risques est proposé. Dans un premier temps, le procédé industriel décrit précédemment sera modélisé structurellement, comportementalement puis fonctionnellement.
Ensuite, à partir d'hypothèses sur les valeurs des variables mesurées, les procédures de diagnostic, pronostic puis mise en sûreté vont être appliquées.
On considérera que les additifs sont déjà présents dans le réacteur. Autrement dit, la partie "Alimentation" du réacteur n'est pas étudiée (ceci pour réduire la complexité des schémas et des analyses).

5.2.1 Modélisation structurelle, fonctionnelle et comportementale

Modélisation structurelle

En analysant l'installation, on constate qu'un découpage en sous-systèmes est possible. Ce découpage correspond aux différentes fonctions globales qui ressortent de l'installation : le siège de la réaction, le circuit de refroidissement, le circuit d'eau froide, l'asservissement en température et en pression ainsi que le système de sûreté en cas de débit d'eau froide trop faible. On distinguera donc les sous-systèmes numérotés :

1. le réacteur $R33030$, les tuyaux sortant dessus et sous le réacteur, ainsi que la soupape de sécurité $PSV33009$
2. la pompe $P33040$, la partie de l'échangeur $E33040$ contenant le milieu réactionnel, les tuyaux allant du réacteur à l'échangeur ainsi que la vanne $XV33041$
3. le circuit d'eau froide avec les vannes $XYSV33027$ et $CV33053$, la partie "eau froide" de l'échangeur ainsi que les tuyaux d'évacuation de l'eau froide réchauffée
4. l'asservissement de purge du réacteur et de protection de l'échangeur en cas de débit d'eau froide insuffisant, i.e. le capteur $FI33053$ et les liaisons avec les vannes $XV33021$ et $XV33041$.
5. l'asservissement en température matérialisé par le capteur $TI33051$ et l'actionneur $FIC33052$ ainsi que les liaisons
6. l'asservissement en pression, représenté par le capteur de pression $PIS33003$ et l'actionneur $PIC33007$
7. les utilités (apport en air, eau, et électricité)
8. l'environnement, composé des bâtiments, de la faune et flore environnantes, des habitations extérieures, des habitants, etc.
9. l'opérateur chargé de superviser le procédé

Ce découpage est représenté en figure 5.6.

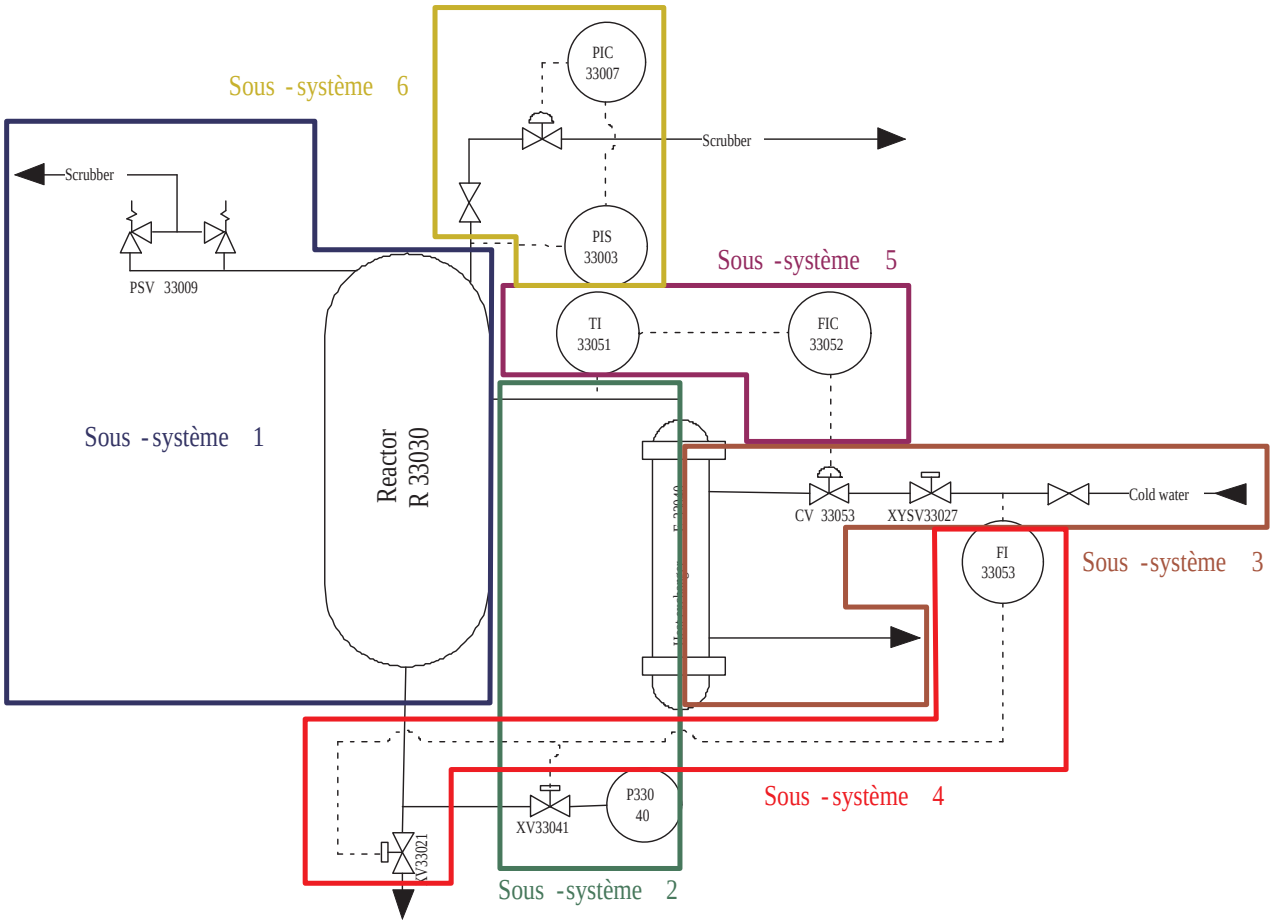


FIG. 5.6 – Découpage en sous-système de l'installation

Modélisation comportementale

La modélisation comportementale proposé dans cette partie s'appuie sur le découpage structurel du point précédent. Les différentes ressources et les variables permettant leur description sont représentés par la figure 5.7 :

- les variables sont représentées par des cercles avec leur intitulés à l'intérieur
- les contraintes décrivant le comportement sont représentées par des rectangles avec l'intitulé des ressources (contraintes détaillées ultérieurement)

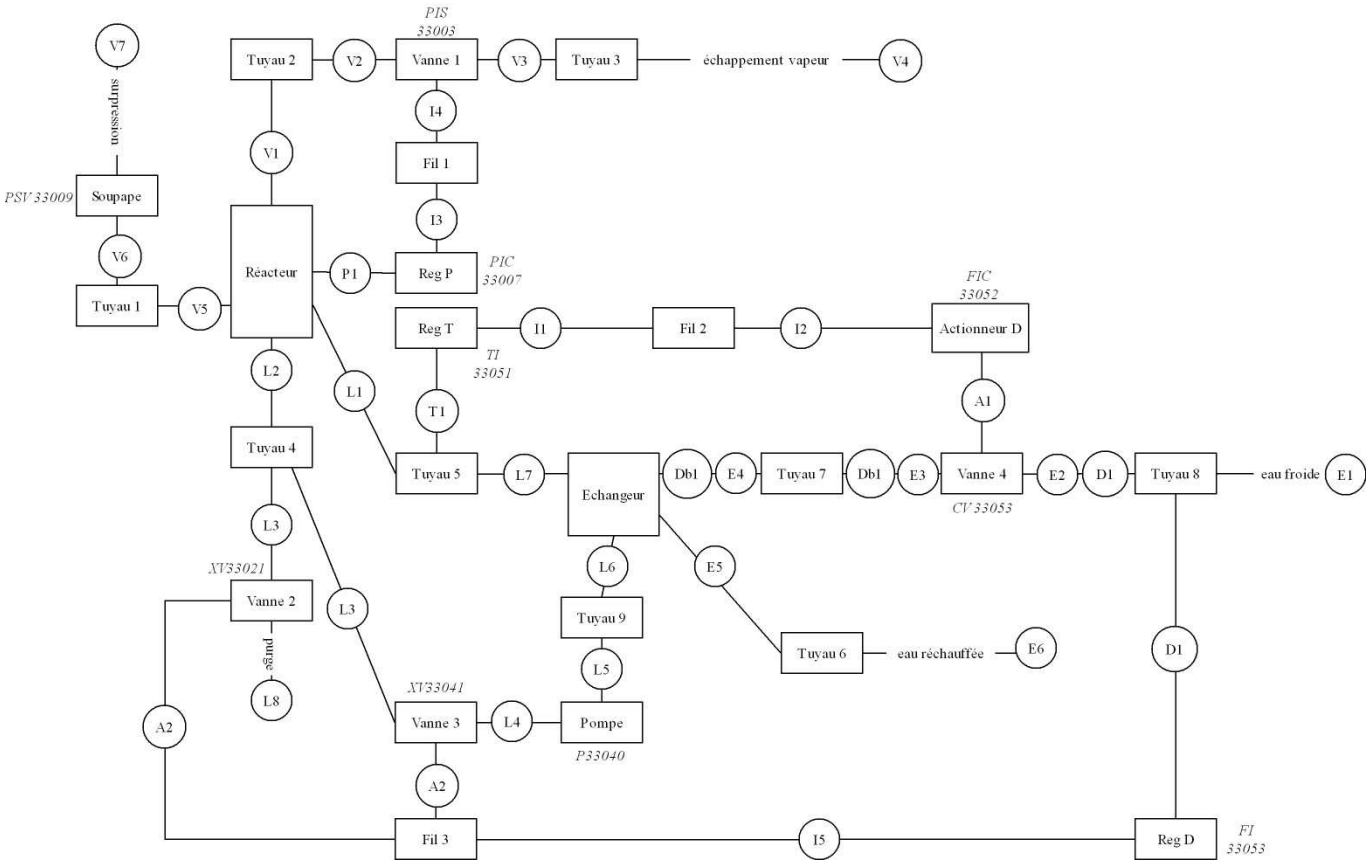


FIG. 5.7 – Variables décrivant le comportement de chacune des ressources de l'application

Les contraintes décrivant les comportements de chacune des ressources pour chacun des sous-systèmes définis précédemment sont représentées par les figures 5.8, 5.9, 5.10, 5.11, 5.12 et 5.13. Pour simplifier les représentations, les modes physiquement impossibles n'ont pas été intégrés aux figures.

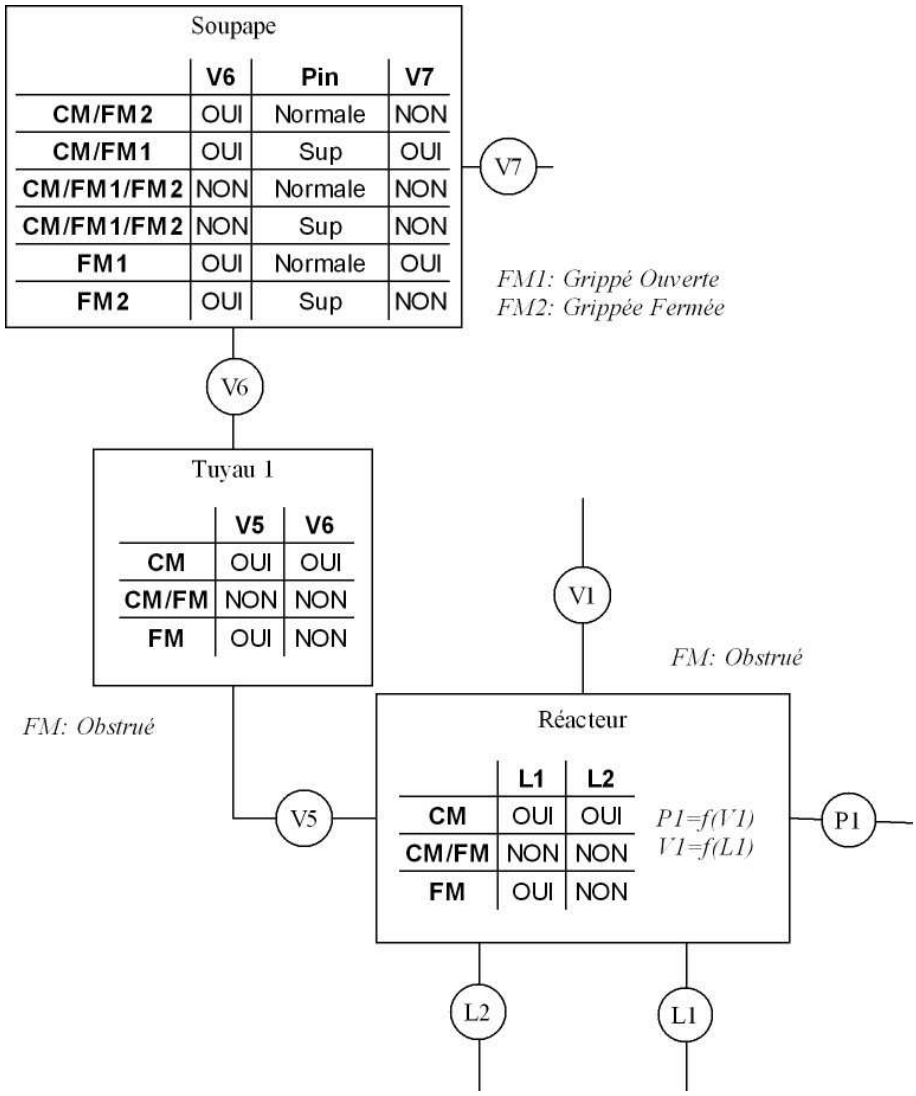


FIG. 5.8 – Représentation des contraintes du sous-système 1

Ainsi la figure 5.8 représente les contraintes des ressources appartenant au sous-système 1 avec les variables suivantes :

- V_j représentant la présence ($V_j = OUI$) ou l'absence ($V_j = NON$) d'un débit d'air
- L_i représentant la présence ($L_i = OUI$) ou l'absence ($L_i = NON$) d'un débit du milieu réactionnel
- P_{in} caractérisant la pression dans le réacteur par rapport au seuil P_{secure} qui est la pression maximale permise par la soupape :
  - $P_{in} < P_{secure} \Rightarrow P_{in} = Normale$
  - $P_{in} \geq P_{secure} \Rightarrow P_{in} = Sup$

La figure 5.9 représente les contraintes des ressources appartenant au sous-système 2 avec les variables suivantes :

- L_i représentant la présence ($L_i = OUI$) ou l'absence ($L_i = NON$) d'un débit du milieu réactionnel
- E_k représentant la présence ($E_k = OUI$) ou l'absence ($E_k = NON$) d'un débit d'eau

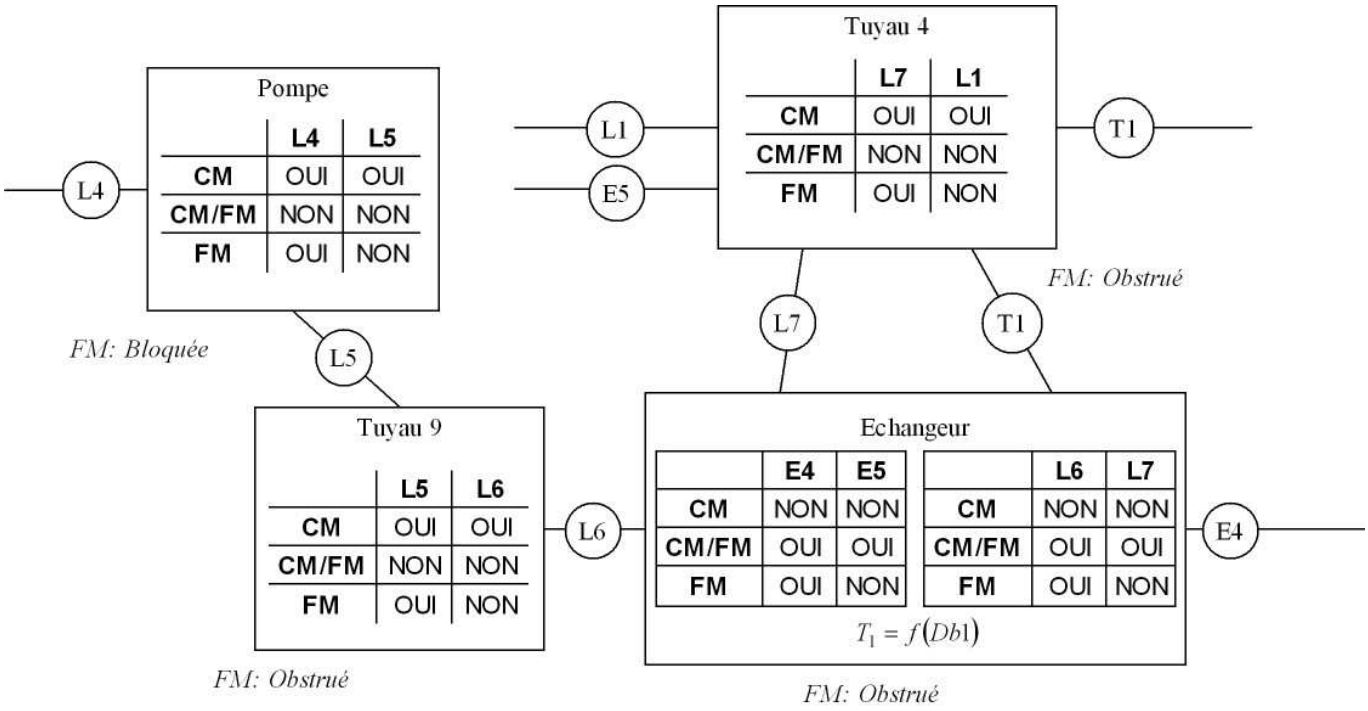


FIG. 5.9 – Représentation des contraintes du sous-système 2

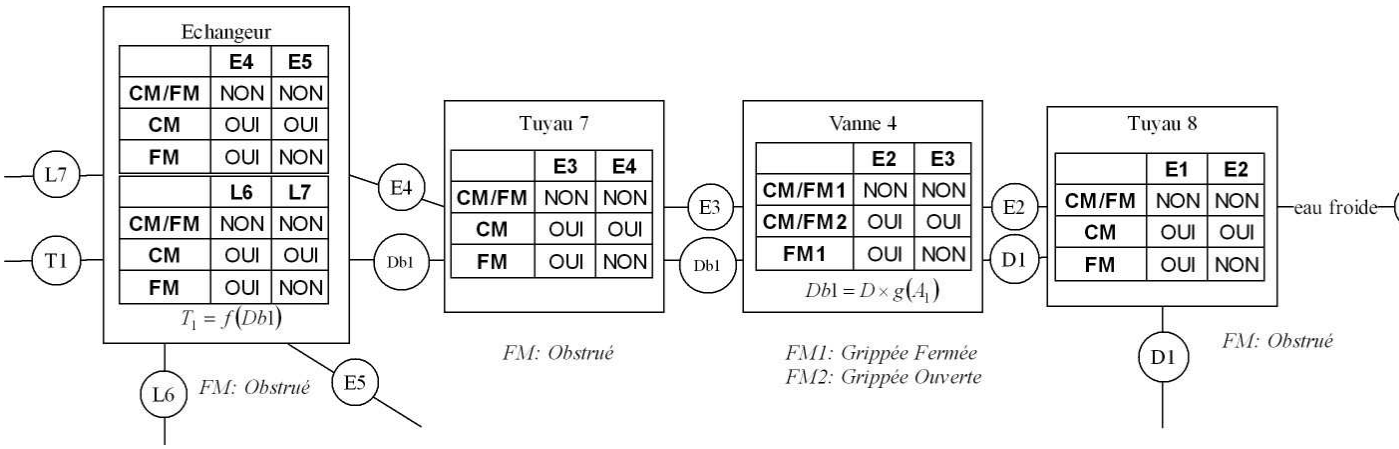


FIG. 5.10 – Représentation des contraintes du sous-système 3

La figure 5.10 représente les contraintes des ressources appartenant au sous-système 3 avec les variables suivantes :

- E_k représentant la présence ($E_k = OUI$) ou l'absence ($E_k = NON$) d'un débit d'eau
- Db_1 caractérise le débit en sortie de la vanne 4. Il est fonction du degré d'ouverture de la vanne 4 et du débit amont D_1

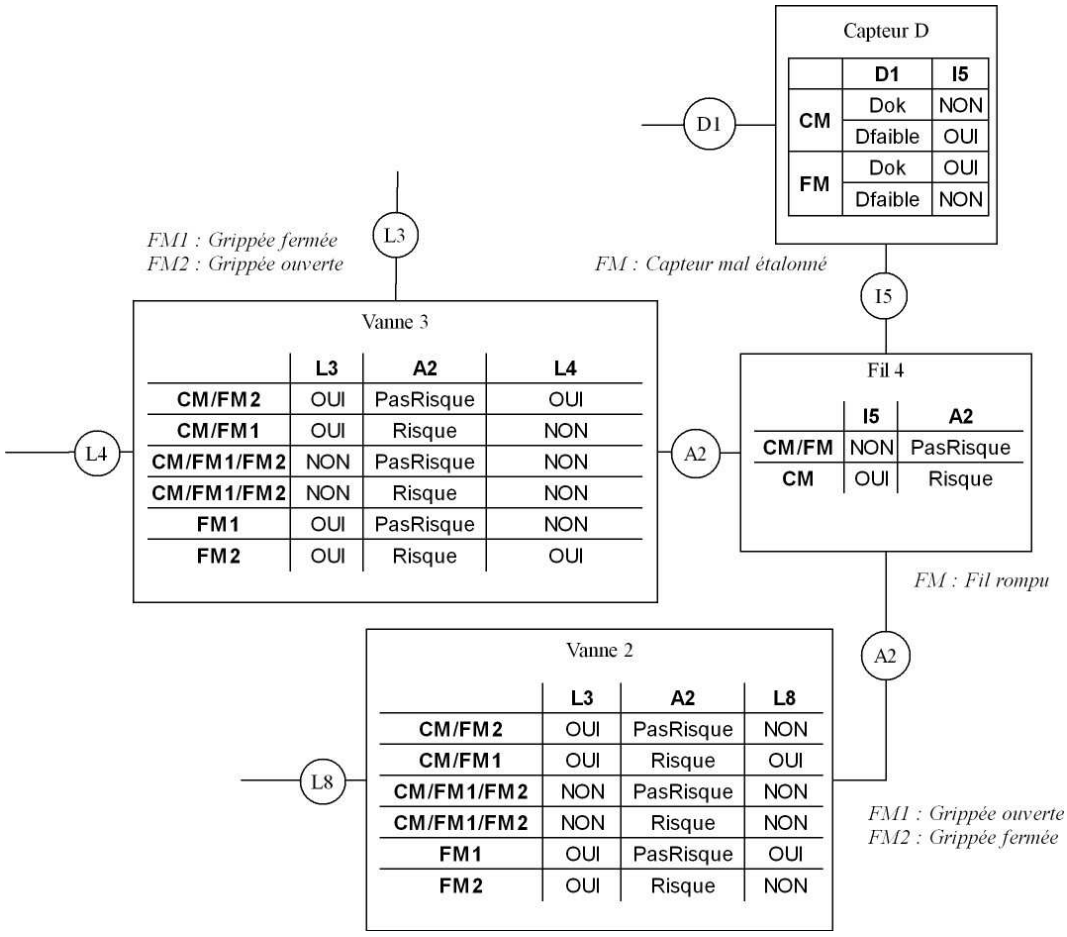


FIG. 5.11 – Représentation des contraintes du sous-système 4

La figure 5.11 représente les contraintes des ressources appartenant au sous-système 4 avec les variables suivantes :

- L_i représentant la présence ($L_i = OUI$) ou l'absence ($L_i = NON$) d'un débit du milieu réactionnel
- I_5 représentant la présence ($I_j = OUI$) ou l'absence ($I_j = NON$) de courant
- D_1 caractérisant le débit d'eau froide par rapport au seuil D_{max} qui est le débit d'eau froide maximal permit par l'installation :
  - $D_1 < D_{max} \Rightarrow D_1 = Dok$
  - $D \geq D_{max} \Rightarrow D_1 = Dfaible$
- A_2 caractérisant la position des la vanne 2 et 3 :

- $A_2 = PasRisque \Rightarrow Vanne\ 3\ fermée\ et\ Vanne\ 2\ ouverte$
- $A_2 = Risque \Rightarrow Vanne\ 3\ ouverte\ et\ Vanne\ 2\ fermée$

La figure 5.12 représente les contraintes des ressources appartenant au sous-système 5 avec les variables suivantes :

- T_1 caractérisant la température du milieu réactionnel par rapport à l'intervalle de température préconisé $[T_{inf}, T_{sup}]$:
  - $T \in [T_{inf}, T_{sup}] \Rightarrow T_1 = Normale$
  - $T < T_{inf} \Rightarrow T_1 = Faible$
  - $T > T_{sup} \Rightarrow T_1 = Forte$
- I_1, I_2 caractérisant l'intensité du courant en sortie du régulateur de température $RegT$ et en sortie du $Fil2$:
  - pour faire baisser la température, le régulateur envoie une intensité $I \in [4mA, 20mA] \Rightarrow I_1 = I+$
  - pour faire augmenter la température, le régulateur envoie une intensité $I \in [-20mA, -4mA] \Rightarrow I_1 = I-$
  - pour ne rien modifier, le régulateur envoie une intensité $I = 1mA \Rightarrow I_1 = 1$
  - enfin, $I = 0mA \Rightarrow I_1 = 0$ si aucune tension n'est présente
- A_1 caractérisant la position de l'action de l'*actionneur D* sur la vanne 4 :
  - $A_1 = PasOrdre \Rightarrow Pas\ de\ changement$
  - $A_1 = Diminue \Rightarrow Diminution\ de\ l'ouverture$
  - $A_1 = Augmente \Rightarrow Augmentation\ de\ l'ouverture$
  - $A_1 = PasAction \Rightarrow Aucune\ action\ engagée$

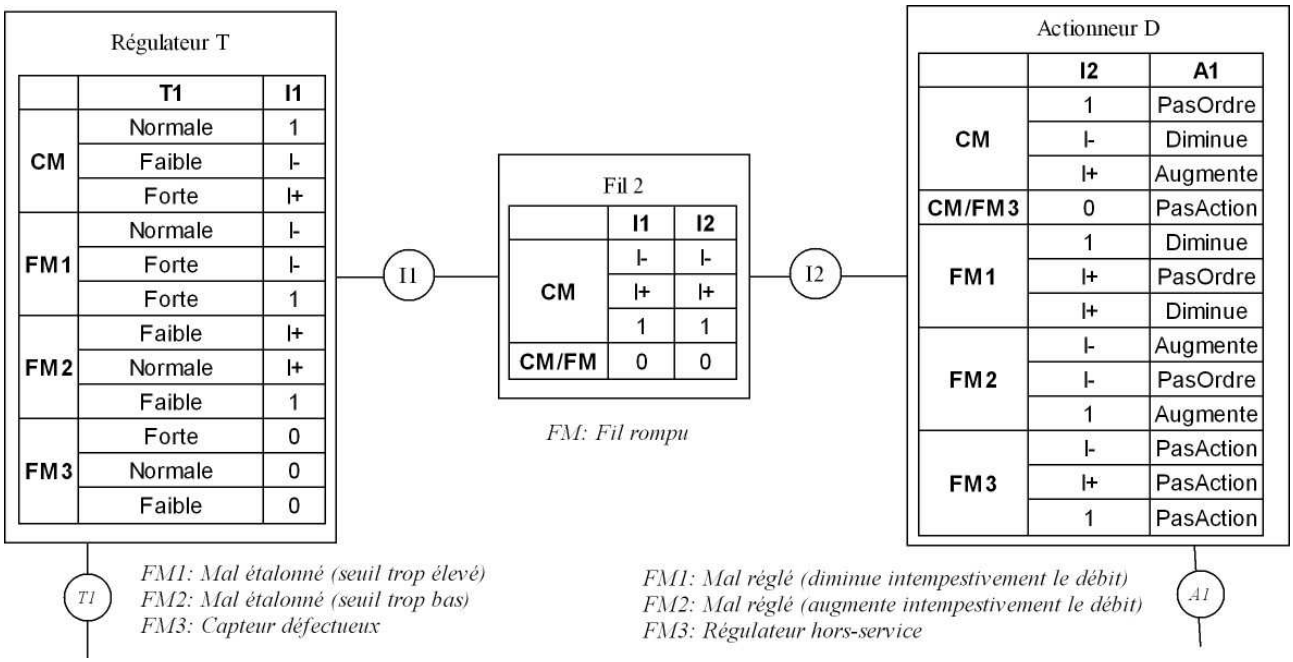


FIG. 5.12 – Représentation des contraintes du sous-système 5

La figure 5.13 représente les contraintes des ressources appartenant au sous-système 6 avec les variables suivantes :

- V_l représentant la présence ($V_l = OUI$) ou l'absence ($V_l = NON$) d'un débit d'air
- P_1 caractérisant la pression dans le réacteur par rapport au seuil P_{max} qui est la pression maximale permise sur le réacteur, sous peine de risque d'explosion :
  - $P_1 < P_{max} \Rightarrow P_1 = Normale$
  - $P \geq P_{max} \Rightarrow P_1 = Trop$
- I_3 et I_4 représentant la présence ($I_j = OUI$) ou l'absence ($I_j = NON$) de courant

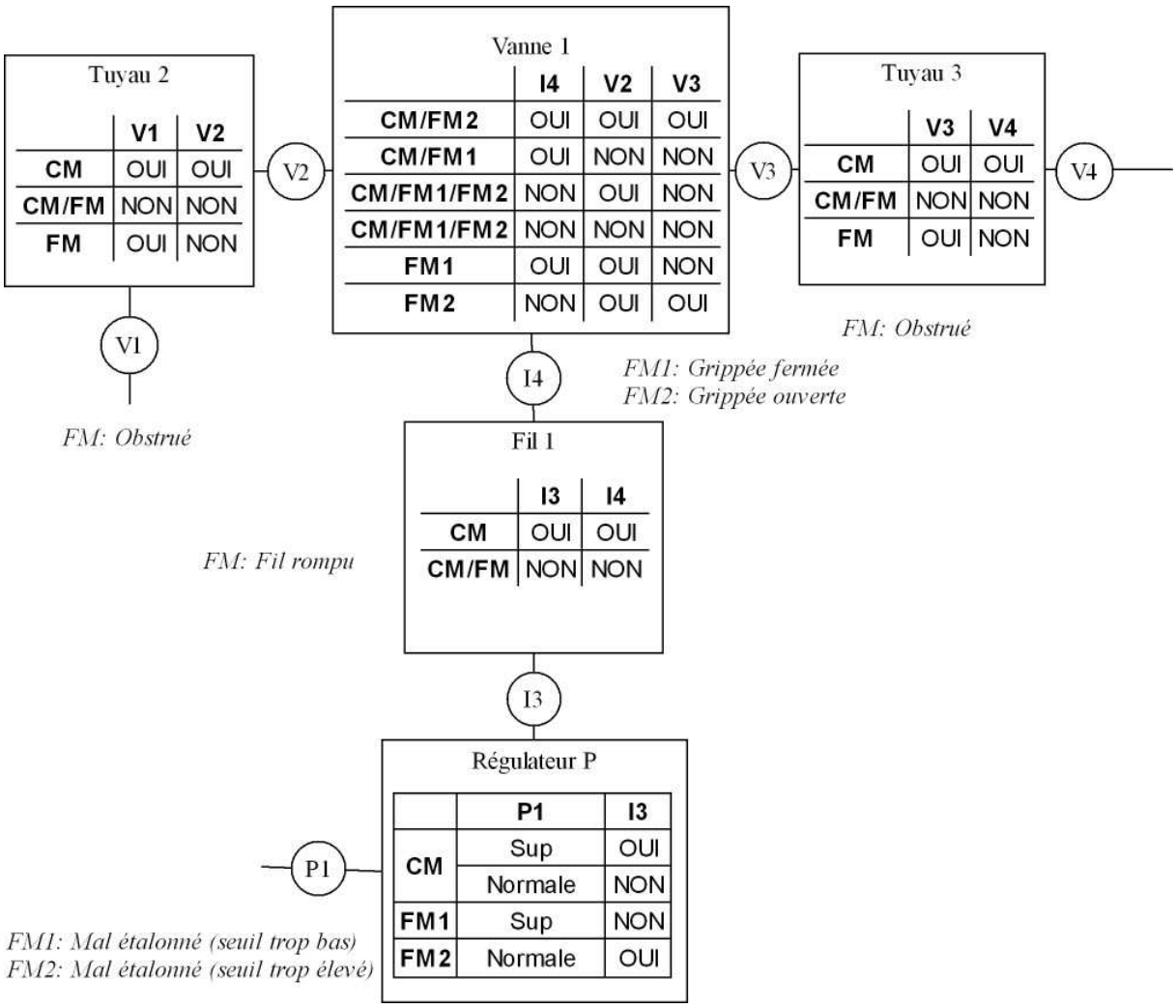


FIG. 5.13 – Représentation des contraintes du sous-système 6

Modélisation fonctionnelle

L'analyse fonctionnelle de l'installation a été réalisée à partir du découpage en sous-systèmes.

Les fonctions sont indexées sous la forme ij où i est le numéro du sous-système et j le numéro de la fonction dans la sous-système.
De la même manière, les modes de défaillance sont indexés sous la forme ijk, avec i le numéro du sous-système, j le numéro de la fonction et k le numéro du mode de défaillance de la fonction.

Pour chacun des sous-systèmes, les fonctions et leurs modes de défaillances associés sont les suivants :

– Pour le **sous-système 1**, composé du Réacteur, de la Soupape, du Tuyau 2, de la Vanne 1 et du Tuyau 3 :

Ressources	Fonctions	Modes de défaillance
Toutes	11 Résister aux chocs	111 Ne pas résister aux chocs
Toutes	12 Résister à la pression	121 Ne pas résister à la pression
Toutes	13 Être étanche	131 Ne pas être étanche
Réacteur	14 Contenir la solution chimique	141 Ne pas contenir la solution chimique
Soupape	15 S'ouvrir en cas de surpression	151 Ne pas s'ouvrir en cas de surpression 152 S'ouvrir intempestivement
Soupape Tuyau 2 Tuyau 3 Vanne 1	16 Laisser passer la vapeur	161 Ne pas laisser passer la vapeur

– Pour le **sous-système 2**, composé de la Vanne 2, de la Vanne 3, de la Pompe, du Tuyau 9, de l'Échangeur et du Tuyau 5 :

Ressources	Fonctions	Modes de défaillance
Pompe	21 Pomper la solution à la bonne vitesse	211 Ne pas pomper la solution 212 Pomper la solution trop vite 213 Pomper la solution trop lentement
Tuyau 9, Tuyau 5	22 Renvoyer la solution dans le réacteur	221 Ne pas renvoyer la solution
Tuyau 9, Tuyau 5	23 Contenir la solution durant le transfert	231 Ne pas contenir la solution
Toutes	24 Être étanche	241 Ne pas être étanche
Vanne 2	25 Purger la solution	251 Ne pas purger la solution 252 Purger la solution intempestivement

– Pour le **sous-système 3**, composé de l'Échangeur, du Tuyau 7, de la Vanne 4, du Tuyau 8 et du Tuyau 6 :

Ressources	Fonctions	Modes de défaillance
Echangeur	31 Refroidir la solution chimique	311 Ne pas suffisamment refroidir la solution 312 Trop refroidir la solution
Tuyau 7, Vanne 4, Tuyau 8	32 Contenir et acheminer l'eau froide	321 Ne pas contenir l'eau froide
Tuyau 6	33 Évacuer et contenir l'eau réchauffée	331 Ne pas évacuer l'eau réchauffée

– Pour le **sous-système 4**, composé du Régulateur D, du Fil 3 et du Capteur D :

Ressources	Fonctions	Modes de défaillance
Régulateur D	41 Mesurer le débit d'eau froide	411 Ne pas mesurer le débit d'eau froide 412 Sur-estimer le débit d'eau froide 413 Sous-estimer le débit d'eau froide
Régulateur D, Capteur D	42 Envoyer les informations à l'opérateur	421 Ne pas envoyer les informations à l'opérateur
Vanne 3	43 Fermer la vanne pour protéger l'échangeur	431 Ne pas fermer la vanne de protection 432 Fermer intempestivement
Vanne 2	44 Ouvrir la vanne de purge en cas de débit d'eau faible	441 Ne pas ouvrir la vanne de purge 442 Ouvrir la vanne de purge intempestivement

– Pour le **sous-système 5**, composé du Régulateur T, du Fil 2 et de l'Actionneur D :

Ressources	Fonctions	Modes de défaillance
Régulateur T, Actionneur D	51 Envoyer les informations à l'opérateur	511 Ne pas envoyer les informations à l'opérateur
Régulateur T	52 Mesurer la température de la solution	521 Ne pas mesurer la température 522 Sur-estimer la température 523 Sous-estimer la température
Actionneur D	53 Réguler le débit d'eau en fonction de la température	531 Ne pas réguler le débit 532 Demander un débit trop fort 533 Demander un débit trop faible

– Pour le **sous-système 6**, composé du Fil 1, Vanne 1, Tuyau 2, Tuyau 3 et du Régulateur P
:

Ressources	Fonctions	Modes de défaillance
Régulateur P	61 Mesurer la pression interne	611 Ne pas mesurer la pression 612 Sur-estimer la pression 613 Sous-estimer la pression
Régulateur P	62 Laisser passer la vapeur	621 Ne laisser passer la vapeur
Vanne 1	63 Réguler la pression dans le réacteur	631 Trop baisser la pression 632 Ne pas baisser la pression

– Pour le **sous-système 7** : les Utilités

Fonctions	Modes de défaillance
71 Alimenter le système en électricité	711 Ne pas fournir d'énergie 712 Fournir trop d'énergie 713 Ne pas fournir assez d'énergie
72 Alimenter le système en eau froide	721 Ne pas fournir d'eau froide

Les résultats de l'analyse AMDEC sont donnés dans les tableaux suivants. Pour simplifier l'écriture, les fonctions ont été remplacées par leur index.

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
11	111 Ne pas résister aux chocs				131 Ne pas être étanche
12	121 Ne pas résister à la pression			Explosion réacteur	
13	131 Ne pas être étanche	Réacteur perforé $FM_1(Reacteur)$			
14	141 Ne pas contenir la solution chimique		Réacteur non étanche 131		
15	151 Ne pas s'ouvrir en cas de surpression	Soupape bloquée fermée $FM_2(Soupape)$	-	Explosion Réacteur	
	152 S'ouvrir intempestivement	Soupape bloquée ouverte			
16	161 Ne pas laisser passer la vapeur	Tuyau 2 bouché $FM_1(Tuyau2)$ Tuyau 3 bouché $FM(Tuyau3)$ Vanne 1 bloquée fermée $FM_1(Vanne1)$		Explosion du réacteur	

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
21	211 Ne pas pomper la solution	Pompe défectueuse $FM(Pompe)$	711 Ne pas fournir d'énergie		221 Ne pas renvoyer la solution 311 Ne pas refroidir la solution
	212 Pomper la solution trop vite	Emballement de la pompe			312 Trop refroidir la solution
	213 Pomper la solution trop lentement	Pompe défectueuse	713 Pas assez de courant		311 Ne pas refroidir la solution
22	221 Ne pas renvoyer la solution	Tuyau 5 bouché $FM(Tuyau5)$	211 Ne pas pomper la solution	Explosion de l'échangeur	
23	231 Ne pas contenir la solution		241 Échangeur non étanche		311 Ne pas refroidir la solution
24	241 Ne pas être étanche	Échangeur bouché $FM(Echangeur)$			231 Ne pas contenir la solution
25	251 Ne pas purger la solution 252 Purger la solution intempestivement		441 Ne pas ouvrir la vanne de purge 442 Ouvrir la vanne de purge intempestivement	Explosion Réacteur	

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
31	311 Ne pas refroidir la solution	Tuyau 8 bouché $FM(Tuyau8)$ Vanne 4 bloquée fermée $FM_1(Vanne4)$ Tuyau 7 bouché $FM(Tuyau7)$ Tuyau 6 bouché $FM(Tuyau6)$ Tuyau 6 bouché $FM(Tuyau6)$	721 Pas d'alimentation en eau froide 211 Ne pas pomper la solution 321 Ne pas contenir l'eau froide 331 Ne pas évacuer l'eau réchauffée 331 Ne pas évacuer l'eau réchauffée 531 Ne pas réguler le débit $(FM_1(ActionneurD))$ 533 Demander un débit d'eau trop faible 412 Sur-estimer le débit d'eau froide 213 Pomper trop lentement	Explosion Réacteur	

			231 Ne pas contenir la solution durant le transfert 432 Fermer la vanne de protection intempestivement		
	312 Trop refroidir la solution	212 Pomper trop vite 532 Demander un débit d'eau trop fort 531 Ne pas réguler le débit ($FM_2(ActionneurD)$)	413 Sous-estimer le débit d'eau froide		
32	321 Ne pas contenir l'eau froide	Échangeur perforé $FM(Echangeur)$ Tuyau 8 perforé Tuyau 7 perforé			311 Ne pas refroidir la solution
33	331 Ne pas évacuer l'eau réchauffée	Tuyau 6 bouché $FM(Tuyau6)$			311 Ne pas refroidir la solution

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
41	411 Ne pas mesurer le débit d'eau froide	Capteur D défectueux	711 Ne pas fournir d'énergie		431 Ne pas fermer la vanne de protection 441 Ne pas ouvrir la vanne de purge
	412 Sur-estimer le débit d'eau froide	Capteur D défectueux			432 Fermer la vanne de protection intempestivement 442 Ouvrir la vanne de purge intempestivement
	413 Sous-estimer le débit d'eau froide	Capteur D défectueux			431 Ne pas fermer la vanne de protection 441 Ne pas ouvrir la vanne de purge
42	421 Ne pas envoyer les informations à l'opérateur	Capteur D défectueux			
43	431 Ne pas fermer la vanne de protection	Régulateur défectueux $FM(CapteurD)$ Vanne 3 bloquée ouverte $FM_2(Vanne3)$	411 Ne pas mesurer le débit d'eau froide 413 Sous-estimer le débit d'eau froide	Explosion de l'échangeur	

	432 Fermer intempestivement	Régulateur défectueux $FM(CapteurD)$ Vanne 3 bloquée fermée $FM_1(Vanne3)$ Fil 3 défectueux $FM(Fil3)$	412 Sur-estimer le débit d'eau froide		311 Ne pas refroidir la solution
44	441 Ne pas ouvrir la vanne de purge	Régulateur défectueux $FM(CapteurD)$ Vanne 2 bloquée fermée $FM_2(Vanne2)$ Fil 3 défectueux $FM(Fil3)$	411 Ne pas mesurer le débit d'eau froide 413 Sous-estimer le débit d'eau froide		251 Ne pas purger la solution
	442 Ouvrir la vanne de purge intempestivement	Régulateur défectueux Vanne 2 bloquée ouverte $FM_1(Vanne2)$	412 Sur-estimer le débit d'eau froide		252 Purger la solution intempestivement

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
51	511 Ne pas envoyer les informations à l'opérateur	Capteur T défectueux			
52	521 Ne pas mesurer la température	Capteur T défectueux	711 Ne pas fournir d'énergie		531 Ne pas réguler le débit
	522 Sur-estimer la température	Capteur T défectueux $FM_1(CapteurT)$			532 Demander un débit trop fort
	523 Sous-estimer la température	Capteur T défectueux $FM_2(CapteurT)$			533 Demander un débit trop faible
53	531 Ne pas réguler le débit	Fil 2 défectueux $FM(Fil2)$	521 Ne pas mesurer la température		311 Ne pas refroidir la solution
	532 Demander un débit trop fort	Actionneur D défectueux $FM_2(ActionneurD)$	523 Sous-estimer la température		312 Trop refroidir la solution
	533 Demander un débit trop faible	Actionneur D défectueux $FM_1(ActionneurD)$	522 Sur-estimer la température		311 Ne pas refroidir la solution

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
61	611 Ne pas mesurer la pression	Capteur P défectueux	711 Ne pas fournir d'énergie	Explosion du réacteur	
	612 Sur-estimer la pression	Régulateur P défectueux $FM_2(RegulateurP)$			631 Trop baisser la pression
	613 Sous-estimer la pression	Régulateur P défectueux $FM_1(RegulateurP)$			632 Ne pas baisser la pression
62	621 Ne pas laisser passer la vapeur	$FM(Tuyau2)$, $FM(Tuyau3)$			
63	631 Trop baisser la pression	Vanne 1 bloquée ouverte $FM_2(Vanne1)$	612 Sur-estimer la pression		
	632 Ne pas baisser la pression	Vanne 1 bloquée fermée $FM_1(Vanne1)$ Tuyau 2 bouché $FM(Tuyau2)$ Tuyau 3 bouché $FM(Tuyau3)$ Fil 1 coupé $FM(Fil1)$	613 Sous-estimer la pression	Explosion du réacteur	

Fcts	Modes de défaillance	Causes		Effets	
		Modes de défaut	Modes de défaillance	Modes de défaut	Modes de défaillance
71	711 Ne pas fournir d'énergie	Panne fournisseur d'énergie Câble rompu			211 Ne pas pomper la solution 411 Ne pas mesurer le débit d'eau froide 521 Ne pas mesurer la température 611 Ne pas mesurer la pression
	712 Fournir trop de courant	Orage			
	713 Ne pas fournir assez de courant				213 Pomper la solution trop lentement
72	721 Ne pas fournir d'eau froide	Canalisation rompue			311 Ne pas refroidir la solution

Pour terminer, les résultats de l'AMDEC sont représentés sous forme de Graphe Causal de Dysfonctionnement, permettant ainsi de visualiser les relations de cause à effets tout en intégrant les informations supplémentaires qu'apportent les portes logiques. La figure 5.14 représente une partie de cette arbre de défaillance.

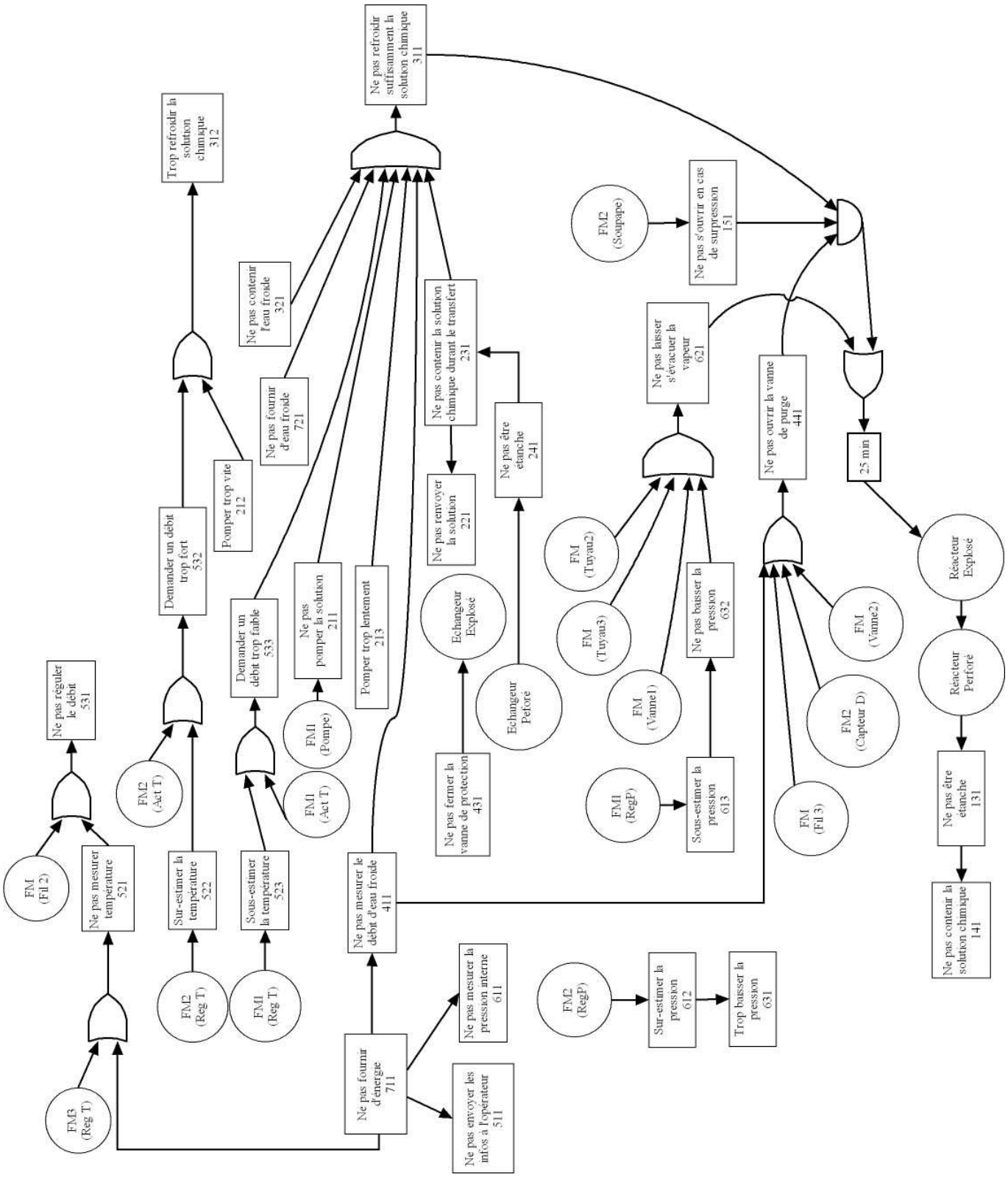


FIG. 5.14 – Extrait de l'arbre de défaillance du système

5.2.2 Diagnostic - Pronostic - Mise en sûreté

On suppose que les variables suivantes sont mesurées : E_1, E_6, V_1, V_4, P_1, T_1, A_1, L_8, L_4, D_1, L_1 et L_3.
En utilisant la méthode des graphes bipartis [Staroswiecki et al., 2000], on trouve les tests donnés dans le tableau 5.1.

TAB. 5.1 – Tests de diagnostics réalisables sur l'application

	Soupape	Réacteur	Echangeur	Reg P	Reg T	Act D	Reg D	Tuyau 1
Test 1					x	x		
Test 2				x				
Test 3							x	
Test 4			x					
Test 5			x					
Test 6		x						

	Tuyau 2	Tuyau 3	Tuyau 4	Tuyau 5	Tuyau 6	Tuyau 7	Tuyau 8	Tuyau 9
1	x	x						
2								
3								
4					x	x	x	
5				x				x
6			x					

	Vanne 1	Vanne 2	Vanne 3	Vanne 4	Fil 1	Fil 2	Fil 3	Pompe
1	x					x		
2					x			
3		x	x				x	
4				x				
5								x
6								

Diagnostic

On considère les informations suivantes :
"La pression dans le réacteur est normale $P_1 = Normale$ et aucun débit de vapeur n'est présent en sortie de l'échappement de régulation de pression $V_4 = NON$. On observe un débit de liquide réactionnel en sortie du réacteur $L_3 = OUI$ ainsi qu'en entrée du réacteur $L_1 = OUI$ et en amont de la pompe $L_4 = OUI$, mais pas en sortie de la vanne de purge $L_8 = NON$. Concernant l'eau froide, un débit est présent en entrée de l'installation $E_1 = OUI$ avec une valeur conforme $D1 = Dok$ et un débit d'eau réchauffée est présent en sortie de l'échangeur

$E_6 = OUI$. Par contre, la température du milieu réactionnel est en hausse et dépasse le seuil T_{sup} ($T_1 = Forte$). Cependant, l'action demandée par l'actionneur D contrôlant le débit d'eau froide est de réduire le degré d'ouverture de la vanne 4 ($A_1 = Diminue$)".

Quelle est l'origine de cette défaillance ? Quelles peuvent être les conséquences de cette défaillance à court et long terme ?
Ces questions légitimes d'un opérateur face à une défaillance trouveront des éléments de réponse en suivant les procédures de diagnostic et de pronostic décrites dans les chapitres précédents.

Étape 1 : Recherche des diagnostics minimaux

L'analyse diagnostique nous conduit à considérer le test 1 du tableau 5.1 comme faux, les autres tests étant vrais.
Les diagnostics minimaux sont donc définis par :

$$D = \{\{RegT\}, \{Fil2\}, \{ActD\}\}$$

Étape 2 : Complétion des diagnostics

En propageant les différents modes de défaut possibles des diagnostics dans le Graphe Causal de Dysfonctionnement (figure 5.14 en page 166) de l'installation, aucun diagnostic incomplet n'apparaît. Ceci provient du fait qu'aucun mode de défaut (valide) n'est la conséquence des diagnostics dans le graphe.

Étape 3 : Elimination des diagnostics physiquement impossibles

- En propageant les différents modes de défaut possibles des diagnostics dans le graphe, on s'aperçoit que le mode de défaut du $Fil2$ ($FM(Fil2)$) conduit au mode de défaillance "Ne pas réguler le débit" $fm_1(RgulerLeDebit)$.
  Or, la contrainte fonctionnelle de la fonction "Réguler le débit" (figure 5.16), issue de l'analyse fonctionnelle de cette même fonction (figure 5.15), montre que les contraintes associées au mode de défaillance fm_1 possèdent toutes la valeur $PasAction$ pour la variable A_1, qui, par hypothèse vaut $Diminue$.
  En conclusion, $\{Fil2\}$ n'est pas un diagnostic cohérent.
  Concernant les autres diagnostics, aucune incohérence n'est relevée.

- En propageant les valeurs des variables mesurées aucun diagnostic physiquement impossible n'est relevé parmi les diagnostics restant.

Les diagnostics restant sont donc :

$$D = \{\{RegT\}, \{ActD\}\}$$

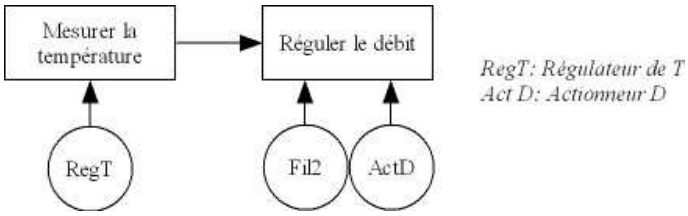


FIG. 5.15 – Analyse fonctionnelle de la fonction "Réguler le débit"

Réguler le débit

Mode	T1	I1	I2	A1
nf	Normale	1	1	PasOrdre
	Faible	I-	I-	Diminue
	Forte	I+	I+	Augmente
	Normale	I-	I-	PasOrdre
	Forte	I-	I-	Augmente
	Forte	1	1	Augmente
	Faible	I+	I+	Diminue
	Normale	I+	I+	PasOrdre
	Faible	1	1	Diminue
fm1	Normale	1	1	PasAction
	Faible	I-	I-	PasAction
	Forte	I+	I+	PasAction
	Normale	I-	I-	PasAction
	Forte	I-	I-	PasAction
	Forte	1	1	PasAction
	Faible	I+	I+	PasAction
	Normale	I+	I+	PasAction
	Faible	1	1	PasAction
	Forte	0	0	PasAction
	Normale	0	0	PasAction
	Faible	0	0	PasAction

Mode	T1	I1	I2	A1
fm3	Forte	1	1	PasOrdre
	Forte	1	1	Diminue
	Normale	1	1	Diminue
	Forte	I-	I-	Diminue
	Normale	I+	I+	Diminue
	Forte	I+	I+	PasOrdre
	Forte	I+	I+	Diminue
	Normale	I-	I-	Diminue
	Forte	I-	I-	PasOrdre
fm2	Normale	I-	I-	Augmente
	Faible	I-	I-	Augmente
	Faible	I-	I-	PasOrdre
	Faible	1	1	PasOrdre
	Faible	1	1	Augmente
	Faible	I+	I+	Augmente
	Faible	I+	I+	PasOrdre
	Normale	1	1	Augmente
	Normale	I+	I+	Augmente

fm1: Ne pas reguler le débit
fm2: Demander un débit trop fort
fm3: Demander un débit trop faible

FIG. 5.16 – Contrainte fonctionnelle de la fonction "Réguler le débit"

Étape 4 : Détermination des modes de défaut

La recherche des modes de défaut s'effectue à nouveau en utilisant la propagation des

valeurs :

– Considérons le premier diagnostic $\{RegT\}$. La propagation des valeurs donnée en figure 5.17 permet de déterminer que le Régulateur de Température est dans le mode de défaut $FM_1(RegT)$ correspondant au mode de défaut "Mal étalonné (seuil trop élevé)".

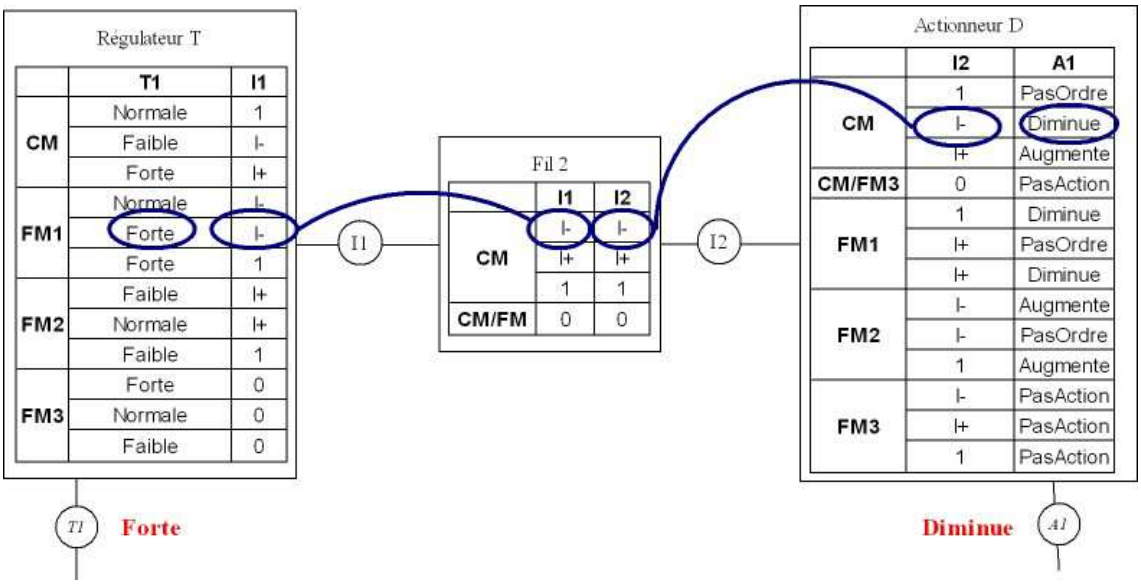


FIG. 5.17 – Affinage du diagnostic $\{RegT\}$

– Considérons maintenant le second diagnostic $\{ActD\}$. La propagation des valeurs donnée en figure 5.18 permet de déterminer que l'actionneur est dans le mode de défaut $FM_1(ActD)$, correspondant au mode de défaut "Mal réglé (diminue intempestivement le débit".

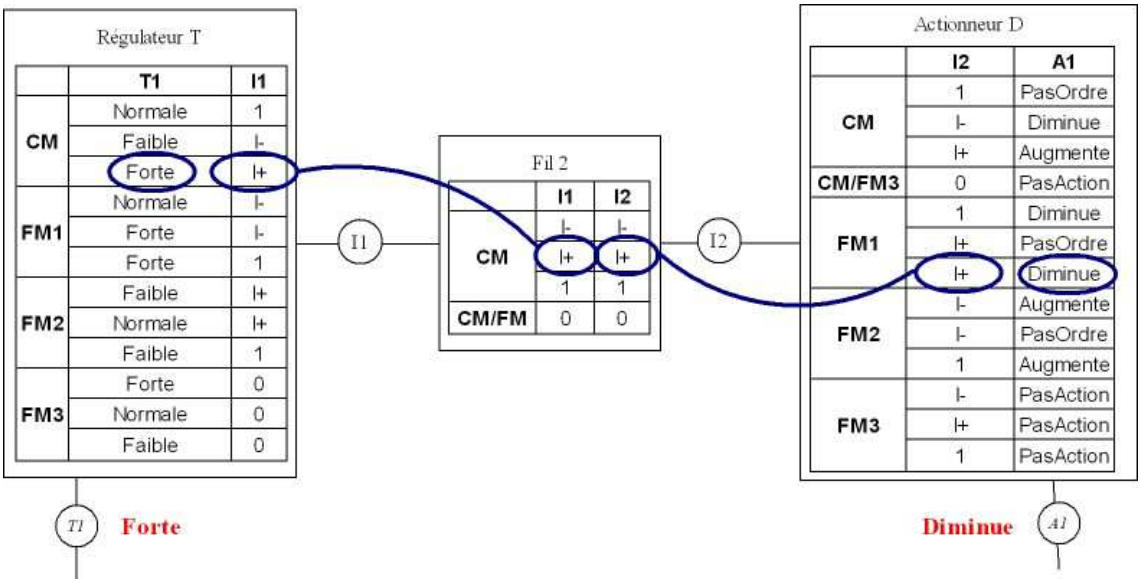


FIG. 5.18 – Affinage du diagnostic $\{ActD\}$

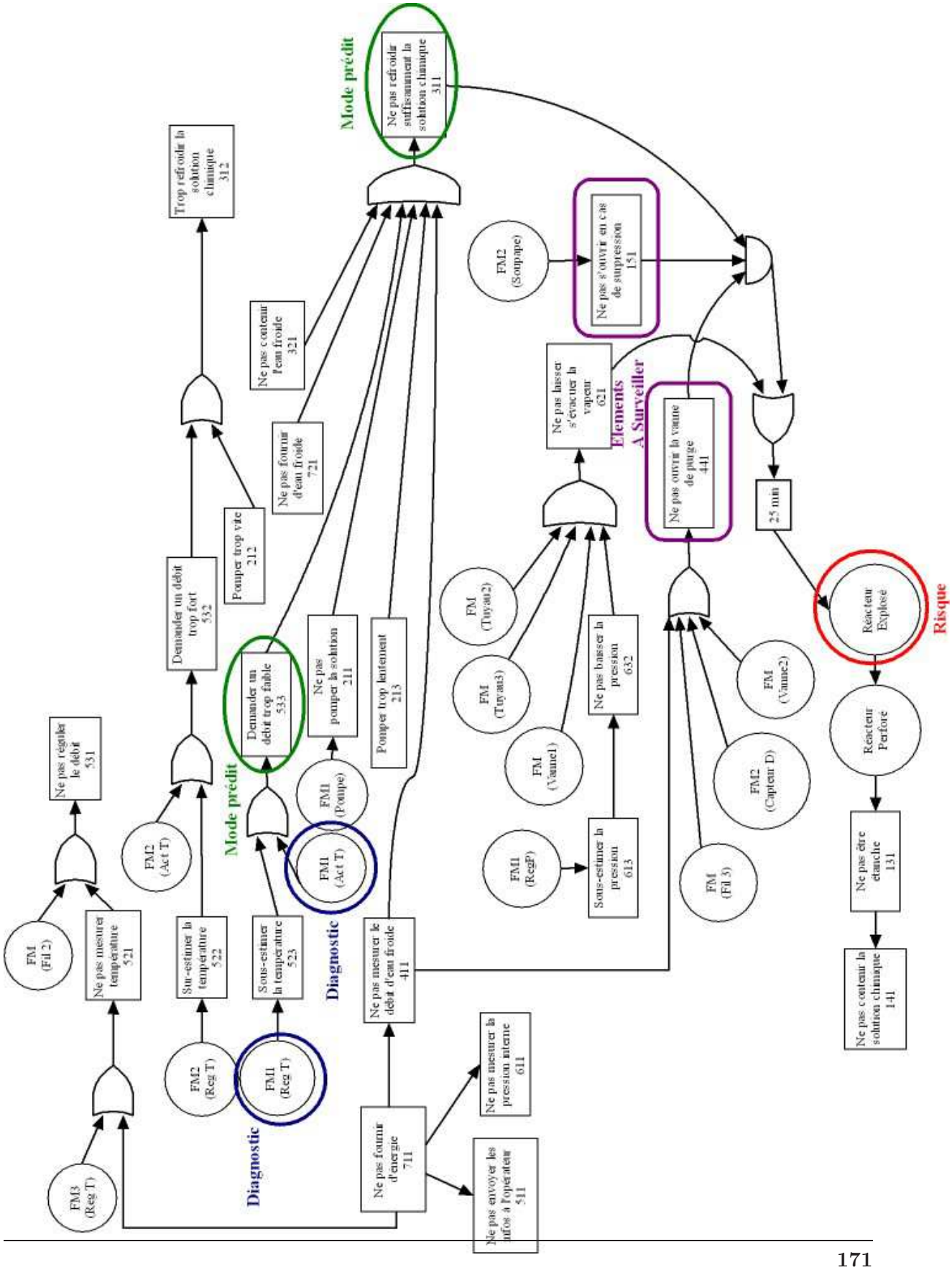


FIG. 5.19 – Recherche des modes prédits et des risques

Pronostic/Mise en sûreté

Supposons que les diagnostics minimaux [Reiter, 1987] et les modes des ressources en défaut [Struss & Dressler, 1989] soient les mêmes que ceux précédemment établis. A partir de ces diagnostics, il est possible de pronostiquer les modes de défaillance et modes de défauts futurs de l'installation. Pour cela, nous allons analyser le Graphe Causal de Dysfonctionnement.
En propageant tour à tour les diagnostics (figure 5.19 en page 171), on s'aperçoit qu'ils conduisent aux mêmes modes, dont voici la liste :

- 522 "Sous-estimer la température"
- 533 "Demander un débit trop faible"
- 311 "Ne pas refroidir la solution chimique"

Ce qui a pour conséquence le risque : "Explosion du réacteur".
Pour éviter une telle conséquence, les éléments suivants du systèmes doivent être opérationnels :

- La purge de la solution (en empêchant l'apparition du mode de défaillance 441 "Ne pas ouvrir la vanne de purge")
- La soupape de sécurité mécanique (en empêchant l'apparition du mode de défaillance 151 "Ne pas s'ouvrir en cas de surpression")

Pour ajouter de l'information à cette analyse, nous allons chercher maintenant à évaluer la probabilité de ce risque, compte tenu des diagnostics.
Le mode de défaillance "Ne pas refroidir la solution chimique" est la conséquence directe des diagnostics. Sa FPE est donc définie par :

$$FPE(NePasRefroidirSolution) = \Big(\ (1, [t_{diag}, +\infty) \ \Big)$$

où t_{diag} est la date où le diagnostic a été effectué.

La FPE du mode de défaillance "Ne pas s'ouvrir en cas de surpression" est la même que celle du mode de défaut $FM_2(Soupape)$ donnée en figure 5.20. En effet, l'expression booléenne est donnée par fm_{151} "Ne pas s'ouvrir en cas de surpression"= $FM_2(Soupape)$.

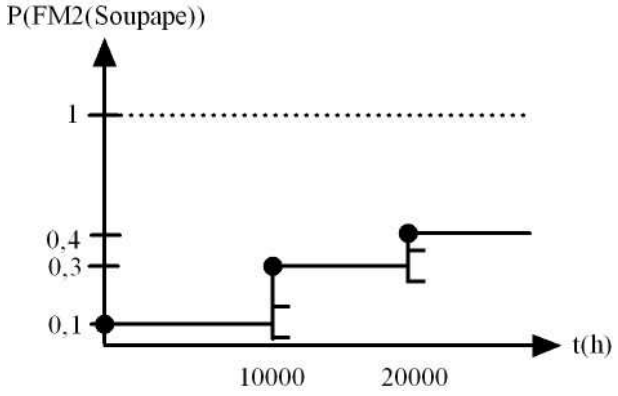


Fig. 5.20 – FPE du mode de défaut $FM_2(Soupape)$

La FPE du mode de défaillance "Ne pas ouvrir la vanne en cas de purge" est donné en figure 5.21. Elle provient de l'analyse de la coupe minimale associée à ce mode, définie par : fm_{444}. "Ne pas ouvrir la vanne en cas de purge"$= FM(Vanne2) + FM_2(CapteurD) + FM(Fil3) + fm711$.

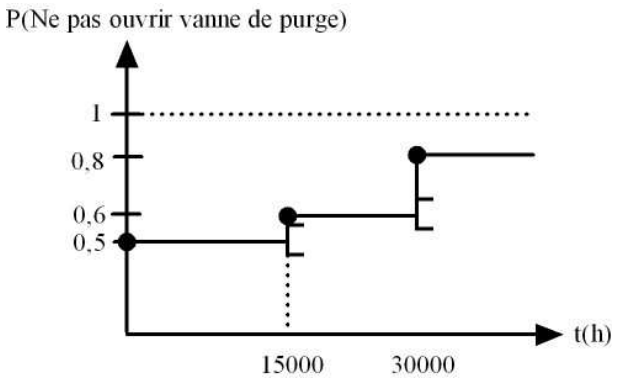


FIG. 5.21 – FPE du mode de défaillance "Ne pas ouvrir la vanne en cas de purge"

En conséquence, le FPE de la conjonction de ces modes est déterminée graphiquement donnée par la figure 5.22 en page 174.

En conclusion, en supposant que la pression est correctement régulée, la probabilité que le réacteur explose, dans ce contexte, est évaluée à $0,18/an$.

5.3 Conclusion

A partir de ce procédé exothermique, les différentes méthodes proposées dans les chapitres précédents ont été appliquées. Grâce aux outils de modélisation du chapitre 2 et des outils de diagnostics proposés dans la chapitre 3, il est possible de déterminer les éléments a priori en défaut en précisant leur mode de défaut et en évitant des diagnostics incohérents et/ou physiquement impossibles. A partir de ces diagnostics, en analysant le Graphe Causal de Dysfonctionnement, il est alors possible de mettre en exergue les futurs modes de défaillance et/ou de défaut du système ainsi que les points du système à surveiller pour éviter ces risques.

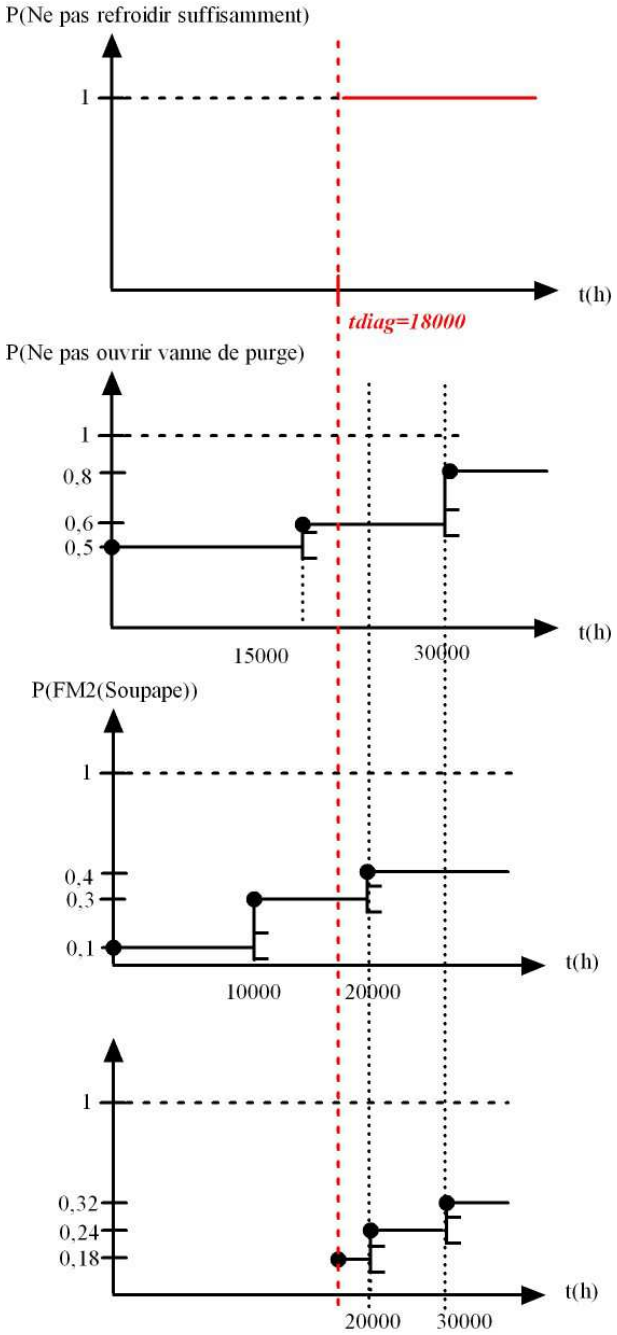


FIG. 5.22 – FPE représentant la conjonction des trois modes amont du mode de défaut "Réacteur explosé"

Conclusion générale et perspectives

Conclusions

L'objectif principal des travaux présentés était de faire le lien entre analyse de risque et analyse diagnostique pour que :

- dans un premier temps, une modélisation commune à ces deux domaines soit proposée
- dans un second temps, les informations issues de l'analyse des risques (ici l'AMDEC) puissent être intégrés dans l'analyse diagnostique en tant que connaissance experte afin de l'affiner
- dans un dernier temps, les résultats de l'analyse diagnostique permettent de pronostiquer les modes de défaut et de défaillance futures du système en les "injectant" dans les résultats de l'analyse des risques (ici l'AMDEC matérialisé sous forme de Graphe Causal de Dysfonctionnement)

Ainsi, en premier lieu (chapitre 1), un parallèle entre les outils d'analyse de risques et les méthodes de diagnostic a été établi pour mettre en avant leurs points communs et leurs différences. Il en est ressorti que l'analyse AMDEC [MIL-STD1629-A, 1983], par les aspects fonctionnels et structurels de sa méthodologie, se présentait comme la méthode d'analyse de risque qui pourrait être intégrée à une analyse diagnostique de type DX [DeKleer & Williams, 1987], privilégiée pour ses capacités à offrir des diagnostics multiples et non exonérés, ce que les méthodes d'analyse de la communauté FDI [Frank, 1996][Iserman, 1997] ne permettent pas.
Ensuite (chapitre 2), des outils de modélisation structurelle, comportementale et fonctionnelle pour établir une base commune pour l'analyse des risques et le diagnostic ont été proposés. Dans cet objectif, ont été développés :

- un modèle structurel basé sur la notion de ressources pouvant être dans un certain nombre de modes possibles (mode normal, mode de défaut, mode physiquement impossible)

- un modèle comportemental basé sur une discrétisation des variables du système. Cette discrétisation nous a permis, en outre, de pouvoir définir un certain nombre de contraintes, qui, si elles sont vérifiées, assurent que la ressource est dans un mode défini. Le comportement d'une ressource reposant sur un certain nombre de variables, nous avons défini des contraintes de comportement comme étant des tuples de ces variables. Par ailleurs, chacun de ces tuple est associé à un mode de comportement de la ressource. On retrouve alors l'idée développée par les approches de diagnostic où une contrainte

associée à une ressource repose sur une ou plusieurs hypothèse(s) sur l'état de la ressource, à savoir ici son mode.

– un modèle fonctionnel qui nous a permis d'étendre le formalisme de l'analyse AMDEC. Contrairement à l'analyse AMDEC originelle où derrière les notions de cause et d'effet de mode de défaillance étaient peu précises, nous avons choisi ici que seuls les modes de défaut et les modes de défaillance pouvaient être des causes et/ou des effets de mode de défaillance. En outre, nous avons ajouté la notion de combinaison de cause et d'effet pour en faire une analyse AMDEC étendue et la notion de cause à effet entre ressources. Un formalisme sous forme de matrice de dysfonctionnement a aussi été développé pour intégrer par la suite les résultats de l'analyse AMDEC informatiquement.

Dans un troisième temps (chapitre 3), nous avons proposé un outil de diagnostic dont les résultats ont permis de faire le lien avec l'analyse des risques. S'appuyant sur les contraintes des différents modèles de comportement et sur leurs modes associés et en intégrant les résultats de l'analyse AMDEC en tant que connaissance experte, des procédures de diagnostic ont été détaillées, dans le but d'affiner les procédures actuelles de diagnostic :

- ► en complétant les diagnostics incomplets
- ► en éliminant les diagnostics physiquement impossible en reprenant le principe de jeux de valeurs physiquement impossible développé dans [Friedrich et al., 1990] et en utilisant les relations causes/effets de l'AMDEC pour rechercher les modes de défaillance conséquents aux diagnostics et vérifier s'ils sont cohérents ou non
- ► en précisant le type de mode de défaut de la ressource en défaut responsable de la détection

D'un autre côté (chapitre 4) un outil de pronostic de modes de défaillances et/ou de modes de défauts a été proposé. Cet outil a été réalisé à partir des méthodes actuelles d'analyse des risques et complété pour permettre de tenir compte de la probabilité d'occurrence des défauts et de leur délai d'apparition par l'introduction de la notion de FPE (Fonction de Probabilité par Episode) associée à chaque événements. Ces FPE permettent de déterminer la probabilité d'occurrence de chaque événement à chaque instant de vie du système. Cet outil repose sur l'analyse du Graphe Causal de Dysfonctionnement, dans lequel sont "propagés" les modes de défaut détectés. Ainsi, il est possible d'établir la liste des risques potentiels du système, leur probabilité et leur délai d'occurrence. Enfin, pour faire face à ces risques, une méthodologie de mise en sûreté a été proposé, analysant le Graphe Causal de Dysfonctionnement pour en tirer les points à surveiller ou les actions à entreprendre pour limiter ces risques, voire les éliminer.

Dans un dernier temps (chapitre 5), nous avons montré que ces résultats pouvaient être appliqués à un réacteur chimique.

Perspectives

Pour compléter ces travaux, deux types de perspectives peuvent être envisagées :

- des améliorations pour parer aux points faibles des méthodes proposées, à savoir par exemple :
  - prendre en compte l'aspect dynamique lors de la recherche de diagnostic et ainsi développer une modélisation comportementale hybride permettant également l'affinage des diagnostics
  - prendre en compte dans la modélisation, non seulement les ressources, mais aussi les produits du système qui, soit transitent, soit sont transformés dans un procédé. La prise en compte des comportements de ces produits ajouterait de l'information intéressante quant à son implication sur le comportement des ressources.
- des approfondissements d'idées, comme par exemple :
  - développer une méthodologie de diagnostic fonctionnel ; en effet, nous avons ici fait le lien entre ressources et fonction. Il paraît alors envisageable de passer d'un diagnostic de défauts à un diagnostic de fonctions
  - introduire une nouvelle porte logique dans le Graphe Causal de Dysfonctionnement capable de définir la probabilité du mode aval en fonction du temps pendant lequel le mode amont était actif. Ce type de porte a un grand intérêt en matière de sécurité, car, par exemple, il permet de définir la probabilité d'explosion d'un container de gaz en fonction du temps pendant lequel il était soumis à un feu.
  - appliquer ces méthodes de diagnostic et pronostic à des systèmes autres que les systèmes physiques. En effet, une modélisation par ressources, fonctions fait penser à la notion de processus de management. Alors pourquoi ne pas développer des outils de diagnostics de processus pour évaluer les risques qu'engendre un défaut d'un processus sur le système ?

Annexes

Annexe 1 : Nomenclature capteurs de sécurité

Dans un système physique donné, les variables sont continues. L'abstraction discrète des variables que nous avons décidé s'explique aussi par une autre raison :

Un système physique que l'on souhaite superviser possède généralement un certain nombre d'organes de mesure (capteurs). La représentation symbolique de ces organes est donnée dans la figure 5.23.

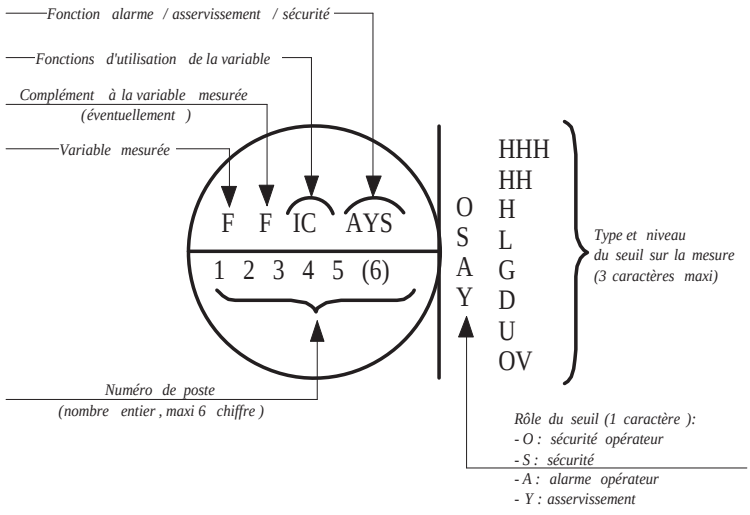


FIG. 5.23 – Représentation graphique d'un organe de mesure

Leur but est d'effectuer un certain nombre d'actions si la variable mesurée est bien dans un intervalle de sécurité. En discrétisant les variables du système, il sera donc possible de déterminer dans lesquels de ces intervalles se trouve la variable.

Par exemple, les points de l'organe de mesure en figure 5.24 signifient :

- P : la variable mesurée est la pression
- IC : les fonctions d'utilisation sont l'indication (I) et la régulation (C) de la pression
- XH : seuil impliquant une action automatique (X) dans le cas d'un niveau haut (H) de

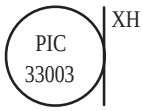


FIG. 5.24 – Exemple de représentation d'un organe de mesure

la pression

Annexe 2 : Implémentation / Approche logicielle

A partir des algorithmes d'analyse diagnostique présentés précédemment, une ébauche logicielle a été entreprise pour expérimenter ces algorithmes. Cette expérimentation a un double but :

- déterminer les entrées et les sorties des algorithmes et leurs formes
- étudier la faisabilité d'une réelle implémentation logicielle pour l'industrie, autrement dit, de quelles informations a-t-on besoin pour lancer les algorithmes et sous quelle forme la plus simple possible peuvent-ils être intégrés au logiciel ?

Entrées/Sorties du programme

En entrée du programme, nous avons besoin :

1. D'un point de vue conception de l'installation cible :
   - d'un modèle structurel reposant simplement sur les ressources du système
   - d'un modèle comportemental comprenant pour chaque ressource :
     - les variables décrivant le comportement du constituant
     - les contraintes de comportement et leurs modes associés
   - d'un modèle fonctionnel, basé sur une analyse AMDEC et représentée sous la forme d'un arbre de défaillance composé de mode de défaut et de modes de défaillance

2. En termes de données sur le système :
   - d'une liste de diagnostic
   - d'une liste de valeurs de variables (qui sont à l'origine de ce diagnostic)

En sortie du programme, nous souhaitons obtenir une liste de mode de défauts expliquant les valeurs des variables mesurées (affinage du diagnostic).

Nous avons choisi comme langage de programmation un langage orientée objet pour sa rigueur de conceptualisation et pour sa présence répandue dans les industries. La section suivante va donc détailler les différents objets intervenant dans l'approche logicielle, leurs paramètres ainsi que leurs fonctions.

Structure des objets

Pour répondre à ces besoins, les objets suivants vont être définis :

- un objet `System`
- un objet `Ressource` représentant les ressources
- un objet `ConstraintTable`, associé à chaque ressource et représentant ses contraintes de comportement
- un objet `Function`
- un objet `FaultTree` décrivant les relations de cause à effet de l'arbre de défaillance

– un objet `DiagAnalysis` permettant d'affiner le diagnostic issu de l'objet `DXresult` en éliminant les diagnostics physiquement impossibles et en spécifiant les modes de défauts.

L'objet `System`

Cet objet représente le système analysé.
Il contient :

– la liste des fonctions du système `Liste[] listeFonctions`
– la liste des ressources du système `Liste[] listeRessources`
– la liste des variables du système `Liste[] listeVar` ainsi que leur domaines de valeurs `Liste[] listeVal`

Le système est défini grâce à son constructeur `System(Liste[] listeFonction, Liste[] listeRessources, Liste[] listeVar, Liste[] listeVal)`

L'objet `Ressource`

Cet objet représente, comme son nom l'indique, une ressource du système.
Il est composé des paramètres suivants :

– `Chaîne name`, correspondant au nom de la ressource
– `ConstraintTable constraintList`, correspondant aux contraintes
– `Liste[] faultModeList` qui est la liste des libellés des modes de défaut de la ressource ainsi que leurs notations (FM_i)

Il est composé également d'un constructeur `Ressource(Chaîne name)` qui permet de définir uniquement le nom, les contraintes étant définis automatiquement dans la suite.
Il contient également la fonction `void addFM(Chaîne FM)` qui permet d'ajouter un mode de défaut à la ressource.

L'objet `ConstraintTable`

Cet objet est un sous objet de `Ressource` car chaque ressource est définie par une table de contrainte décrivant son comportement.
Cet objet a pour paramètres :

– `int nombreModes` qui est le nombre de modes de la ressource
– `ListeChaîne listeVar` qui est la liste des variables impliquées dans les contraintes
– `TableauChaîne table` qui contient les données des contraintes

Il contient un constructeur `ConstraintTable(Fichier fichier)` importe les contraintes existantes d'un fichier.

Cet objet possède également comme fonctions :

- `set(ListeChaîne[] contrainte)` qui insère une nouvelle contrainte
- `check()` qui vérifie la cohérence des contraintes (valeurs des variables)
- `nombreFM()` qui renvoie le nombre de modes de défaut
- `numColonneVar(Chaîne var)` qui renvoie le numéro de la colonne ou se trouve la variable var dans la table des contraintes
- `selectModeswhere(Liste[] valeurs)` qui la liste des modes compatibles en fonctions des valeurs des variables données en argument (du type "A=0","B=1", etc.)
- `nombreVar()` qui renvoie le nombre de variables impliquées dans la contrainte
- `nombreModes()` qui renvoie le nombre de modes dans la contrainte
- `affiche()` qui affiche la table des contraintes

L'objet `Graph`

Cet objet va permettre d'utiliser les relations de cause à effet provenant d'un Graphe Causal de Dysfonctionnement. De manière formelle, un Graphe Causal de Dysfonctionnement est en réalité un graphe biparti $G = (S, A)$ (Figure 5.25) avec :

- S, l'ensemble des sommets répartis en deux partitions :
  - P, le sous-ensemble contenant les portes (ET, OU, NON et DELAI)
  - M, le sous-ensemble contenant les modes de défaillance et les modes de défauts
- A, l'ensemble des arêtes orientées

où :

- $FM_j(r)$ est le mode de défaut j de la ressource r
- $fm_k(f)$ est le mode de défaillance k de la fonction f

Ainsi, le résultat de l'AMDEC est donc un graphe biparti $G = (S, A)$ tel que

$$\begin{cases} P \subset S \\ M = \left(\bigcup_{r \in \Pi} FM_j(r) \right) \cup \left(\bigcup_{f \in \Phi} fm_k(f) \right) \subset S \end{cases}, P \oplus M = S$$

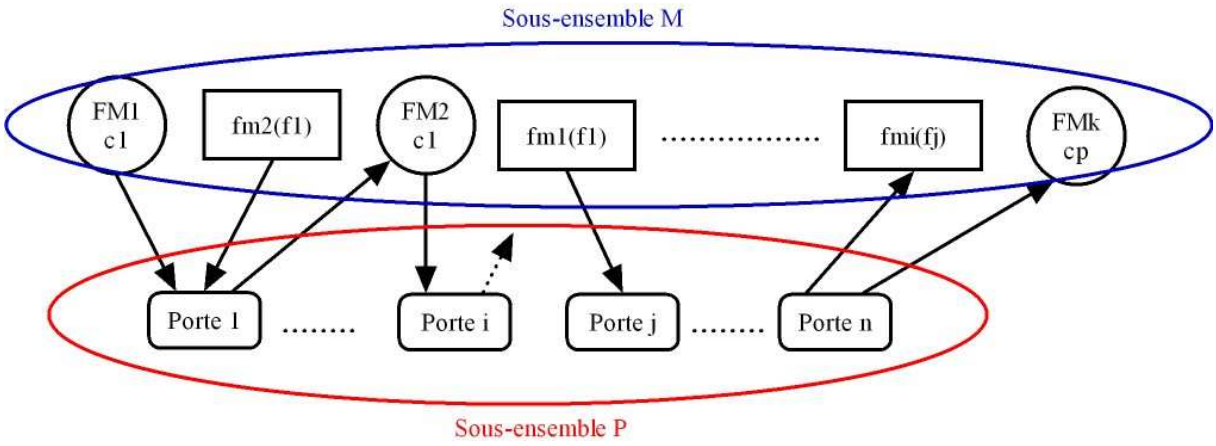


FIG. 5.25 – Représentation formelle d'un arbre de défaillance sous forme d'un graphe biparti

Ainsi, le Graphe Causal de Dysfonctionnement sera représenté par une liste de triplet du type : `[mode départ (cause), mode arrivé (conséquence), porte traversée]`.

L'objet `FaultTree` a pour paramètres :

- `TableauChaîne[][]` ft qui contient la liste des relations cause/effet

Ses constructeurs sont les suivants :

- `Graph ()` qui crée un arbre de défaut vierge
- `Graph (Fichier fichier)` qui importe un arbre de défaillance existant dans un fichier

Il possède également les méthodes suivantes :

- `setRelation (Chaîne relation)` qui ajoute une relation de type cause/effet, une relation étant un triplet du type `[mode, mode, porte]`
- `check()` qui vérifie la cohérence de l'arbre en recherchant si les modes de défauts et les modes de défaillance existent réellement
- `effetValide(Liste[] listeMode)` qui retourne les effets valide (au sens booléen) des éléments d'une liste de modes donnés en argument
- `affiche()` qui permet d'afficher le tableau des relations causes/effets

EXEMPLE :

Le tableau 5.2 représente l'arbre de défaillance donnée dans la figure 5.26.

TAB. 5.2 – Exemple d'implémentation logicielle d'un arbre de défaillance

Cause	*Effet*	*Porte*
m1	m3	OR
m2	m3	OR
m1	m4	OR
m2	m4	OR
m4	m6	AND
m5	m6	AND

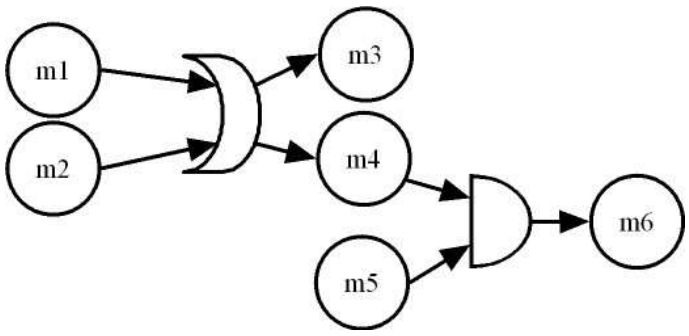


FIG. 5.26 – Exemple d'implémentation logicielle d'un arbre de défaillance

L'objet `DiagAnalysis`

L'objectif de cet objet est d'intégrer les résultats d'une analyse diagnostique préalable ainsi que les tests qui ont permis d'établir ces tests.

Les paramètres de cet objet sont les suivants :
- `System s` qui est le système analysé
- `Liste[] diagSet` qui représente liste des diagnostics issus de l'analyse avec l'arbre HS
- `Liste[] testSet` qui liste des tests utilisés (un test étant une liste de ressources)
- `Graph G` qui est l'arbre de défaillance du système
- `Liste[] listeValVarMes` qui est la liste des valeurs des variables issues des capteurs et des actionneurs (données du type "A=1", "B=6",etc.)

Il a pour constructeur la méthode `DiagAnalysis(System s, Liste[] diagSet, Liste[] testSet, FaultTree ft, Liste[] listeValVarMes)`.

Il possède également les méthodes suivantes :

- `removeImpossibilities()` qui élimine les diagnostics physiquement impossibles en deux étapes : en utilisant l'arbre de défaillance et en propageant les valeurs des variables mesurées
- `affinage()` qui cible les modes de défaut des diagnostics
- `listOfCompWhereIsInvolvedVar(Chaîne var, Liste[] test)` qui renvoie la liste des ressources appartenant à un test où une variable donnée est impliquée dans leurs contraintes de comportement
- `varTest (Liste[] test)` qui renvoie la liste des variables impliquées dans le test, i.e. impliquées dans les contraintes de comportement des ressources impliquées dans ce test
- `selectModeswhere(Liste[] valeurs, ConstraintTable tab)` qui recherche la liste des mode possibles en fonction des valeurs de variables données dans une table de contraintes d'une ressource
- `numColonneVar(Chaîne var, ConstraintTable tab)` qui renvoie le numéro de la colonne de la table de contrainte d'une ressource où se trouve la variable

Exemple

L'exemple choisi pour la simulation est celui du rétroprojecteur (section 3.2.5 en page 99). Nous considérons les mêmes hypothèses que celles de l'exemple utilisé dans le chapitre 3 :

- les variables mesurées sont P, A, L et V
- les tests disponibles sont donnés par le tableau 5.3

	Prise	Inter	Amp	Vent
Test 1	x	x	x	
Test 2	x	x		x
Test 3			x	x

TAB. 5.3 – Tests établis en utilisant les graphes bipartis pour le cas 1

Considérons ensuite les mêmes hypothèses sur les valeurs des variables que celles de l'exemple théorique : $P = IN, A = OFF, L = 1, V = 1$.
Les tests 1 et 2 sont faux, et le test 3 est vrai.
Les diagnostics minimaux préalables sont donc les suivants :

$$\{\{Prise\}, \{Int\}, \{Amp, Vent\}\}$$

La simulation annonce le mode de défaut $FM_1(Inter)$, après un temps de calcul de $t < 1s$. Ce qui confirme la réponse théorique.

Bibliographie

AFNOR [1986], Techniques de l'analyse de la fiabilité des systčmes - procédures d'analyse des modes de défaillances et de leur effets (amde), Norme nf x 60 510.

AFNOR [1991], Sécurité des machines. notions fondamentales, principes généraux de conceptions, partie 1, Norme nf en 292-1.

AFNOR [1997], Sécurité des machines - principes pour l'appréciation du risque, Norme nf en 1050.

ARAMIS : Accidental Risk Assessment Methodology for IndustrieS in th context of the Seveso II directive [2004], Technical report.
URL: *http ://aramis.jrc.it*

Baracchini, P. & Thalmann, P. [2005], *Guide de la mise en place du management environnemental selon ISO 14001*, ISBN 2880746191, PPUR Editions.

BARPI [n.d.], 'Bureau d'analyse des risques et des pollutions industrielles - ministčre de l'ecologie et du développement durable - aria.ecologie.gouv.fr'.
URL: *http ://aria.ecologie.gouv.fr*

Bishop, C. [1994], 'Neural networks and their applications', *Revue Science Instrument* **65**(6), 1803–1832.

Boéri, D. [2003], *Maîtriser la qualité : tout sur la certification et la qualité totale : les nouvelles normes ISO 9001, v.2000*, Paris : Maxima.

Boyd, M. [1991], Dynamical fault tree models : Techniques for analysis of advanced fault tolerant computer systems, PhD thesis, Duke University, USA.

Brand, S. [2004], *Rule-Based Constraint Propagation - Theory and Applications*, ISBN : 90-6196-526-8.

Brémaud, P. [1999], *Markov Chains Gibbs Fields, Monte-Carlo simulation, and Queues*, Springer-Verlag, Berlin.

Cabarbaye, A. & Ngom, L. [2001], Simulation des arbres d'événements, *in* 'Qualita'.

Clifton, A. [2005], *Hazard Analysis Techniques for System Safety*, Wiley.

ClubDesCertifiésDuMFQ [1994], *Guide d'application des normes ISO 9001-9002-9003 étendu à la rédaction du manuel qualité (version 1994)*, ISBN 2909430316, Nanterre : Mouvement Français pour la Qualité.

Cordier, M., Dague, P., Dumas, M., Lévy, F., Montmain, J., Staroswieky, M. & Travé-Massuyès, L. [2000], A comparative analysis of ai and control théory approaches to model-based diagnosis, *in* '14th European Conference on Artificial Intelligence, Berlin, Allemagne'.

Dague, P. & Travé-Massuyès, L. [2004], 'Raisonnement causal en physique qualitative', *Intellectica* (38), 247–290.

DeKleer, J. & Brown, J. [1983], *Assumptions and ambiguities in mechanistic mental models*, Erlbaum Publishers.

DeKleer, J. & Brown, J. [1984], 'A qualitative physics based on confluences', *Artificial Intelligence* **24**, 7–83.

DeKleer, J. & Williams, B. [1987], 'Diagnosis multiple faults', *Artificial Intelligence* **32**, 97–130.

DeKleer, J. & Williams, B. [1989], Diagnosis with behavioral modes, *in* 'Proceedings of the 11th IJCAI Conference, Detroit', pp. 1324–1330.

Désinde, M., Flaus, J. & Ploix, S. [2006*a*], Outil et méthodologie pour l'évaluation des risques de procédé en temps réel, *in* 'Lambda-Mu 15 / Lille'.

Désinde, M., Flaus, J. & Ploix, S. [2006*b*], Risk analysis and diagnosis modelling for online control of process, *in* 'ESREL, Estoril, Portugal'.

Désinde, M., Flaus, J. & Ploix, S. [2006*c*], Risks analysis a help to real-time risks control, *in* 'AISS, Nice, France'.

Dugan, J., Bavuso, S. & Boyd, M. [1992], 'Dynamic fault tree models for fault tolerant computer systems', *IEEE Trans. Reliability* **41**(3), 363–377.

Düstegör, D., Cocquempot, V., Staroswiecki, M. & Frisk, E. [2004], Isolabilité structurelle des défaillances - application à un modèle de vanne, *in* 'JESA'.

Farmer [1967], Siting criteria - a new approach, *in* 'Symposium Containment and Siting of Nuclear Power Plants, International Atomic Energy Agency, Wieden'.

Feray-Beaumont, S. [1989], Modèle qualitatif de comportement pour un système d'aide à la supervision des procédésle qualitatif de comportement pour un systčme d'aide ŕ la supervision des procédés, PhD thesis, Institut Polytechnique de Grenoble, Grenoble, France.

Feray-Beaumont, S., Leyval, L. & Gentil, S. [1989], 'Declarative modelling for process supervision', *Artificial Intelligence* **34**, 135–150.

Flaus, J. [2001], Une formalisation explicite de l'état et des flux dans la méthode mosar, *in* 'Groupe Français en Génie des Procédés (GFGP)'.

Flaus, J. [2003], Un modèle de danger unifié pour l'analyse systémique des risques, *in* 'Groupe Français en Génie des Procédés (GFGP)'.

Flaus, J. [2004], *Cours de sécurité générale, filière PRIHSE*, Grenoble.

Flaus, J., Adrot, O. & Désinde, M. [2006], A mixed structural/functional graph based model for fault diagnosis and systemic risk analysis, *in* 'ESREL, Estoril, Portugal'.

Flaus, J. & Granddamas, O. [2002], Towards a formalisation of mads, system failure analysis model, *in* 'Lambda-Mu 13/ESREL'.

Frank, P. [1996], 'Analytical and qualitative model-based fault diagnosis - a survey and some new results', *European Journal of Control* **2**, 6–28.

Friedrich, G., Gottlob, G. & Nejdl, W. [1990], Physical impossibilities instead of fault models, *in* 'Proceedings of the 8th AAAI Conference, Boston', pp. 331–336.

Fron, A. [1994], *Programmation par contraintes*, Addison-Wesley.

Froquet, L. [2005], Contribution à l'analyse des risques : proposition d'une méthode par scénarios et capitalisation de connaissance, PhD thesis, Automation Laboratory of Grenoble (LAG), INPG, Grenoble, France.

Gentil, S., Montmain, J. & Combastel, C. [2004], 'Combining fdi and ai approaches within causal-model-based diagnosis', *IEEE Transactions on Systems, Man and Cybernetics* **34**(5), 2207–2221.

Gertler, J. [1991], Analytical redundancy method in fault detection and isolation - survey and synthesis, *in* 'Proceeding of the IFAC Symposium on Fault Detection Supervision and Safety for Technical Process, Baden-Baden, Allemagne'.

Gertler, J. & Monajemy, R. [1993], Generating directional residuals with dynamic parity equations, *in* 'IFAC Symposium on Fault Detection Supervision and Safety for Technical Process, Sydney, Australia', pp. 507–512.

Gey, J. & Courdeau, D. [2005], *Pratiquer le management de la santé et de la sécurité au travail : Maîtriser et mettre en oeuvre l'OHSAS 18001*, ISBN 2124750836, AFNOR Editions.

Hamscher, W., Console, L. & DeKleer, J. [1992], *Readings in Model-Based Diagnosis*, Morgan Kaufmann, San Mateo.

Hayès, P. [1985], *Naïve physics I : Ontology for liquids*, J.R. Hobbs , R.C. Moore Editions, Formal Theories of sthe Common Sense World, Ablex Publishing Corporation.

Iserman, R. [1997], 'Supervision, fault detection and fault diagnosis method - an introduction', *Control Engineering Practice* **5**, 639–652.

Ishikawa, K. [1996], *La gestion de la qualité : Outils et applications pratiques*, ASIN 2100030795, Dunod.

Iwasaki, Y. & Simo, H. [1994], 'Causality and model abstraction', *Artificial Intelligence* **67**, 143–194.

Jordan, W. & Marshall, G. [1972], Failure modes, effects and criticality analysis, *in* 'Annual Reliability and Maintenability Symposium, San Francisco, California'.

Kolodner, J. [1993], *Case Based Reasoning*, Morgan Kaufmann.

Kuipers [1984], 'Commonsense reasoning about causality : Deriving behavior from structure', *Artificial Intelligence* **24**, 169–204.

Kumar, A. & Chittaro, L. [1998], 'Reasoning about function and its applications to engineering', *Artificial Intelligence in Engineerings* .

Lawley, H. [1974], 'Operating study and hazard analysis', *Chemical Engineering Progress* **4**, 45–56.

LeMoigne, J. [1990], *La modélisation des systèmes complexes*, AFCET Systèmes DUNOD.

LeMoigne, J. [1994], *La Théorie du Systčme Général, Théorie de la Modélisation*, PUF Editions.

Lesecq, S., Petropol, S. & Barraud, A. [2001], Asynchronous motor parametric faults diagnosis using wavelet analysis, *in* 'IEEE SDEMPED, Goriza, Italy'.

Leyval, L. [1991], Raisonnement causal pour la simulation de procédés industriels continus, PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France.

Lievens, C. [1976], *La sécurité des systčmes*, Cepadues.

Limnios, N. [2005], *Arbres de défaillances - 2čme édition*, number ISBN 2866012992, Lavoisier.

Meshkat, L., Dugan, J. & Andrews, J. [2002], 'Dependability analysis of systems with on-demand and active failure modes, using dynamic fault trees', *IEEE Trans. Reliability* **51**(2), 240–251.

MIL-STD1629-A, S. [1983], Procedures for performing a failure modes and effects analysis, notice 1.

Montmain, J. [2005], 'Supervision homme-machines', *Techniques de l'Ingénieur, Dossier S7620* **S2**.

Montmain, J. & Gentil, S. [2000], 'Dynamical causal model diagnostic reasoning for online technical process supervision', *Automatica* .

Mortureux, Y. [2002], 'Arbres de défaillance, des causes et d'événement', *Techniques de l'Ingénieur* **SE2**.

Nyberg, M. & Krysander, M. [2003], Combining ai, fdi, and statistical hypothesis-testing in a framework for diagnosis, *in* 'IFAC Safeprocess, Washington, USA'.

Pages, A. & Gondran, M. [1980], *Fiabilité des systčmes*, Collection de la Direction des Etudes et Recherche d'EDF, Eyrolles.

Patton, R. & Chen, J. [1997], 'Observer based fault detection and isolation : robustness and application', *Control Engineering Practice* **5**, 671–682.

Piechowiak, S. [2003], 'Intelligence articificielle et diagnostic', *Techniques de l'Ingénieur, Dossier S7217* **MT1**.

Ploix, S., Désinde, M. & Touaf, S. [2005], Automatic design of detection test in complex dynamic system, *in* '16th IFAC Worldcongress, Praha'.

Ploix, S., Touaf, S. & Flaus, J. [2003], A logical framework for isolation in fault diagnosis, *in* 'SafeProcess, Washington, USA'.

Price, C. [1999], *Computer-Based Diagnosis Systems*, Springer-Verlag, Professional Issues.

Reiter, R. [1987], 'A theory of diagnosis from first principles', *Artificial Intelligence* **32**, 57–95.

Ressencourt, H. & Travé-Massuyès, L. [2006], Hierarchical modelling and diagnosis for embedded systems, *in* 'SafeProcess, Beijing, PR China'.

Schneeweiss, W. [1999], *The fault tree method*, LiLoLe-Verlag GmbH.

Staroswiecki, M., Cassar, J. & Declerck, P. [2000], 'A structural framework for the design of fdi in large scale industrial plants', *Issues of fault diagnosis for Dynamic System, Springer Verlag* .

Struss, P. & Dressler, O. [1989], Physical negation : Integrating fault models into the general diagnostic engine, *in* 'Proceedings of the 11th IJCAI Conference, Detroit', pp. 1318–1323.

Tang, Z. & Dugan, J. [2004], Minimal cut/sequence generation for dynamic fault trees, *in* 'Annual Reliability and Maintainability Symposium 2004, Los Angeles, USA'.

Thomson, W. [1999], A review of on-line condition monitoring techniques for three-phase squirrel-cage induction motors-past, present and future, *in* 'IEEE SDEMPED'99, Gijon, Spain'.

Tixier, J., Dusserre, G., Salvi, O. & Gaston, D. [2002], 'Review of 62 risk analysis methodologies of industrial plants', *Journal of Loss Prevention in the process industries* **15**, 291–303.

Travé-Massuyès, L., Bousson, K., Evrard, J., Guerrin, F., Lucas, B., Missier, A., Tomasena, M. & Zimmer, L. [1993], 'Non-causal versus causal qualitative modelling and simulation', *Intelligent System Engineering Journal* **23**, 159–182.

Travé-Massuyès, L. & Dague, P. [2003], *Modèles et raisonnements qualitatifs*, Hermès Editions.

Veseley, W., Golberg, F., Roberts, N. & Haasl, D. [1981], *Fault Tree Handbook*, US Nuclear Regulatory Commission, Washington USA NUREG 0492.

Villemeur, A. [1988], *Sűreté de fonctionnement des systèmes industriels - Fiabilité - Facteurs humains - Informatisation*, Eyrolles Editions.

Zwingelstein, G. [1995], *Diagnostic des défaillances - Théorie et pratique pour les systčmes industriels*, Hermès Editions.

Zwinglestein, G. [1999], 'Sûreté de fonctionnement des systèmes industriels complexes', *Techniques de l'Ingénieur* **S8250**.

Résumé en français

Cette thèse propose de combiner deux types de connaissances : la connaissance du comportement d ?un système (utilisée pour l ?analyse diagnostique) et la connaissance issue de l ?analyse AMDEC (Analyse des Modes de Défaillance, de leurs Effets et leurs Criticités) du système. Pour que ces connaissances puissent être supplémentaires, un formalisme commun à ces deux connaissances est proposé. Dans la suite, les résultats de l ?analyse AMDEC, en tant que connaissance experte supplémentaire au modèle comportemental, sont intégrés lors de l'analyse diagnostique pour affiner cette analyse diagnostique. D'un autre côté, une méthode de pronostic de défaillances/défauts est proposée en intégrant les résultats de l'analyse diagnostique aux résultats de l ?analyse AMDEC. Cette thèse se conclut par une application des méthodes proposées sur un procédé exothermique industriel.

Abstract

This thesis proposes to combine two types of knowledges : the knowledge coming from the behaviour of the system (useful for diagnosis analysis) and the knowledge coming from the FMEA analysis of the system (Failure Mode and Effect Analysis). Firstly a common formalism of these two types of knowledge is proposed so that the knowledges become aditionnal. Then, the results of the FMEA analysis are integrated into the the diagnosis anlysis in order to refine it, considering the knowledge coming the FMEA analysis as an expert knowledge. Besides, a prognosis method is proposed in order to forecast future failures and future faults : this method has been developped by combining the results of the diagnosis analysis and the results of the FMEA analysis. Finally, the methods described in this thesis are applied on an industriel exothermic process.