N° Ordre...../FHC/UMBB/2011

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES

Faculté des **H**ydrocarbures et de la **C**himie

Mémoire de Magister

Présenté par

MAHDI Ismahane

Filière : Génie Electrique et Electrotechnique
Option : Infotronique

Analyse qualitative et quantitative de la sûreté de fonctionnement des convertisseurs statiques

---

Devant le jury :

BEN AZZOUZ	Djamel	Prof	UMBB	Président
BOUKRA	Abdelmadjid	MC/A	USTHB	Examinateur
MEGLOULI	Hocine	MC/A	UMBB	Examinateur
RAHMOUNE	Fayçal	MC/A	UMBB	Examinateur
NADJI	Bécharia	MC/A	UMBB	Encadreur

Année Universitaire : 2011/2012

Remerciements

Je souhaite exprimer toute ma gratitude à M^{me}. NADJI, Maître de conférences à l'Université de Boumerdès, pour sa grande disponibilité dans la direction de ce mémoire. Elle a éclairé ce travail de ses conseils judicieux, elle m'a prodigué ses encouragements tout au long de la thèse et elle a su me faire partager ses nombreuses connaissances, sa vision toujours claire et synthétique.

Je remercie sincèrement les membres de jury, à leur tête Monsieur le président, pour avoir accepté d'étudier ce modeste travail avec beaucoup d'intérêt, pour leurs remarques constructives et intéressantes et d'être là le jour de ma soutenance. C'est pour moi un grand honneur.

Je remercie chacun des membres du Laboratoire LREEI pour leur soutien et leur disponibilité.

Mon amicale reconnaissance s'adresse à tous mes camarades de Magister pour l'ambiance très sympathique et le climat d'entraide qu'ils ont su créer, et spécialement: Amine ABERKANE et Mohamed CHAOULI. Sans oublier mes collègues et chers amis de boulot surtout ceux du département informatique de l'ANSEJ : Amine, Atika, Ibrahim, Imen, Kamel, Rafika, Samir, Wahiba, Zinou et Mme Merdas, et à leurs têtes notre cher collègue et ami Mr. Fayçal DJOUDI que je lui souhaite de la réussite dans ces travaux de recherche en France. Je pense à mes amis des deux promotions 2005 et 2006 Génie Informatique de l'Université de Boumerdès ainsi ceux de l'Université des Sciences et de la Technologie Houari Boumediene.

Je voudrais rendre hommage à tous ceux qui, plus ou moins récemment, de près ou de loin, à leur manière m'ont aidé à mener à bien ce modeste travail.

A la mémoire de mon très cher grand père,

A mes chers parents,

A ma chère grande mère,

A mes chères sœurs : Nassima, Karima et Imene,

A mon adorable neveu Adam Yacine,

Et à tous ceux qui m'aiment …..

Sommaire

Sommaire

CHAPITRE II Fiabilité des Systèmes Électroniques

Liste des figures

Liste des tableaux

Résumé

L’énergie électrique occupe un rôle de plus en plus important au sein de tous les domaines. Cela occasionne une forte émergence de l’électronique de puissance, qui constitue un domaine technologique crucial pour l’obtention de systèmes performants, fiables et compétitifs. En effet, quasiment tous les systèmes utilisent des convertisseurs statiques pour traiter l’énergie électrique.

Pour cela, nous avons développé une méthodologie globale permettant de faire une analyse qualitative et quantitative de la sûreté de fonctionnement. C’est un outil d’aide à la décision pour l’évaluation des paramètres de la sûreté de fonctionnement.

Dans un premier temps, nous présentons les notions et les méthodes de la sûreté de fonctionnement. Ainsi, le système à étudier sera présenté qui est l’onduleur de tension MLI dans un actionneur électro-hydrostatique (EHA) de surface de vol. L’étude s’effectue sur les défauts qui peuvent se présenter dans une cellule de commutation, en appliquant les deux méthodes d’analyse : Analyse des Modes de Défaillances, de leurs Effets et de leurs Criticités (AMDEC) et l’Arbre des Défaillances (AdD).

Pour une meilleure analyse de sûreté de fonctionnement, nous avons introduit une approche de modélisation basée sur les arbres de décision binaires (BDD), et ceci, en démarrant de l’AMDEC à l’AdD et enfin nous terminons par la génération de l’arbre binaire réduit.

Mots clefs— sûreté de fonctionnement, fiabilité, modélisation, algèbre de Boole, arbre binaire.

Abstract

Electric power is of increasing importance in all domains. As a consequence, static converters have become widespread, and power electronics is crucial for improving performance, reliability and competitiveness. Indeed, almost all the systems use static inverters to treat electrical energy.

For this, we have developed a global methodology allowing to model and to evaluate the dependability. It is a tool of decision-making aid by developing the qualitative and quantitative evaluation of the parameters of the dependability.
Initially, we have presented the concepts and the methods of dependability. Thus, the system studied will be presented which is the inverter of tension MLI in an electro-hydrostatic actuator (EHA) of surface of flight. The study is carried out on the defects which can arise in a cell of commutation, by applying two methods of analysis: Failure Modes, Effects and Criticality Analysis (FMECA) and the Fault Tree (FTA).
For better analysis of dependability, we have introduced an approach of modelling based on the binary decision diagrams (BDD), and this while starting of the FMECA to Fault Tree and we finish by the generation of the reduced binary tree.

Key words—FMECA, dependability, modeling, Boolean algebra, binary tree.

ملخّـــص

تحتل الطاقة الكهربائية دورا مهما أكثر فأكثر في جميع مجالات الحياة. هذا ما يسمح بظهورشديد لإلكترونيك الطاقة , و التي تشكل مجال تكنولوجي أساسي لإنتاج أنظمة فعالة , موثوقة و تنافسية . في الواقع , كل النظم تستخدم محولات ثابتة لمعالجة الطاقة الكهربائية .

لهذا قمنا بتطوير منهجية شاملة تسمح بالتحليل النوعي و الكمي للإعتمادية . بل هو أداة لدعم اتخاذ القرار من خلال تقييم معـــايير الإعتمادية .

في بادئ الأمر , نقدم مفاهيم و مناهج الإعتمادية . و بالتالي , النظام قيد الدراسة يتم تقديمه و الذي هو محوَل في المحرك الكهربائي الهيدروليكي (EHA) لسطح محرّك طائرة. تتمَ الدَراسة على الإختلالات الّتي قد تطرأ على خليَة التَحويل, و ذلك باستخدام طريقتين للتَحليل : تحليل طرق الإخفاق, آثارها و أخطارها (AMDEC) و شجرة الإخفاقات (AdD).

و لأجل تحليل أفضل للإعتماديَة, أدرجنا منهجية لإعداد نماذج مبنية على أشجار القرار الثّنائيَة (BDD), و هذا, انطلاقًا من AMDEC إلى AdD. و أخيرًا, نختمها بإنشاء شجرة ثنائيَة مخفَضة.

كلمـــات مفتاحـيّة ــ الإعتماديّة , الموثوقيّة, إعداد نماذج , جبر المنطق , أشجار ثنائية .

Introduction Générale

Introduction Générale

L'énergie électrique occupe un rôle de plus en plus important au sein de tous les domaines. Cela occasionne une forte émergence de l'électronique de puissance, qui constitue un domaine technologique crucial pour l'obtention de systèmes performants, fiables et compétitifs. En effet, quasiment tous les systèmes utilisent des convertisseurs statiques pour traiter l'énergie électrique. Cela va de l'alimentation de quelques watts des téléphones portables à des puissances de plusieurs dizaines de mégawatts sur les réseaux de transport. Or, compte tenu du rôle majeur attribué aujourd'hui aux convertisseurs statiques en tant qu'organe d'alimentation, de traitement et d'adaptation, il est clair qu'ils conditionnent largement la sûreté de fonctionnement des systèmes auxquels ils sont raccordés.

Afin d'obtenir des systèmes électroniques dans lesquels les utilisateurs placent une grande confiance, des études de sûreté de fonctionnement doivent être menées. Or, la quasi totalité des convertisseurs sur le marché, ne tolèrent pas le défaut le plus commun : celui du court-circuit d'un semi-conducteur. Pour répondre à des critères économiques, l'interrupteur fonctionne à la limite de ses possibilités. Il est ainsi soumis à des contraintes souvent très sévères et bien que le semi-conducteur soit robuste, le composant peut se mettre en défaut dans la mesure où 1% de l'énergie transférée peu suffire à le détruire.

Notre travail s'inscrit dans le cadre de la sûreté de fonctionnement des onduleurs de moyenne puissance, particulièrement l'onduleur classique à bras de secours. Sa topologie peut tolérer une défaillance interne. Pour cela, nous avons développé une méthodologie globale, qui est un outil d'aide à la décision, en valorisant l'évaluation qualitative et quantitative des paramètres de la sûreté de fonctionnement. Et ceci, en partant de l'analyse des modes de défaillances, de leurs effets et de leurs criticités (AMDEC) et en arrivant à l'arbre de décision binaire (BDD).

Ce mémoire comporte cinq chapitres.

Dans le premier chapitre, nous présentons les différentes méthodes de sûreté de fonctionnement. Nous rappellerons dans un premier temps quelques concepts de la sûreté de fonctionnement et les méthodes d'analyse les plus utilisées.

Le deuxième chapitre décrit les principales lois associées à la fiabilité ainsi que les mécanismes de défaillance des composants électroniques.

Dans le troisième chapitre les trois méthodes d'analyse de la sûreté de fonctionnement : l'AMDEC, l'analyse par l'Arbre de Défaillance et l'analyse en utilisant les Diagrammes de Décision Binaires (BDDs) seront présentées en détail. Elles seront, par la suite, appliquées dans l'étude de la sûreté de fonctionnement de notre système dans le quatrième chapitre.

Dans le cinquième chapitre nous citons les fonctions employées ainsi que la structure de données adoptée, pour la conception et la réalisation de la plateforme informatique permettant une analyse qualitative et quantitative de la sûreté de fonctionnement. Les résultats obtenus seront discutés par la suite. Nous terminerons avec une conclusion générale.

Chapitre I
Notions et concepts de la Sûreté de Fonctionnement

I.1 Introduction

Avant de décrire les différentes méthodes d'analyse de la Sûreté de Fonctionnement (SdF), il est essentiel d'introduire des notions et des concepts s'appuyant sur des normes internationales. Il est donc fondamental d'avoir des définitions précises pour le diagnostic de défaillances, de pannes et de défauts. Par ailleurs, les différentes instances de normalisation (CEI, ISO, AFNOR et CEN) ont entrepris de réviser la terminologie utilisée dans les domaines de la maintenance et de la sûreté de fonctionnement. Dans un contexte de fortes exigences pour des systèmes de plus en plus fiables et sûrs, et de durées de garanties croissantes, il est impératif de vérifier le plus tôt possible que les performances des systèmes sont conformes au cahier des charges.

Dans les différents secteurs de l'industrie, de nombreuses méthodes d'analyse de la sûreté de fonctionnement ont été développées. La maîtrise de la sûreté de fonctionnement d'un système donné nécessite de prendre en compte simultanément les différentes technologies et méthodes d'analyse.

I.2 Bref historique

Au début de l'ère industrielle aux années 1930, l'apparition de l'électricité puis son développement conduisirent les ingénieurs à chercher à rendre fiable cette source de puissance. Avec le développement du transport aérien, on a vu apparaître un autre aspect de la fiabilité : le recueil d'informations statistiques sur la fréquence de pannes des divers équipements des avions. [I.Bazovsky, 1961]

Durant les années 1940, les premiers modèles de fiabilité prévisionnelle apparaissent en Allemagne où se développaient les projets de missile V1 sous la direction de Von Braun. [I.Bazovsky, 1961], [G.Galabrese, 1947]

Le manque de fiabilité était souvent devenu le cauchemar des ingénieurs, c'est ainsi que naquit la **« loi de Murphy »** en **1949** : « *Si un ennui a la moindre chance de se produire, dites-vous qu'il se produira !* » [A.Bloch, 1982]

La commission **AGREE** (Advisory Group on Reliability of Electronic Equipement), créée en **1952** par le Department of Defense et l'ensemble des industries électroniques [P.DT O'CONNOR, 1981], insista sur le besoin pour les nouveaux équipements d'être testés pendant

plusieurs milliers d'heures dans des environnements très contraignants (incluant des hautes et basses températures, des vibrations, …) afin de découvrir le maximum de points faibles suffisamment tôt pour permettre de corriger ces défauts avant le lancement de la production en série. La commission recommanda également, d'une part, le calcul du **MTBF** (durée moyenne entre deux défaillances consécutives d'une entité réparée). [MIL-STD 781]

Des évolutions comparables sont observées dans les pays européens ; c'est ainsi qu'en France, en **1955**, le **CENT** (Centre National d'Etudes des Télécommunications) commençait ses premiers travaux sur la fiabilité. [M.Schwob, G.Peyrache, 1969]

C'est au début des années **1960** que fut créée la méthode de l'**AMDE** (Analyse des Modes de Défaillance et de leurs Effets) dans le domaine de l'aéronautique.

Les méthodes d'analyse prévisionnelle de la fiabilité et de la sécurité des systèmes se généralisèrent dans l'industrie aéronautique. Il faut aussi mentionner l'existence des premières **banques de données de fiabilité** qui commencent à être disponibles à la fin de cette décennie.

Cette période voit aussi la naissance de la revue « **IEEE Transactions on Reliability** » créée par **IEEE** (Institute of Electrical and Electronic Engineers).

De nombreuses et nouvelles méthodes furent développées dont la méthode de **l'Arbre des Conséquences** pendant la première moitié des années 1980.[S.Levine, F.Stetson, 1984]

Ainsi, il apparait au milieu des années **1980** que la prise en compte des techniques de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité tend à se généraliser, d'une part, pour la maîtrise et la gestion des grands risques industriels et, d'autre part, pour la conception de matériels de grande consommation.

I.3 Sûreté de fonctionnement des systèmes

Dans un contexte international extrêmement concurrentiel, les entreprises doivent maîtriser les différents outils qui leur permettent de rester compétitives et doivent s'engager dans des actions d'amélioration à tous les niveaux. La complexité croissante des systèmes, la réduction de leurs coûts de conception et d'exploitation, leur utilisation de plus en plus importante dans la vie quotidienne font que la sûreté de fonctionnement est devenue incontournable dans le développement de tout système industriel.

La sûreté de fonctionnement (SdF) fait partie des enjeux majeurs de ces dernières années et des années à venir. Cette notion désigne à la fois un ensemble de moyens et un ensemble de résultats produits par ces moyens :

– des méthodes et des outils pour caractériser et maîtriser les effets des aléas, des pannes et des erreurs [Pagès et Gondran, 1980], [O'Connor, 2002] ;

– la quantification des caractéristiques des systèmes pour exprimer la conformité dans le temps de leurs comportements et de leurs actions [Pagès et Gondran, 1980].

I.3.1 Eléments de la sûreté de fonctionnement

La démarche et le raisonnement de la sûreté de fonctionnement s'appuient sur des grandeurs qui seront précisées dans cette partie. On peut trouver plus de détails dans les ouvrages suivants : [Villemeur, 1988], [Pagès et Gondran, 1980], [Procaccia et Morilhat, 1996], [Ayyub et McCuen, 1997], [Zwingelstein, 1996], [O'Connor, 2002].

Différents auteurs définissent la SdF comme :

– la fiabilité, la disponibilité, la maintenabilité et la sécurité ;

– la science des défaillances [Villemeur, 1988] ;

– la confiance justifiée dans le service délivré [Laprie et al., 1995] ;

– le maintien de la qualité dans le temps.

La définition « Fiabilité, Disponibilité, Maintenabilité et Sécurité » qu'on retrouve dans l'acronyme FDMS (RAMS en anglais), fait référence aux définitions de ces termes et met en avant leur complémentarité. Si la fiabilité, la maintenabilité, la disponibilité ou la sécurité sont aussi des performances d'un système, la SdF ne se réduit pas uniquement à une de ces performances, elle se construit à travers toutes ces performances.

La définition « science des défaillances » suppose la connaissance, l'évaluation, la prévision, la mesure et la maîtrise des défaillances. Ainsi la sûreté de fonctionnement apparaît davantage comme l'aptitude d'une entité à satisfaire une ou plusieurs fonctions requises dans des conditions données [Villemeur, 1988].

La définition « confiance justifiée dans le service délivré » dépend principalement de la perception des utilisateurs. Le service délivré par un système est son comportement perçu par son, ou ses utilisateurs, sachant qu'un utilisateur est un autre système (humain ou physique) qui interagit avec le système considéré [Laprie et al., 1995].

La définition « maintien de la qualité dans le temps » prend en compte la conformité aux exigences (explicites ou non). Elle présente le défaut de laisser supposer qu'une activité SdF se conduit nécessairement dans le cadre d'une démarche qualité, ce qui est insuffisant.
La définition de la SdF sera considérée globalement comme la conjugaison de ces quatre définitions. L'ensemble de ces définitions est cohérent et fournit une image plus complète de la SdF prise selon plusieurs points de vue.

I.3.1.1 Fiabilité

La fiabilité (Reliability en anglais) est l'aptitude d'une entité à accomplir les fonctions requises dans des conditions données pendant une durée donnée. Elle est caractérisée par la probabilité R(t) que l'entité E accomplisse ces fonctions, dans les conditions données pendant l'intervalle de temps [0, t], sachant que l'entité n'est pas en panne à l'instant 0.

$$R(t) = P\,[E \text{ non défaillante sur } [0, t]] \qquad \textbf{(I.1)}$$

I.3.1.2 Disponibilité

La disponibilité (Availability en anglais) est l'aptitude d'une entité à être en état d'accomplir les fonctions requises dans les conditions données et à un instant donné. Elle est caractérisée par la probabilité A(t) que l'entité E soit en état, à l'instant t, d'accomplir les fonctions requises dans des conditions données.

$$A(t) = P\,[E \text{ non défaillante à l'instant } t] \qquad \textbf{(I.2)}$$

I.3.1.3 Maintenabilité

La maintenabilité (Maintainability en anglais) est l'aptitude d'une entité à être maintenue ou rétablie dans un état dans lequel elle peut accomplir une fonction requise, lorsque la maintenance est réalisée dans des conditions données avec des procédures et des moyens prescrits. Elle est caractérisée par la probabilité M (t) que l'entité E soit en état, à l'instant t, d'accomplir ses fonctions, sachant que l'entité était en panne à l'instant 0.

$$M(t) = P\,[E \text{ est réparée sur } [0, t]] \qquad \textbf{(I.3)}$$

I.3.1.4 Sécurité

La sécurité (Safety en anglais) est l'aptitude d'une entité à éviter de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques. Elle est caractérisée par la probabilité S (t) que l'entité E ne laisse pas apparaître dans des conditions données, des événements critiques ou catastrophiques.

S (t) = P [E évite des événements critiques ou catastrophiques sur [0, t]] **(I.4)**

Il est à noter que dans le domaine de l'informatique la sécurité a souvent deux facettes : la sécurité-innocuité (Safety en anglais) qui vise à se protéger des défaillances catastrophiques et la sécurité-confidentialité (Security en anglais) qui correspond à la prévention d'accès ou de manipulations non autorisées de l'information et concerne la lutte contre les fautes intentionnelles.

I.3.2 Les principales lois

La fiabilité est, de plus en plus fréquemment, une grandeur quantitative et nécessite la connaissance des distributions de durée de vie afin de l'estimer.

Nous présentons dans cette section les lois et les modèles de fiabilité susceptibles, selon l'expérience, de représenter des distributions de durée de vie qui interviennent le plus fréquemment dans l'analyse de la sûreté de fonctionnement. Nous rappelons les principales propriétés de ces lois, les fonctions associées, les densités de probabilité ainsi que les taux de défaillance [Marcovici et Ligeron, 1974].

I.3.2.1 Loi exponentielle

Cette loi décrit la vie des matériels qui subissent des défaillances brutales. La loi exponentielle est la plus couramment utilisée en ***fiabilité électronique*** pour décrire la période durant laquelle le taux de défaillance des équipements est considéré comme constant (défaillance aléatoire).

Elle est définie par un seul paramètre, le taux de défaillance, **λ**.

Elle est caractérisée par :

– la fiabilité

$$R(t) = e^{-\lambda t} \quad \textbf{(I.5)}$$

– la densité de probabilité

$$f(t) = \lambda e^{-\lambda t} \quad \textbf{(I.6)}$$

– le taux de défaillance

$$\lambda(t) = \lambda \quad \textbf{(I.7)}$$

I.3.2.2 Loi de Weibull

La loi de Weibull, est souvent utilisée en ***mécanique*** ; elle caractérise bien le comportement du produit dans les trois phases de vie selon la valeur du paramètre de forme **β** : période de

jeunesse (β < 1), période de vie utile (β = 1) et période d'usure ou vieillissement (β > 1). La loi de Weibull est définie par deux paramètres **η** et **β**.

Elle est caractérisée par :

– la fiabilité

$$R(t) = e^{-\left(\frac{t}{\eta}\right)^{\beta}} \quad \textbf{(I.8)}$$

– la densité de probabilité

$$f(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{-\left(\frac{t}{\eta}\right)^{\beta}} \quad \textbf{(I.9)}$$

– le taux de défaillance

$$\lambda(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} \quad \textbf{(I.10)}$$

I.3.2.3 Loi normale

La loi normale elle s'applique à de nombreux phénomènes. La loi normale est définie par la moyenne **μ** et l'écart type **σ** :

– la fonction de répartition

$$F(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2} dx \quad \textbf{(I.11)}$$

– la densité de probabilité

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{(t-\mu)}{\sigma}\right)^2} \quad \textbf{(I.12)}$$

I.3.2.4 Loi lognormale

Une variable aléatoire continue et positive t est distribuée selon une loi lognormale si son logarithme est distribué suivant une loi normale. Cette distribution est utilisée en fiabilité pour modéliser ***les défaillances par fatigue***. La loi lognormale a deux paramètres **μ** et **σ** :

– la fiabilité

$$R(t) = 1 - \Phi\left(\frac{log(t) - \mu}{\sigma}\right) \quad \textbf{(I.13)}$$

– la densité de probabilité

$$f(t) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{(log(t)-\mu)}{\sigma}\right)^2} \quad \textbf{(I.14)}$$

– le taux de défaillance

$$\lambda(t) = \frac{e^{-\frac{1}{2}\left(\frac{(log(t)-\mu)}{\sigma}\right)^2}}{t\int_0^{\infty} \sigma\sqrt{2\pi} f(t)dt} \quad \textbf{(I.15)}$$

I.3.2.5 Loi Gamma

Elle représente la loi de probabilité d'occurrence de ***a*** événements dans un processus poissonien. Par exemple si t_i est le temps entre les défaillances successives d'un système, et que t_i suive une distribution exponentielle, le temps cumulé d'apparition de *a* défaillances suit une loi Gamma :

– la densité de probabilité

$$f(t) = \frac{t^{a-1} e^{-\frac{t}{b}}}{b^a \Gamma(a)} \quad \textbf{(I.16)}$$

– le taux de défaillance

$$\lambda(t) = \frac{t^{a-1} e^{-\frac{t}{b}}}{b^a \int_t^{\infty} \Gamma(a) f(u)\, du} \quad \textbf{(I.17)}$$

I.3.2.6 Loi Bêta

Cette loi représente la probabilité pour qu'un matériel survive jusqu'à un instant t, quand on essaie n matériels. La loi Bêta a deux paramètres ***a*** et ***b*** :

– la densité de probabilité

$$f(t) = \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} t^{a-1} \cdot (1-t)^{b-1} \quad \textbf{(I.18)}$$

I.3.2.7 Loi uniforme

La loi uniforme est souvent utilisée en fiabilité pour ***les essais bayésiens*** en l'absence de connaissances pour construire l'information a priori.

– la fonction de répartition

$$F(t) = \frac{t-a}{b-a} \quad \textbf{(I.19)}$$

– la densité de probabilité

$$f(t) = \frac{1}{b-a} \quad \textbf{(I.20)}$$

I.3.2.8 Autres lois

Il existe d'autres lois de probabilité comme : la ***loi binômiale***, la ***loi de Poisson***, la ***loi hypergéométrique***,...

I.3.3 Méthodes d'analyse de la fiabilité des systèmes

Dans le processus de développement de systèmes complexes, la sûreté de fonctionnement est devenue une caractéristique essentielle. Ainsi, afin d'optimiser le développement de ces systèmes, il est impératif de disposer de méthodes permettant d'évaluer les paramètres de la sûreté de fonctionnement.

Les méthodes d'analyse de la sûreté de fonctionnement d'un système complexe sont nombreuses et on peut trouver plus de détails dans les ouvrages suivants : [Villemeur, 1988], [Pagès et Gondran, 1980], [O'Connor, 2002].

Dans le tableau I.1, nous avons caractérisé chaque démarche ou méthode étudiée selon trois critères :

– méthode inductive ou déductive ;

– méthode quantitative ou qualitative ;

– les objectifs recherchés.

Démarches/Méthodes	Inductive/Déductive	Quantitative/Qualitative	Objectifs visés
Analyse préliminaire de risques (APR)	Inductive	Qualitative	Repérer à priori les risques à étudier
Analyse des modes de défaillance de leurs effets et de leurs criticités (AMDEC)	Inductive	Qualitative	Evaluer les conséquences des défaillances
Arbre de défaillances (AdD)	Déductive	Quantitative	Evaluer les scénarios d'un événement redouté
Diagramme de fiabilité (DF)	Déductive	Quantitative	Représenter un modèle du système à partir de la fiabilité des composants
Méthode de l'Espace des Etats (MEE)	Inductive	Quantitative	Repérer le passage par les états de défaillance sur le fonctionnement du système
Réseaux de Pétri (RdP)	Inductive	Quantitative	Repérer le passage par les états de défaillance sur le fonctionnement du système
Arbre d'événement (AE)	Inductive	Quantitative	Evaluer les conséquences possibles d'un événement
Méthode de Combinaisons de Pannes Résumées (MCPR)	Inductive	Qualitative	Déterminer les combinaisons de défaillances
Méthode de Diagramme Causes-Conséquences (MDCC)	Inductive/Déductive	Quantitative	Analyse d'un événement initiateur
Table de Vérité (TV)	Inductive	Quantitative	Recenser toutes les combinaisons d'états

Tab. I.1: Principales démarches et méthodes de fiabilité.

Les méthodes qui partent des causes pour en déduire les conséquences sont appelées inductives et celles qui partent des conséquences pour en déduire les causes sont appelées déductives.

Nous avons qualifié de quantitatives les méthodes qui offraient une possibilité importante de quantification et de qualitatives les méthodes qui l'excluaient, ou dans lesquelles cet aspect est marginal.

Les méthodes les plus utilisées, APR, AMDEC, AdD, DF, MEE et RdP, sont présentées dans les paragraphes suivants. Dans l'annexe A, nous retrouvons la description des autres méthodes : AE, MCPR, MDCC et TV.

I.3.3.1 Analyse Préliminaire des Risques (APR)

L'Analyse Préliminaire des Risques (APR) est une méthode générale couramment utilisée pour l'identification d'une part des risques d'un système complexe et d'autre part pour l'évaluation de la gravité des conséquences liées aux risques [Villemeur, 1988], [Zwingelstein, 1996]. Il est recommandé de commencer l'APR dès les premières phases de la conception en utilisant toutes les données alors disponibles et de continuer à compléter cette analyse jusqu'à la fin de vie du système [Villemeur, 1988].

L'APR a pour objectifs :

– l'identification des dangers et de leurs causes (entités et situations dangereuses, accidents potentiels,...) ;

– l'évaluation et l'acceptation des risques permettant une hiérarchisation ;

– la proposition de mesures propres à réduire et à contenir les risques à des niveaux acceptables.

Cette démarche APR, ou la variante analyse préliminaire des dangers (APD), est généralement une première étape indispensable lorsque des questions de sécurité sont posées. Elle l'est beaucoup moins s'il n'est question que de la fiabilité, maintenabilité ou disponibilité. Quand elle est réalisée dès le début du projet, dès la première phase de développement du système, elle sert de référence tout au long du projet.

Le principal avantage de l'Analyse Préliminaire des Risques est de permettre un examen relativement rapide des situations dangereuses dans des systèmes complexes. Par rapport aux autres méthodes présentées ci-après, elle apparaît comme relativement économique en termes de temps passé et ne nécessite pas un niveau très détaillé de description du système étudié.

En revanche, l'APR ne permet pas de caractériser l'enchaînement des évènements susceptibles de conduire à un accident majeur pour des systèmes complexes. Elle permet d'identifier des points critiques devant faire l'objet d'études plus détaillées.

La méthode APR est basée sur la liste des éléments qui peuvent se conjuguer pour provoquer un accident : entités dangereuses, situations dangereuses, accidents potentiels,... Cette liste est établie par des experts.

I.3.3.2 Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités (AMDEC)

L'Analyse des Modes de défaillance et de leurs Effets (AMDE) est une méthode qui nécessite la connaissance de tous les modes de défaillance du système [Villemeur, 1988]. En effet, l'efficacité de l'AMDE repose sur la représentativité de ces modes de défaillance et l'exhaustivité de cette liste.

L'AMDE est une méthode qui permet :

– d'évaluer les effets de chaque mode de défaillance des composants d'un système sur les diverses fonctions de ce système ;

– de déterminer l'importance de chaque mode de défaillance sur le fonctionnement normal du système et d'en évaluer l'impact sur la fiabilité du système considéré ;

– de hiérarchiser les modes de défaillances connus suivant la facilité de détection et de traitement.

L'AMDE est établie à partir des définitions suivantes :

– une défaillance est la cessation de l'aptitude d'un élément ou d'un système à accomplir une fonction requise ;

– un mode de défaillance est l'effet par lequel une défaillance est observée sur un élément du système ;

– une cause de défaillance est constituée par les évènements qui conduisent aux défaillances ;

– l'effet d'un mode de défaillance regroupe les conséquences associées à la perte de l'aptitude d'un élément à remplir une fonction requise ;

– les moyens de détection représentent les actions permettant de mettre en évidence le mode de défaillance (les tests périodiques, les inspections, les alarmes,...).

Pour évaluer la criticité d'une défaillance, l'Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités (AMDEC) [Villemeur, 1988] apparaît comme une suite logique à l'AMDE. La criticité évalue, pour chaque mode de défaillance, le triplet probabilité d'occurrence - gravité - risque de la non-détection. La gravité associe, pour chaque mode de défaillance, une classe des effets : mineurs, significatifs, critiques ou catastrophiques. Le

risque de non-détection est d'autant plus grand qu'aucune barrière matérielle ou logicielle n'est prévue pour éviter la défaillance. La criticité d'une défaillance est égale au produit de la probabilité d'occurrence, de la gravité et du risque de non-détection. L'AMDEC reprend en effet les principales étapes de l'AMDE et ajoute une évaluation quantitative de la criticité. Dans les projets où il est difficile d'évaluer la criticité, car les choix technologiques ne sont pas définis, on utilise uniquement l'AMDE.

Il est à noter qu'une version dérivée de l'AMDEC, adaptée uniquement au développement de logiciels, est la méthode de l'Analyse des Effets des Erreurs de Logiciel (AEEL).

L'objectif principal de l'AEEL est d'identifier les défaillances potentielles et les erreurs de conception et de programmation afin d'analyser leurs effets internes et externes.

Il existe plusieurs types d'AMDEC : AMDEC produit, AMDEC processus, AMDEC projet, AMDEC service ou administrative, AMDEC sécurité, ...

Le principal avantage de l'AMDEC est son aptitude à détecter les défaillances des éléments conduisant à la défaillance globale du système. Elle constitue un outil précieux pour l'identification de défaillances potentielles et les moyens d'en limiter les effets ou d'en prévenir l'occurrence.

Par contre, dans le cas de systèmes particulièrement complexes comptant un grand nombre de composants et d'interactions, l'AMDEC est très difficile à maîtriser, compte tenu du volume important d'informations à traiter et l'impossibilité de décrire des défaillances multiples.

Bien entendu, cette analyse cause/conséquence n'a d'intérêt que si l'on en tire des suites. Aussi, les divers documents sur l'AMDEC comprennent des prolongements relatifs à l'exploitation du système.

I.3.3.3 Arbre de Défaillance (AdD)

L'analyse par un Arbre de Défaillances (AdD) est une méthode qui permet, à partir d'un événement redouté, recensé à l'aide d'une APR, de déterminer les enchaînements d'évènements ou combinaisons d'évènements pouvant conduire à cet événement redouté [Pagès et Gondran, 1980], [Villemeur, 1988], [Cocozza-Thivent, 1997], [Birolini, 1997]. Cette analyse permet de descendre de cause en cause jusqu'aux évènements de base susceptibles d'être à l'origine de l'événement redouté. Cette méthode d'analyse sera détaillée dans chapitre **III**.

I.3.3.4 Diagramme de Fiabilité (DF)

La Méthode du Diagramme de Fiabilité (DF) est utilisée pour analyser et calculer la fiabilité des systèmes [Pagès et Gondran, 1980], [Cocozza-Thivent, 1997], [Villemeur, 1988], [Birolini, 1997]. Elle est aussi dénommée Méthode du Diagramme de Succès. L'analyse par DF a pour but de représenter l'architecture du système.

Pour cette modélisation, des blocs représentent généralement des composants, des sous-systèmes ou des fonctions. La modélisation consiste à rechercher les liens entre ces blocs.

Un bloc est considéré comme un interrupteur fermé lorsque l'entité est en état de fonctionnement ou un interrupteur ouvert lorsque l'entité est en état de panne. Le système est déclaré en état de fonctionnement si le "signal" qui entre dans le diagramme est récupéré à la sortie, sinon le système est en panne.

D'une part, le DF permet une analyse qualitative en déterminant les chemins qui conduisent à la réussite de la mission du système et la recherche des composants apparaissant dans le plus grand nombre de ces chemins et les scénarios qui conduisent à l'échec de la mission, pour éviter les incidents.

D'autre part, le DF permet une analyse quantitative qui a pour objectif en particulier de définir la probabilité de bon fonctionnement du système. Les calculs reposent sur les probabilités de réussite des missions des constituants du système.

La méthode a ses limites d'application : il faut s'assurer de l'indépendance entre les blocs et ne permet pas de modéliser des systèmes dynamiques.

I.3.3.5 Méthode de l'Espace des Etats (MEE)

La Méthode de l'Espace des Etats (MEE) (ou processus de Markov) permet l'analyse de fiabilité des systèmes réparables [Villemeur, 1988], [Pagès et Gondran, 1980], [Birolini, 1997].

Supposons un système constitué par des composants, chaque composant ayant un nombre fini d'états de fonctionnement et de panne ; on construira un graphe dont les sommets correspondront aux différents états du système et les arcs correspondront aux transitions (panne et réparation) entre états. Pour un système à n composants, si chaque composant a deux états (fonctionnement et panne), le nombre maximum d'états est 2^n.

Le calcul de la fiabilité est réalisé à partir de différents états du système.

Le principal avantage de cette méthode est de permettre la modélisation de systèmes réparables.

La méthode est souvent difficile à appliquer pour des systèmes de grande taille caractérisés par une explosion des états.

I.3.3.6 Réseaux de Pétri (RdP)

Les réseaux de Petri (RdP) [Petri, 1962], ont connu depuis leur invention un réel succès en raison de leur simplicité mathématique, des avantages de la représentation graphique et de leur compacité. C'est l'un des outils les plus populaires pour la modélisation de systèmes à événements discrets et les domaines d'applications sont très vastes. [Dutuit et al., 1997], [Ladet, 1989], [Valette, 2002].

Rappel sur les Réseaux de Pétri

Un *RdP* marqué est un 5-uplet $R=<P, T, Pré, Post, M_0>$ avec :

P : un ensemble fini de places ;

T : un ensemble fini de transitions, tel que $P \cap T = \{\}$;

Pré : $P \times T \rightarrow$ N l'application *places précédentes* qui indique le nombre de jetons prélevés par les transitions dans les places en amont ;

Post : $T \times P \rightarrow$ N l'application *places suivantes* qui indique le nombre de jetons déposés par les transitions dans les places en aval ;

M_0 : le marquage initial.

Un *RdP* peut être vu comme un graphe avec deux types de sommets : des places et des transitions. Les places sont marquées par des jetons et le réseau évolue par le franchissement des transitions selon les applications *Pré* et *Post*. On utilise également la notation *C=Post-Pré* où *C* est la matrice d'incidence.

Les Réseaux de Pétri t-temporels

Un réseau de Petri t-temporel est une paire $N_{tl}=<R, D>$ où R est un réseau de Petri $<P, T, Pré, Post, M_0>$ marqué et D est une fonction qui à chaque transition t_i du réseau fait correspondre un intervalle statique de temps $d(t_i) = [d_{imin}(t_i), d_{imax}(t_i)]$ qui décrit une durée de sensibilisation.

L'intervalle de temps est associé aux transitions. La transition t_i doit rester sensibilisée durant au moins d_{imin} unités de temps et au plus d_{imax} unités de temps avant d'être franchie. Par contre les jetons peuvent à tout moment être consommés par une autre transition.

Les Réseaux de Pétri de haut niveau

Les réseaux de Pétri ont été utilisés avec succès dans un certain nombre d'applications réelles dans divers domaines. Cependant, deux inconvénients principaux ont limité leur utilisation pour des applications de grande complexité :

- le premier est que le réseau de Pétri n'est pas adapté pour la manipulation des données. Même pour des problèmes simples, la structure du réseau devient trop complexe ;
- le deuxième inconvénient est qu'il n'y a pas de hiérarchie dans le réseau de Pétri et qu'il n'est donc pas possible de construire le modèle d'un système complexe comme une composition de sous modèles.

Pour résoudre ce problème de la manipulation de données, beaucoup de travaux ont proposé l'extension des formalismes réseau de Pétri. Ces extensions sont appelées réseaux de Pétri de haut niveau. On peut citer, les réseaux de *Pétri Colorés* et les réseaux de *Pétri prédicats transitions*.

Dans les réseaux de *Pétri Colorés*, le pouvoir de description est augmenté en associant des couleurs aux jetons, aux places et aux transitions. Chaque jeton a une couleur qui lui permet d'être distingué des jetons portant d'autres couleurs. Un ensemble de couleurs est associé à chaque place et détermine les couleurs des jetons qui peuvent être déposés dans la place. Pour chaque transition est associé un ensemble de couleurs qui représente les différentes manières de tirer la transition.

Les réseaux de *Pétri Prédicats Transitions* (*RdP PT*), introduisent le concept de variables. Chaque transition possède une fonction de sensibilisation spécifiée comme une formule logique avec des variables. Les transitions sont des règles d'un système logique de premier ordre, qui est un système logique avec des variables.

Un réseau de *Pétri prédicats transitions* est un 3-tuple $RdP\ PT = <R, A, M_0>$, où :

R est la structure de réseau de Pétri Définie par le 4-tuple $< P, T, Pré, Post>$;

A est l'annotation du *RdP PT*, définie par le 4-tuple $A=<X, A_x, A_c, A_a>$, où :

X est un ensemble de variables ;

A_x est une application associant à chaque arc un vecteur de variable X ;

A_c est une application associant à chaque transition une condition sous la forme d'un prédicat utilisant les variables ;

A_a est une application associant à chaque transition une action sous la forme d'une suite d'affectation de valeurs aux variables ;

M_0 est le marquage initial du réseau de Pétri. Chaque jeton est un vecteur de variables. Le marquage initial définit les valeurs des variables du jeton.

Dans un réseau de *Pétri Prédicats Transitions*, une transition est sensibilisée (ou non) pour un ensemble défini de jetons de ses places d'entrée. Si les variables associées à l'arc sont remplacées par les valeurs des variables des jetons et la condition de sensibilisation est vraie, alors la transition est sensibilisée. Une fois sensibilisée, la transition peut être tirée. L'action associée à une transition définit les valeurs des variables des arcs sortants. Ce sont les mêmes valeurs des jetons générés dans les places de sortie par le tir de la transition.

Les Réseaux de Pétri Prédicats Transitions Différentiels

L'utilisation des variables dans les réseaux de *Pétri Prédicats Transitions* a motivé son application pour la modélisation des systèmes hybrides, ce qui a donné naissance aux réseaux de *Pétri Prédicats Transitions Différentiels* (*RdP PTD*). Dans les réseaux de *Pétri Prédicats Transitions*, les variables associées aux jetons ne peuvent être modifiées que par un franchissement de transition. En d'autres termes, ces variables ne peuvent avoir une évolution continue. L'idée des réseaux de *Pétri Prédicats Transitions Différentiels* est que chaque marquage modélise une configuration du système. Un système d'équations différentielles est associé à chaque place. Ce système activé par l'arrivée de jetons, décrit l'évolution continue dans le temps des variables associées aux jetons présents dans la place.

Les réseaux de *Pétri Prédicat Transitions Différentiels* possèdent des fonctions de sensibilisation associées aux transitions. Elles définissent des seuils sur les variables continues mises en jeu dans les équations associées aux places d'entrées. Lorsque la variable atteint le seuil défini pour la fonction de sensibilisation, la transition sensibilisée à laquelle est associée cette fonction est tirée. Un autre élément des réseaux de *Pétri Prédicats Transitions Différentiels* est la fonction de jonction. Les fonctions de jonction sont utilisées pour introduire des discontinuités sur les variables continues et sont similaires aux actions dans le cas des réseaux de *Pétri Prédicats Transitions*.

Un réseau de *Pétri Prédicats Transitions Différentiel* est un 3-uplet, $RdP\ PTD = < R, A, M_0 >$, où :

R est la structure du réseau de Pétri définie par le 4-tuple $< P, T, Pré, Post >$;

A est l'annotation du *RdP PTD*, définie par le 4-tuple $A=<X, A_p, A_f, A_e, A_j >$, où :

X est l'ensemble des variables de A ;

A_p est une application qui associe à la place P_i un vecteur de variable X_{pi} de l'ensemble des variables X, avec : $X = U_{pi \in P} X_{pi}$

A_f est une application qui associe à chaque place P_i un système d'équations différentielles et/ou algébriques F_i :

$$F_i(\dot{X}_i, X_i, t) = \begin{bmatrix} f_{ij}(\dot{X}_i, X_i, t) \\ \vdots \\ f_{ik}(\dot{X}_i, X_i, t) \end{bmatrix}, \quad j = 1...k$$

Où :

$f_{ij}...f_{ik}$ sont des équations représentant l'évolution des variables continues. Ainsi chaque place du réseau (qui représente un état discret) peut décrire l'évolution de l'ensemble des variables continues X_i (grâce au système d'équations F_i) ;

A_e est une application qui associe à chaque transition t_i une fonction de sensibilisation e_i. La fonction de sensibilisation utilise les variables A_p qui sont associées aux places d'entrée de la transition. Le seuil e_i associé à la transition t_i est défini comme la première solution de :

$e_i(X_i, \dot{X}_i, t)=0$;

A_j est une application qui associe à chaque transition t_i une fonction de jonction j_i. Une fonction de jonction est activée lors du franchissement de la transition correspondante. La fonction de jonction j_i associée à la transition t_i, lors du franchissement de t_i à la date t calcule les valeurs des variables ainsi que leurs dérivées.

$$j_i : \begin{cases} \dot{X}(t^+) = j_{i\,\dot{X}}(\dot{X}_i, X_i, t^-) \\ X(t^+) = j_{i\,X}(\dot{X}_i, X_i, t^-) \end{cases}$$

Ce qui signifie que les valeurs des variables, et de leurs dérivées, juste après t (à t^+) sont calculées à partir de valeurs des variables, et de leurs dérivées, juste avant t (à t^-). Ceci permet de redéfinir les attributs des jetons en cas de discontinuité.

M_0 est le marquage initial. Chaque jeton est un vecteur de variables identique au vecteur de variables associé à la place. Le marquage initial définit les valeurs des variables associées au jeton à l'instant $t=0$.

I.3.4 Comparaison des méthodes d'analyse

Pour identifier la meilleure méthode pour étudier la sûreté de fonctionnement d'un système donné, nous comparons les principales caractéristiques des méthodes. Nous excluons de cette comparaison l'APR considérée plutôt comme une méthode préliminaire et la TV qui devient vite inutilisable par l'explosion de combinaisons possibles pour plusieurs états de fonctionnement et de panne des composants.

La comparaison a été faite sur un ensemble de critères [O'Connor, 2002]:

– moyens de représentations associées à la méthode (moyens spécifiques de représentation du système) ;

– système irréparable (pour un tel système, tous les composants sont considérés comme irréparables) ;

– système réparable (pour un tel système, tous les composants sont considérés comme réparables) ;

– comportement dysfonctionnel du système;

– comportement fonctionnel du système;

– modélisation par niveau du système.

Dans le tableau I.2, le signe '+' désigne le fait que la méthode possède la caractéristique et le signe '-' le contraire.

Méth. / Caract.	AMDEC	AdD	DF	MEE	RdP	AE	MCPR	MDCC
Système irréparable	+	+	+	-	+	+	+	+
Système réparable	+	+	+	+	+	+	+	+
Comportement fonctionnel du système	-	-	-	+	+	-	-	-
Comportement dysfonctionnel du système	+	+	+	+	+	+	+	+
Modélisation par niveau du système	+	+	+	+	+	+	+	+

Tab. I.2: Comparaison des méthodes

La comparaison montre que la méthode RdP présente beaucoup d'avantages, elle est la seule qui est utilisable aussi bien pour des systèmes irréparables que réparables. Elle prend en compte, d'une part, des stratégies complexes de réparation, et, d'autre part, permet de considérer à la fois le comportement dysfonctionnel et fonctionnel des systèmes. La modélisation du système par niveau est relativement simple. De plus, les RdP sont la seule méthode qui permet d'englober l'aspect dynamique.

I.4 Conclusion

Les systèmes électroniques sont de plus en plus utilisés dans l'industrie. Le développement d'un système électronique est envisagé selon l'approche de l'ingénierie concourante dans le cadre d'un cycle de développement.

La complexité importante des systèmes électroniques et la réduction des coûts de conception et d'exploitation incitent les industriels à maîtriser davantage la sûreté de fonctionnement.

Afin d'optimiser le développement de ces systèmes, des méthodes sont utilisées dans chaque étape du cycle de développement pour analyser la fiabilité. L'ensemble de ces méthodes constitue un processus à part entière : le processus de fiabilité. Les principales méthodes d'analyse de la fiabilité ont été comparées entre elles.

Les RdPs sont la méthode qui peut être utilisée dans toutes les phases d'un cycle de développement d'un système complexe, elle est la plus adaptée pour évaluer la fiabilité d'un système électronique, car elle répond, d'une part, aux principales contraintes liées à la fiabilité d'un système électronique et, d'autre part, elle est utilisable dans chaque étape du cycle de développement. En outre, la méthode RdP est la seule qui permet d'analyser le comportement fonctionnel/dysfonctionnel et l'aspect dynamique du système.

Chapitre II
Fiabilité des systèmes électroniques

II.1 Introduction

Dans le premier chapitre, nous avons montré la nécessité d'associer la fiabilité, paramètre de la sûreté de fonctionnement, à la réalisation des systèmes. De nombreuses entreprises se sont aperçues que ce paramètre est un facteur important de leur compétitivité. Les concepteurs et les utilisateurs des systèmes complexes montrent un grand intérêt pour des évaluations des différents paramètres de la sûreté de fonctionnement du système global, des parties matérielle, logicielle et des interactions entre les différentes parties du système [Borrel, 1996]. Pour les composants matériels, les défaillances ont à la base une dégradation physique [Ledoux, 1993].

II.2 Fondements de la fiabilité

Très tôt, de grandes entreprises ont montré un grand intérêt pour la fiabilité : General Motors, depuis les années 1940, la NASA, le Department of Defense au Etats-Unis, depuis les années 1950, Airbus, Air Force, Bell Telephone Laboratoires, depuis les années 1960, Thomson, Philips, Kodak, Citroën, ... depuis les années 1970 [Villemeur, 1988].

La recherche de la diminution du coût des défaillances en exploitation a entraîné une augmentation des exigences de fiabilité sur les systèmes. Ainsi, en 1995, General Electric a estimé que les coûts de non-fiabilité représentaient de 8 à 12 milliards dollars et a décidé d'augmenter le niveau de qualité de ses produits dans le cadre de la politique Six Sigma. La société Renault, quant à elle, estime qu'une modification effectuée au démarrage d'une série coûte de 10 à 100 fois plus cher que ce qu'elle aurait coûté deux ans ou 15 mois auparavant [Rieuneau, 1993].

La maîtrise de la fiabilité d'un système représente un enjeu économique important pour toute entreprise. La mesure de cette grandeur est un premier pas indispensable vers sa maîtrise [DeMarco, 1982].

La fiabilité recouvre de multiples aspects : l'analyse de défaillance des systèmes, la fiabilité prévisionnelle, les banques de données de fiabilité, les essais de fiabilité, la fiabilité opérationnelle, les méthodes prévisionnelles de fiabilité et de sécurité, l'assurance de la fiabilité et de la qualité [Villemeur, 1988].

II.2.1 Les mesures associées à la fiabilité

La *"qualité"* du matériel, du point de vue de la fiabilité, est donnée par un certain nombre d'indicateurs ou mesures de la performance qui sont présentés dans ce paragraphe.

II.2.1.1 Fonction de répartition, Densité de probabilité

On considère un système (une carte électronique, un moteur, une voiture, un avion,...) pouvant se trouver dans différents états. Cet ensemble d'états, noté E, se décompose en deux sous ensembles formant une partition : le sous-ensemble M des états de marche et le sous-ensemble D des états de défaillance.

Considérons T la variable aléatoire qui représente le temps écoulé entre la mise en service d'une entité et la première défaillance observée. La fiabilité à l'instant t est la probabilité qu'une entité E soit non défaillante sur la durée [0; t].

On appelle également fiabilité, la probabilité associée R(t) définie par :

$$R(t) = P\,(t < T) \qquad \textbf{(II.1)}$$

La figure **II.1** ci-dessous présente une allure de la fonction R(t) en fonction du temps.

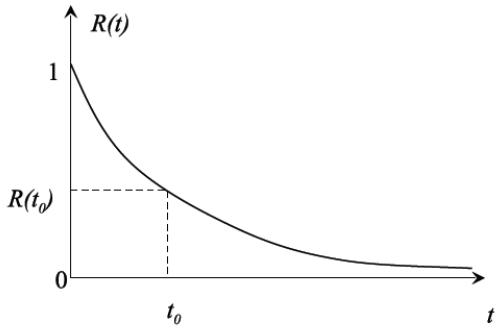


Fig. II.1 – Courbe de survie ou de fiabilité.

Pour compléter l'approche théorique de la notion de fiabilité, il est nécessaire de définir aussi les notions suivantes, qui sont issues de la théorie des probabilités.

La fonction F(t) **(II.2)** représente la fonction de répartition de la variable aléatoire T. C'est la "défiabilité" R(t) (la probabilité de défaillance du système) ou la probabilité complémentaire à 1 de la fiabilité R(t) définie par :

$$F\,(t) = P\,(t \geq T) = 1 - R(t) \qquad \textbf{(II.2)}$$

La fonction *f(t)* **(II.3)** désigne la densité de probabilité de t et elle est donnée par :

$$f(t) = dF(t)/dt = -\,dR(t)/dt \qquad \textbf{(II.3)}$$

La fonction de répartition F(t) et la fonction de fiabilité R(t) sont exprimées à partir de la fonction de densité f(t) dans les relations suivantes **(II.4)** :

$$F(t) = \int_0^t f(u)du$$

$$R(t) = 1 - F(t) = 1 - \int_0^t f(u)du = \int_t^\infty f(u)du \quad \textbf{(II.4)}$$

II.2.1.2 Taux de défaillance instantané

Le taux instantané de défaillance, λ(t), est une des mesures caractéristiques de la fiabilité. La valeur λ(t)dt représente la probabilité conditionnelle d'avoir une défaillance dans l'intervalle de temps [t; t + dt], sachant qu'il n'y a pas eu de défaillance dans l'intervalle de temps [0; t]. Ainsi, en appliquant le théorème des probabilités conditionnelles, puis le théorème des probabilités totales, λ(t) s'écrit :

$$\lambda(t)dt = \frac{Prob(défaillant\ sur\ [t; t+dt]\ sans\ défaillance\ sur\ [0;t])}{Prob(non\ défaillant\ sur\ [0;t])}$$

$$\lambda(t)dt = \frac{Prob(défaillant\ sur\ [0; t+dt]) - Prob(défaillant\ sur\ [0;t])}{Prob(non\ défaillant\ sur\ [0;t])}$$

$$\lambda(t) = \frac{f(t)}{R(t)} = -\frac{1}{R(t)} \cdot \frac{dR(t)}{dt} \quad \textbf{(II.5)}$$

On déduit que :

$$R(t) = \exp\left[-\int_0^t \lambda(u)du\right] \quad \textbf{(II.6)}$$

II.2.1.3 Métriques de la SdF

Des grandeurs associées à la SdF peuvent être calculées à partir des mesures de probabilités. Contrairement aux précédentes, qui sont fonction du temps, les grandeurs suivantes caractérisent des durées moyennes [Villemeur, 1988] :

– MTTF : Durée moyenne de fonctionnement d'une entité avant la première défaillance (anglais Mean Time To Failure)

$$MTTF = \int_0^\infty R(t)\,dt \quad \textbf{(II.7)}$$

– MTTR : Durée moyenne de réparation (anglais Mean Time To Repair)

$$MTTR = \int_0^\infty [1 - M(t)]\,dt \quad \textbf{(II.8)}$$

– MUT : Durée moyenne de fonctionnement après réparation (anglais Mean Up Time)

– MDT : Durée moyenne d'indisponibilité après défaillance (anglais Mean Down Time)

– MTBF : Durée moyenne entre deux défaillances (anglais Mean Time Between Failure)

$$MTBF = MDT + MUT \quad \textbf{(II.9)}$$

Ces durées sont représentées dans la figure **II.2** :

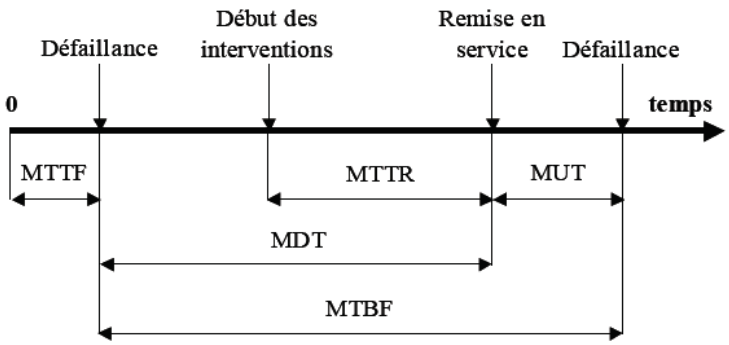


Fig. II.2 – Durées moyennes associées à la SdF

II.2.2 Les mécanismes de défaillance

II.2.2.1 Le taux de défaillance

Tout système contient inévitablement des fautes qui se manifesteront potentiellement par l'apparition de défaillances au cours de la vie opérationnelle du système.

Il est donc important de connaître les mécanismes de défaillance pour déterminer l'architecture optimale d'un système et pour évaluer ses paramètres de la sûreté de fonctionnement d'où sa fiabilité.

La fiabilité des systèmes, des sous-ensembles et des composants est souvent décrite par la courbe caractéristique dite en *"baignoire"* (figure **II.3**). Elle décrit l'évolution du taux de défaillance ***λ(t)*** en fonction du temps ***t*** et permet de mettre en évidence, de manière empirique, trois phases de la vie du produit.

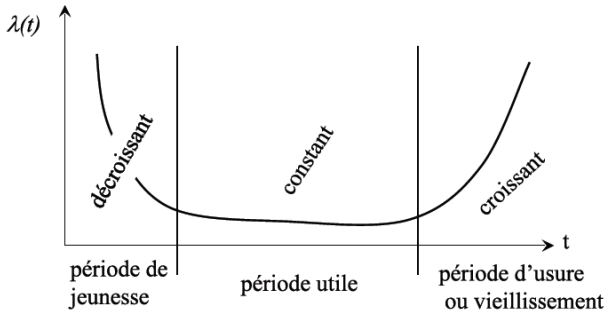


Fig. II.3 – Courbe en baignoire

Pour les composants électroniques la courbe, représentant le taux de défaillance, a la même allure que la courbe en baignoire (figure **II.3**). Elle est donc composée de trois phases nettement distinctes :

– ***La première phase*** définit la période de jeunesse, caractérisée par une décroissance rapide du taux de défaillance. Pour un composant électronique cette décroissance s'explique par l'élimination progressive des défauts dus aux processus de conception ou de fabrication mal maîtrisé ou à un lot de composants défectueux. Aujourd'hui, cette période est réduite, compte

tenu de la grande qualité des composants. Les distributions de probabilité utilisées pour ces conditions sont la loi de *Weibull (β< 1)* et la loi *lognormale (σ> 1)*.

– ***La deuxième phase*** définit la période de vie utile généralement très longue. Le taux de défaillance est approximativement constant. Les pannes sont dites aléatoires, leur apparition n'est pas liée à l'âge du composant mais à d'autres mécanismes d'endommagement. Le choix de la loi *exponentielle* est tout à fait satisfaisant dans cette phase.

– ***La dernière phase*** est la période de vieillissement, elle est caractérisée par une augmentation progressive du taux de défaillance avec l'âge du système. Cette augmentation est due aux phénomènes de vieillissement tels que l'usure, l'érosion, etc. Cette période est très nettement au-delà de la durée de vie réelle d'un composant électronique.

Les distributions de probabilité utilisées pour ces conditions sont la loi de *Weibull (β> 1)* et la loi *lognormale (σ< 1)*.

II.2.2.2 Défaillance – Cause de défaillance – Mode de défaillance – Effet de défaillance – Mécanisme de défaillance

Une défaillance (ou panne) est la cessation de l'aptitude d'une entité ou d'un système à accomplir une fonction requise. Elle résulte d'une ou plusieurs fautes (écart anormal avec la caractéristique désirée), ou encore anomalie de fonctionnement. Une défaillance peut être complète, partielle, fugitive, intermittente ou permanente. Elle peut conduire à la panne du système [Mavier.J, 2007].

La cause de défaillance est l'ensemble de circonstances associées à la conception, la fabrication ou l'emploi qui ont entrainé une défaillance [Mazouni M.H, 2008].

Le mode de défaillance est le cheminement d'un défaut initial vers une défaillance "visible" du système [Vallon, 2003].

L'effet de défaillance est l'ensemble de conséquences qu'un mode de défaillance a sur une opération, une fonction ou un état [MIL-HDBK, 1995].

Le mécanisme de défaillance est tout processus physique, chimique, électrique ou encore thermique, qui a comme conséquence une défaillance [MIL-HDBK, 1995].

II.2.2.3 Les mécanismes de défaillances

Les mécanismes de défaillance constituent les briques de base dans l'étude de la sûreté de fonctionnement tout au long du développement d'un système donné. Or, les composants

électroniques présentent des mécanismes de dégradation complexes telles que les charges de surface, la polarisation, le décollement de fils de connexion, la migration métallique, l'électro-migration, le défaut de silicium, ... [Nelson, 1990], [Ye et al., 2003], [Bechou et al., 2007].

- ✓ Les charges de surface représentent la présence de charges en surface des oxydes de grille. Ce mécanisme intervient dans la dérive de la tension de seuil ;
- ✓ La polarisation suppose la présence de molécules polarisables dans l'oxyde de grille. Elle entraîne la dérive de la tension de seuil ;
- ✓ Le décollement de fils de connexion est lié directement au processus d'assemblage et provoque souvent des courts-circuits ;
- ✓ La migration métallique est le déplacement des atomes du métal dans le silicium et entraîne des courts-circuits ou des circuits ouverts ;
- ✓ L'électro-migration représente le déplacement des atomes dans les couches métalliques et provoque des courts-circuits ;
- ✓ Un défaut de silicium est provoqué par des impuretés, des défauts de structure ou bien par des états de surface et entraîne des courts-circuits.

Ces mécanismes de défaillances agissent principalement sur les caractéristiques suivantes du composant :

- ✓ la tension de seuil (dérive provoquée par surcharge, perte ou inversion de charge, polarisation, ...) ;
- ✓ les circuits (ouverture/fermeture par impureté, défaut de structure dans le matériel, oxydation, ...).

II.3 Estimation de la fiabilité prévisionnelle des systèmes électroniques

Dans la fiabilité prévisionnelle, l'estimation des paramètres des lois de fiabilité est nécessaire pour réagir le plus tôt possible sur la conception du système. Cette estimation s'effectue à partir de modèles et de données appropriées à chaque technologie du système.

II.3.1 Les réseaux de Pétri

Les réseaux de Pétri sont des modèles état-transition. Leurs intérêts et qu'ils permettent d'exprimer de manière aisée les mécanismes de parallélisme, de synchronisation, de partage ou d'assemblage de ressources grâce au concept de marquage. L'intérêt est de pouvoir modéliser le comportement du système sans connaître à priori l'ensemble de ses états.

Ils peuvent être utilisés à la fois pour des analyses de type qualitatif (vérification de propriétés) ou pour des analyses de type quantitatif, comme l'évaluation de performances fonctionnelles ou de sûreté de fonctionnement.

Les réseaux de Pétri sont utilisés pour modéliser des systèmes évoluant dans le temps. Ces évolutions sont mises en évidence grâce aux marques. L'ensemble de marques forme le marquage. Le marquage définit à un instant donné l'état du système. L'ensemble de transitions représente quant à lui l'ensemble des événements dont les occurrences provoquent des modifications sur l'état du système. Chaque marquage peut être représenté par un vecteur et le réseau peut être modélisé comme un ensemble d'équations algébriques. L'état du système est alors décrit par son graphe de marquages.

Il existe plusieurs classes de réseaux de Pétri. Les réseaux de Pétri stochastiques associent à chaque transition un taux de franchissement aléatoire. Si tous les taux sont constants (distribution exponentielle de la durée de franchissement) alors le graphe de marquage est homogène à un graphe de Markov.

Très souvent cette hypothèse ne peut être satisfaite en raison de la complexité des phénomènes à modéliser. On utilise alors les modèles de type état-transition (graphe d'état ou RdP) comme supports de simulation par des méthodes de Monte Carlo [Perèz.A, 2009].

II.3.2 Utilisations et recueils des données de fiabilité

Lorsqu'une équipe de conception développe un nouveau système, elle doit disposer d'un certain nombre de recommandations ou de règles, issues du savoir-faire (de l'expérience) qui sont indispensables à une conception sûre de fonctionnement. Les recueils des données sont des outils incontournables et indispensables.

En électronique, un domaine où le calcul de la fiabilité est pratiqué depuis de nombreuses années, les bases de données de fiabilité sont disponibles et nombreuses. Il y a une certaine difficulté à constituer ces recueils, compte tenu de la complexité des composants. Néanmoins, ils sont de plus en plus utilisés.

En pratique, on utilise souvent des bases de données connues ou, mieux encore, quand cela est possible, les données issues de retours d'expériences auprès des fabricants des composants [H.Belhadaoui, Janvier 2011].

Les recueils de données de fiabilité les plus connus pour des dispositifs électroniques sont présentés dans le tableau **II.2**.

Ces recueils sont mis à jour régulièrement pour tenir compte des évolutions technologiques.

En électronique, la durée de vie des données fournies par les recueils est relativement courte (de 3 à 6 ans).

II.3.3 Données de fiabilité disponibles pour les composants électroniques

Les données numériques sur les taux de défaillance des différents composants électroniques sont essentielles pour mener à bien les prévisions de fiabilité.

Ces données doivent êtres sûres. Les différents recueils de données sont malheureusement basés sur des critères de défaillance différents et sur des utilisations peu comparables, si bien que les taux minimaux, moyens et maximaux provenant de quatorze sources d'informations différentes ; il a été constaté que des variations de l'ordre de 1000 peuvent exister entre différentes données, publiées avant 1963 aux États-Unis (Tableau **II.1**).

Il est évident qu'en fonction des applications, on devra disposer, pour effectuer des prévisions convenables, de données mieux adaptées.

Taux de défaillance horaire / Composant	Mini	Moyen	Maxi	Rapport
Condensateur	1.10^{-8}	300.10^{-8}	1000.10^{-8}	1000
Diode silicium	$15.\ 10^{-8}$	$154.\ 10^{-8}$	$385.\ 10^{-8}$	26
Moteur C.A	$15.\ 10^{-8}$	$6700.\ 10^{-8}$	$55600.\ 10^{-8}$	3700
Résistance à couche	$2.\ 10^{-8}$	$40.\ 10^{-8}$	$100.\ 10^{-8}$	50
Relais	$10.\ 10^{-8}$	$2300.\ 10^{-8}$	$12500.\ 10^{-8}$	1250
Transistors	$10.\ 10^{-8}$	$730.\ 10^{-8}$	$1700.\ 10^{-8}$	170

Tab. II.1: Taux de défaillance des différents composants électroniques

Il existe un certain nombre de sources de données sur les taux de défaillance (Tableau **II.1**). Parmi les plus importantes, on peut citer le **MIL-HDBK-217** préparé par R.C.A. sur un contrat de l'U.S. Air Force et le rapport A.R.I.N.C n°203-1-344.

Les données du **MIL-HDBK-217** ont été reprises par le centre de fiabilité du **C.N.E.T**. et publiées dans la revue *« Fiabilité »* [MIL-HDBK, 1995].

Source	Titre	Editeur
FIDES	Méthodologie de fiabilité pour les systèmes électroniques.	DGA-DM/STTC/CO/477
IEEE STD	IEEE Guide to the Collection and Presentation of Electrical, Electronic Sensing Component and Mechanical Equipment Reliability Data for Nuclear Power Generating Stations.	Institution of Electrical and Electronic Engineers, New York, USA.
MIL-HDBK-217	Military Handbook Reliability Prediction of Electronic Equipment.	United States Department of Defense.
BT-HRD	Handbook for Reliability Data.	British Telecommunications.
EPRD	Electronic Parts Reliability Data.	Reliability Analysis Center, RAC, New York, USA.
GJB	Chinese Military Standard.	Beijing Yuntong Forever Sci.-Tech.
RDF(CNET)	Recueil de données de fiabilité.	Centre National d'Etudes des Télécommunications UTE, Paris, France.
Telcordia/Bellcore	Reliability Prediction Procedure for Electronic Equipment.	Telcordia Technologies, New Jersey, USA.

Tab. II.2: Recueils de données de fiabilité en électronique

II.3.4 Exemple de données de base extraites du MIL-HDBK-217

II.3.4.1 Semi-conducteur

Les taux de défaillance minimaux annoncés pour *les diodes* sont de 10.10^{-8} défaillances par heure et de 20.10^{-8} pour *les transistors*. Ces chiffres sont probablement convenables pour des diodes à pointe au germanium de fabrication relativement ancienne et pour des transistors à jonctions alliées au germanium, mais ne sont pas représentatifs des composants actuels [MIL-HDBK, 1995].

II.3.4.2 Résistances

On trouvera ci-dessous les taux de défaillance indiqués pour une puissance égale à la puissance nominale de la résistance et pour une température de 40 °C [MIL-HDBK, 1995].

Résistance agglomérées : 6.10^{-8} h^{-1}

Résistance à couche de carbone 65.10^{-8} h-1

Résistance bobinées de puissance 220.10^{-8} h^{-1}

II.3.5 Modèle de fiabilité prévisionnelle pour les composants électroniques : FIDES

L'expression du taux de défaillance dépend de plusieurs facteurs dont la technologie de conception, la technologie de fabrication et l'environnement de fonctionnement du composant [FIDES, 2004]. Ainsi, le taux de défaillance dépend d'un taux de défaillance de base, pondéré par des facteurs de technologie, de conception, de fabrication, d'utilisation, d'environnement,...

Actuellement, dans le domaine de l'électronique, ces calculs sont principalement réalisés en se basant sur des standards tels que la MIL-HDBK-217 [MIL-HDBK, 1995].

De fait, il est courant que les concepteurs appliquent des facteurs correctifs "fait maison" basés sur des retours d'expérience «Rex», des habitudes d'utilisation, des coutumes ou des avis d'experts.

Les différents standards existants (MIL-HDBK- 217, UTEC 80-810, PRISM, BellCore, British Telecom HRD, …).

De part sa notoriété, le manuel de prédiction de fiabilité MILHDBK-217, est devenue aujourd'hui une exigence incontournable dans la majorité des spécifications techniques. Il comprend un ensemble de modélisations de taux de défaillances pour de nombreux composants électroniques tels que les circuits intégrés, les transistors, les diodes, les résistances, les condensateurs, les relais, les Switch, les connecteurs, etc.

L'évaluation du taux de défaillance d'un composant λ_p est basée sur le principe suivant :

$$\lambda p = \lambda_b \, \pi T \, \pi Q \, \pi E \, \pi S \, \pi R \, \pi C$$

λ_b représente le taux de défaillance de base du composant (exprimé à partir d'un modèle classique par exemple la loi exponentiel) tandis que les facteurs π traduisent des contraintes telles que la température d'emploi, la qualité de fabrication, des contraintes environnementales, etc. Le calcul du taux de panne d'une carte est basé sur un modèle série des composants considérés.

FIDES est née en 2004, c'est une méthodologie globale d'ingénierie de la fiabilité en électronique [FIDES, 2004], qui porte sur une évaluation prévisionnelle de la fiabilité et sur une maîtrise du processus de fiabilité. Au travers des paramètres technologiques, physiques ou de processus, FIDES permet d'agir sur l'ensemble du cycle de vie des produits pour améliorer et maîtriser leur fiabilité.

L'équation générale de FIDES est :

$$\lambda = \lambda_{Physique} \prod Part-manufactur\,ing \prod \Pr ocess$$

Où :

- $\lambda_{physique}$ représente la contribution physique ;
- $\Pi_{Part\_manufacturing}$ traduit la qualité et la maîtrise de fabrication du composant ;
- $\Pi_{Process}$ représente le processus de développement, de fabrication et d'utilisation du produit.

L'objectif de FIDES est de permettre une évaluation réaliste de la fiabilité des composants électroniques.

II.3.6 Données du retour d'expériences (REX)

La méthodologie d'estimation de la fiabilité opérationnelle s'appuie sur les données obtenues par retour d'expériences (REX).

Le REX vise une meilleure connaissance du comportement des composants, de leurs modes de dégradation, de dysfonctionnement ou d'endommagement. Il est basé sur la collecte et la gestion des faits techniques, observés pendant toute la durée de vie du système, de sa mise en service jusqu'à sa désintégration. Dans ce contexte, la constitution d'un échantillon correct de données pour l'estimation d'un modèle de vie d'un système passe par la reconstruction de l'histoire complète d'exploitation de ce système, pour une période d'observation déterminée.

Plusieurs précisions sont à définir sur les données à collecter en exploitation :

- Systèmes qui font l'objet de cette collecte,

- Informations à collecter pour ces systèmes,
- Structure de données adoptée,
- Niveau de qualité souhaitable pour la collecte.

Exploitation des données du REX suppose trois phases :

- Sélection, parmi les systèmes suivis, d'un échantillon pertinent,
- Reconstruction de l'histoire de chaque matériel pris en compte,
- Calcul des temps jusqu'à la défaillance ou jusqu'à censure pour l'échantillon choisi.

La fonction de vraisemblance nécessite l'accès à des données de défaillances issues des banques de données de REX. Elle permet ainsi de déterminer *la loi de fiabilité* observée du système à partir des observations de défaillance.

En même temps, l'objectif du REX est *l'amélioration de la qualité et de la fiabilité* des systèmes et pour cet objectif le traitement de données brutes sera indispensable. La qualification et la validation de données du REX sont nécessaires avant tout calcul. Ces deux étapes représentent la tâche la plus difficile et la plus critique du REX, mais elles sont indispensables pour obtenir des résultats crédibles.

Les données nécessaires pour établir un échantillon pour construire la vraisemblance sont :

- Date de la mise en service du système,
- Dates de défaillances du système,
- Dates de censure, la fin d'observation des essais.

Toutes les banques de données du REX doivent permettre l'extraction de ces données. La construction d'un échantillon des données collectées par le REX pour estimer le modèle de vie associé au système passe par la valorisation de l'exploitation du système pendant une durée d'observation. Cette durée est plutôt courte afin de permettre le calcul de la fiabilité opérationnelle le plus vite possible. Contrairement aux données d'essais, la taille d'échantillon est souvent importante pour les données issues du REX.

II.4 Conclusion

La fiabilité d'un système électronique se construit, s'analyse et s'évalue. La capitalisation et la consolidation des connaissances tout au long du cycle de développement du système constituent le principe fondateur d'une méthodologie globale d'évaluation de la fiabilité. La modélisation fonctionnelle et dysfonctionnelle du système, indispensable à toute maîtrise de la fiabilité, est réalisée par réseaux de Pétri.

La fiabilité prévisionnelle nécessite la connaissance de tous les mécanismes de défaillance des composants pour chaque technologie intégrée dans le système et des lois de fiabilité associées.

La conclusion qui s'impose, au point de vue prédictions de fiabilité, est de ne pas utiliser un recueil de données sans être assuré que les composants utilisés sont de même nature que ceux sur lesquels ont été relevés les taux de défaillance.

Chapitre III
Méthodes d'Analyses de la Sûreté de Fonctionnement : de l'AMDEC à l'Arbre de Décision Binaire

III.1 Introduction

Ce chapitre présente les trois méthodes d'analyse de la sûreté de fonctionnement : l'AMDEC, l'analyse par l'Arbre de Défaillance et l'analyse en utilisant les Diagrammes de Décision Binaires (BDDs)

La plupart des concepts mathématiques et outils algorithmiques sur les BDDs sont des notions classiques de l'algèbre de Boole et du calcul des probabilités. Ils ne sont donc en rien originaux puisqu'ils sont décrits dans toutes les monographies sur la sûreté de fonctionnement [Pagès et Gondran, 1980],[Villemeur, 1988],[Limnios, 1991],[Cocozza, 1997].

La notion de « **coupe minimale** » est en revanche présentée. Cette notion, qui est pourtant au cœur des méthodes booléennes d'analyse de la sûreté de fonctionnement, n'avait pas, jusqu'à une date récente, été formellement définie.

III.1 Méthode d'Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités (AMDEC)

L'AMDEC tient une place importante en sûreté de fonctionnement car c'est une méthode permettant de formaliser un grand nombre de connaissances sur les installations industrielles. La méthode AMDEC [Monchy, 2000] est une méthode qualitative et inductive visant à recenser les défaillances [Zwingelstein, 2009], puis à en estimer les risques.

L'AMDEC est vue comme un outil rentrant dans une stratégie d'amélioration permanente de la fiabilité et de la disponibilité des composants électroniques [Renson, 2003]. La méthode est basée sur quatre étapes séquentielles: la décomposition fonctionnelle, l'analyse qualitative, l'analyse quantitative et finalement une proposition d'actions correctrices.

Un rapport d'analyse ou grille AMDEC résume le bilan de l'étude de l'analyse des défaillances. La description fonctionnelle adoptée est basée sur une décomposition arborescente et hiérarchique de la structure de l'équipement. L'étape suivante d'analyse qualitative consiste à faire une identification des modes de défaillances, de leurs effets et des causes conduisant au dysfonctionnement d'un élément du système. Ces trois notions sont liées par la relation suivante : Cause → Mode → Effet. La troisième étape d'analyse quantitative permet le calcul de l'indice de criticité C, déduit par le produit des trois indices nominaux O, G et D, représentant respectivement l'indice de l'occurrence d'une défaillance due à une cause

particulière, l'indice de gravité des effets de la défaillance et l'indice relatif à la possibilité de détecter la défaillance avant qu'elle ne produise l'effet.

La criticité C calculée permet de recenser les défaillances dont le niveau de criticité est supérieur à un seuil qui varie en fonction de critères de fiabilité [Monchy, 2000]. Après le calcul de la criticité C et sa comparaison avec le seuil établi, différentes actions correctrices sont menées qui permettent d'améliorer les valeurs des indices précédents et de définir les indices finaux O', G' et D' sur la même base que l'évaluation précédente. La nouvelle criticité C' calculée quantifie l'amélioration réalisée [Noureddine M, 2010].

Le résultat de l'étude AMDEC est présenté dans un tableau récapitulant tous les défauts pouvant apparaître sur cette cellule de commutation et tous les modes de défaillance qui en découlent. A chaque fois, le mode de défaillance est analysé (cause, effet, gravité et criticité). De manière très schématique, l'AMDEC se déroule suivant l'algorithme : [Mazouni M.H, 2008]

Début_AMDEC

Pour chaque composant d'un système faire

Pour toutes les fonctions d'un composant faire

Pour chaque mode de défaillance faire

Identifier les causes ( ) ;

Identifier les effets ( ) ;

Evaluer la criticité : C = D * O * G ;

Tant que C > seuil faire

Opter pour des actions correctives ( ) ;

Réévaluer la nouvelle criticité : C' = D' * O' * G' ;

Fin Tant que ;

Fin Pour ;

Fin Pour ;

Fin Pour ;

Fin_AMDEC.

III.2 Méthode de l'Arbre de Défaillances (AdD)

La démarche consiste à s'appuyer sur la connaissance des éléments constitutifs du système étudié ainsi que les différents modes et causes de défaillance pour identifier tous les scénarios conduisant à l'événement redouté.

L'analyse par arbre de défaillances est une étude prioritaire des défaillances relatives à des événements redoutés dont la gravité de production est plus significative. Elle permet de considérer des combinaisons d'évènements pouvant conduire à un évènement redouté.

La méthode exige une parfaite connaissance des scénarios événementiels et donc du fonctionnement du système et de son interaction avec son environnement.

L'algorithme suivant montre d'une manière systématique le déroulement d'une analyse par Arbre de Défaillances: [Mazouni M.H, 2008]

Début_AdD

Considérer un événement final Top : Z ;

Label : Pour un événement intermédiaire faire

Identifier les causes ( ) ;

Définir les événements en question ( ) ;

Lier ces événements à l'aide des portes logiques ( ) ;

Pour chaque événement faire

Si cet événement est décomposable alors

Goto Label ;

Fin Si ;

Fin Pour ;

Fin Pour ;

Fin_AdD

III.3 Méthode de l'Arbre de Décision Binaire

III.3.1 Fonctions et formules booléennes

On note B l'ensemble {0, 1} des *valeurs booléennes* (0 pour faux et 1 pour vrai). Les *variables booléennes* prennent leurs valeurs dans B et sont notées par des lettres latines en italique, *x*, *y*, *z*, éventuellement indicées : x_1, x_2,... Les ensembles de variables booléennes sont notés par des lettres latines majuscules en italique : *X, Y, Z.*

III.3.1.1 Définition 1

Une *fonction booléenne f* sur un ensemble de variables booléennes $X = \{x_1,..., x_n\}$ est une fonction de B^n dans B.

Soit $X = \{x_1,..., x_n\}$ un ensemble de variables booléennes (supposé ordonné). Une *affectation* de *X* est un élément σ de B^n. On note σ $[x_i]$ la valeur de la variable booléenne x_i dans l'affectation σ (c'est-à-dire la $i^{ème}$ composante du vecteur σ).

Les fonctions booléennes sont des objets mathématiques abstraits. Pour les manipuler concrètement, on utilise une syntaxe (ou, autrement dit, des structures de données) : les formules booléennes.

III.3.1.2 Définition 2

L'ensemble F des *formules booléennes* est construit par induction structurelle à partir des deux constantes booléennes 0 et 1, d'un ensemble fini ou dénombrable de variables booléennes *X*, des connecteurs logiques usuels Λ (et), V (ou), ¬ (négation) et des parenthèses (et). C'est le plus petit ensemble tel que :

- 0 et 1 sont des formules.
- Les variables de *X* sont des formules.
- Si *f* et *g* sont des formules, alors *f* Λ *g*, *f* V *g*, ¬f et ¬g sont aussi des formules.

Les parenthèses sont utilisées pour délimiter les sous-formules.

Par exemple, $f = (x \Lambda y) V (\neg x \Lambda z)$ est une formule booléenne construite à partir de l'ensemble de variables $X = \{x, y, z\}$.

Le lien entre fonctions et formules booléennes est fait via les tables de vérité des connecteurs Λ, V et ¬. On étend la notion d'affectation aux formules. Soient σ une affection sur *X* et *f* et *g* deux formules construites sur *X*. On pose σ [1]=1, σ [0]=0. La valeur d'une formule est

calculée récursivement à partir de la valeur de ses sous-formules comme indiqué par la table suivante :

f	g	$\neg f$	$\neg g$	$f \wedge g$	$f \vee g$	$f \Rightarrow g$	$f \Leftrightarrow g$
1	1	0	0	1	1	1	1
1	0	0	1	0	1	0	0
0	1	1	0	0	1	1	0
0	0	1	1	0	0	1	1

Tab. III.1: Table de vérité de différentes formules usuelles

Une affectation σ *satisfait* la formule *f* (respectivement *falsifie f*) si $f(\sigma) = 1$ (resp. $f(\sigma) = 0$). Une formule *f* est *satisfiable* s'il existe une affectation qui la satisfait. Sinon elle est dite *insatisfiable* (on dit aussi que c'est une *antilogie*). Si les 2^n affectations qu'il est possible de construire sur les *n* variables satisfont *f*, *f* est une *tautologie*.

La table ci-dessus comporte deux autres connecteurs logiques l'implication => et la bi-implication (ou équivalence) <=>. Ces deux connecteurs peuvent être vus comme des raccourcis d'écriture. Ainsi $f => g$ est équivalent à $\neg f \vee g$ et $f <=> g$ est équivalent à $(f => g) \wedge (g => f)$.

D'autres connecteurs sont utiles dans le cadre des modèles booléens d'analyse de la sûreté:

- L'opérateur « *k* sur *n* », noté $@k(f_1, ..., f_n)$, où $f_1, .., f_n$ sont des formules. Une affectation σ satisfait $@k(f_1, ..., f_n)$ si elle satisfait au moins *k* des f_i.
- L'opérateur de *cardinalité*, noté $\#(l,h)(f_1, ..., f_n)$, où $f_1, .., f_n$ sont des formules. Une affectation s satisfait $\#(l,h)(f_1, ..., f_n)$ si elle satisfait au moins *l* et au plus *h* des f_i.
- L'opérateur « *si-alors-sinon* », noté *ite*(*f*,*g*,*h*) (pour *If-Then-Else)*. Il est équivalent à la formule $(f \wedge g) \vee (\neg f \wedge h)$. Il peut être lu : si *f* est vrai alors *g* sinon *h*.

Tous les connecteurs mentionnés jusqu'à présent travaillent sur un ensemble de variables *X*. Il est parfois utile de définir des opérateurs de projection : Soient $Y = \{x_1, ..., x_n, y\}$ un ensemble de variables booléennes et *f* une formule construite sur *Y*.

- Le *cofacteur positif* (respectivement *négatif*) de *f* par rapport à *y*, noté $f_{y=1}$ (respectivement $f_{y=0}$) est une formule telle que pour toute affectation σ de $X = \{x_1, ..., x_n\}$, σ satisfait $f_{y=1}$ si et seulement si σ *[y=1]* (respectivement σ *[y=0]*) satisfait *f*, où σ *[y=v]* dénote l'affectation de *Y* donnant la même valeur que σ aux x_i et la valeur *v* à *y*.
- Le quantificateur *existentiel* de *y* dans *f*, noté $\exists y\ f$ est une formule équivalente à la formule $f_{y=1} \vee f_{y=0}$.
- Le quantificateur *universel* de *y* dans *f*, noté $\forall y\ f$ est une formule équivalente à la formule $f_{y=1} \wedge f_{y=0}$.

III.3.2 Diagrammes de décision binaires

Les diagrammes de décision binaires de Bryant constituent la structure de données la plus efficace connue à ce jour pour coder et manipuler des fonctions booléennes. Depuis leur introduction dans le domaine de la sûreté de fonctionnement, les BDDs se sont effectivement avérés être la technique la plus efficace pour traiter les modèles booléens d'analyse de sûreté de fonctionnement.

Le BDD associé à une formule booléenne est un codage compact de la table de vérité de cette dernière. Ce codage est fondé sur la décomposition de Shannon des formules booléennes.

En choisissant un ordre total sur les variables et en appliquant récursivement la décomposition de Shannon, la table de vérité de toute formule peut être représentée graphiquement par un arbre binaire. Chaque nœud interne de cet arbre code une formule f et peut se lire comme un opérateur si-alors-sinon (figure **III.1**).

La valeur de la formule pour une certaine affectation est obtenue en descendant le long de la branche correspondante à partir du nœud racine de l'arbre jusqu'à une feuille.

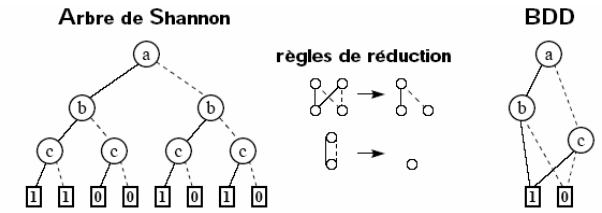


Fig. III.1 – Transformation d'arbre de Shannon en BDD

Bien sûr, une telle représentation est très coûteuse en place mémoire. Il est toutefois possible de la réduire considérablement en utilisant les deux règles suivantes :

- ✓ le partage des sous-arbres isomorphes. Puisque deux sous-arbres isomorphes codent la même fonction, un seul suffit. En particulier, on aura besoin que de deux feuilles (une étiquetée par 0, l'autre par 1) ;
- ✓ la suppression des nœuds inutiles. Un nœud dont les deux arêtes sortantes pointent sur le même nœud code la même fonction que ce dernier ($f = (\neg x \wedge g) \vee (x \wedge g)$). Un tel nœud est donc inutile.

En appliquant ces deux règles aussi souvent que nécessaire, on obtient le BDD associé à la formule.

III.3.3 Coupes minimales

Dans les modèles booléens d'analyse de sûreté, les variables représentent la survenue de pannes de composants élémentaires et les formules décrivent les pannes des systèmes en fonction de ces dernières. Cela a au moins deux conséquences :

- ✓ d'une part, il est intéressant de connaître les jeux minima de pannes de composants élémentaires qui induisent une panne du système étudié. C'est cette idée de jeu minimum que capture la notion de coupe minimale ;
- ✓ d'autre part, il existe une asymétrie fondamentale entre les littéraux positifs (qui représentent la survenue de pannes) et les littéraux négatifs (qui représentent leur non-survenue et sont donc, en un sens, moins porteurs d'information). Pour cette raison, la notion classique d'implicant premier, qui formalise l'idée de solution minimale d'une formule booléenne, n'est pas suffisante dans le cadre de l'analyse de sûreté de fonctionnement.

Soient f et g deux formules booléennes construites sur un ensemble X de variables. Si tous les minitermes de f sont des minitermes de g, on dit que f implique g et on note f |= g. Un implicant de f est un produit π, tel que π |= f.

Un implicant est premier s'il est minimal pour l'inclusion, c'est-à-dire s'il n'existe pas de sous-ensemble strict de π qui soit un implicant de f. On note PI[f] l'ensemble des implicants premiers d'une formule f (PI pour « Prime Implicants »). Toute formule est équivalente à la disjonction de ses implicants premiers.

Soit, par exemple, la formule f=(a Λ b) V (¬a Λ c) construite sur X={a,b,c}. f admet les implicants suivants : a Λ b Λ c, a Λ b Λ ¬c, ¬a Λ b Λ c, ¬a Λ ¬b Λ c, a Λ b, ¬a Λ c et b Λ c. Parmi eux, seuls les trois derniers sont des implicants premiers.

Cet exemple permet d'illustrer la critique « fiabiliste » de la notion d'implicant premier. Le produit $\neg a \wedge c$ contient un littéral négatif ($\neg a$) qui n'est pas porteur d'une information bien intéressante. En fait, on ne voudrait garder de ce produit que c. Mais on a $b \wedge c \models c$ et donc $b \wedge c$ n'est pas minimal (ou premier).

L'ensemble des coupes minimales que l'on souhaite obtenir est donc { a Λ b, c}. Intuitivement, une coupe minimale est donc une partie positive minimale d'une affectation satisfaisant f (d'une solution de f) : si les pannes représentées par cette partie positive sont survenues et si aucun autre composant n'est en panne, alors le système est en panne. Il est surprenant de constater que, bien que les modèles booléens d'analyse de la sûreté de fonctionnement sont utilisés depuis de nombreuses années, la notion de coupes minimales n'a été formalisée que très récemment.

III.3.4 Zéro Suppressed BDDs

Pour travailler avec les coupes minimales et les implicants premiers, on a besoin de coder des ensembles de produits. Les BDD ne peuvent pas servir directement à coder de tels ensembles. En effet, une variable peut apparaître positivement, négativement ou pas du tout dans un produit. Il faudrait donc des diagrammes ternaires de décision. Une autre solution, qui s'est imposée, consiste à utiliser les BDD, mais en leur donnant une autre sémantique. C'est la notion de « Zero-Suppressed BDD » (ZBDD) proposée par Minato [Minato, 1993]. L'idée de Minato est d'étiqueter les nœuds par des littéraux et de décomposer les ensembles de produits en fonction de la présence ou de l'absence d'un littéral :

- ✓ les feuilles 0 et 1 codent respectivement l'ensemble vide de produits et l'ensemble ne contenant que le produit vide ;
- ✓ un nœud interne N étiqueté par le littéral q et pointant vers les nœuds N_1 (fils alors) et N_0 (fils sinon) code l'ensemble de produits $S=\{\{q\} \cup \pi ; \pi \in S_1\} \cup S_0$, où S_1 et S_0 sont respectivement les ensembles de produits codés par N_1 et N_0.

Cette représentation des ensembles de produits est canonique. Elle nécessite de changer la deuxième règle de réduction des BDDs qui devient : les nœuds dont le fils gauche (branche alors) est la feuille 0 sont inutiles.

Les opérations ensemblistes (union, intersection, différence) se programment sur les ZBDD de façon similaire aux opérations logiques sur les BDD. Elles utilisent le même mécanisme de cache. Leur complexité dans le pire des cas est donc le produit des tailles de leurs opérandes.

Grâce au partage des sous-arbres isomorphes, les BDD et les ZBDD rendent possibles le codage et la manipulation de très grosses fonctions (ou de très gros ensembles de produits).

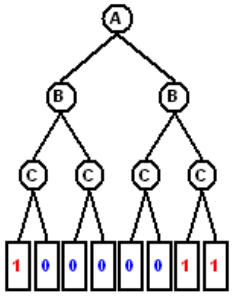
III.3.5 Algorithme

L'algorithme de génération de BDD à partir de l'arbre de Shannon sera présenté à travers l'exemple suivant pour mieux cerner les différentes étapes de construction des BDDs.

Soit la formule logique suivante :

F='A'B'C + AB'C + ABC

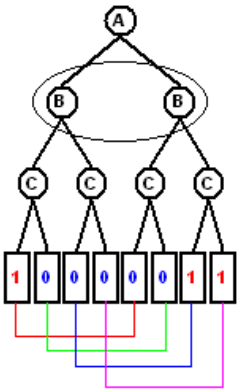
L'arbre de Shannon correspondant est le suivant :



1. Fusion des nœuds non terminaux

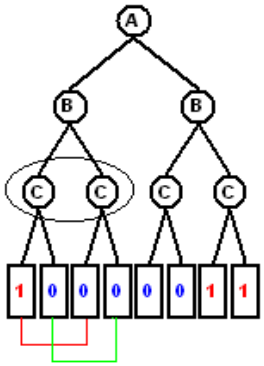
- Niveau 2 :

Les deux nœuds adjacents entourés par une ellipse ne sont pas identiques parce que les valeurs de leurs nœuds terminaux correspondants ne sont pas égales, d'où ils ne seront pas fusionnés dans le BDD.

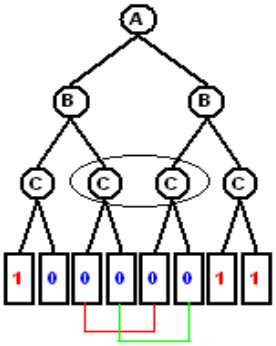


- Niveau 3 :

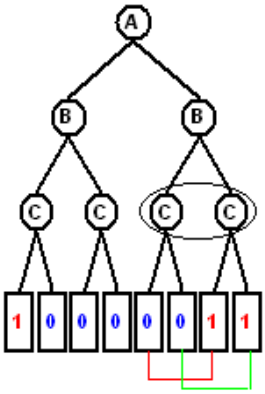
Les deux nœuds adjacents entourés par une ellipse ne sont pas identiques parce que les valeurs de leurs nœuds terminaux correspondants ne sont pas égales, ainsi ils ne seront pas fusionnés dans le BDD.



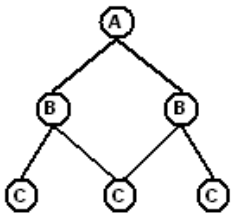
Les deux nœuds adjacents entourés par une ellipse sont identiques parce que les valeurs de leurs nœuds terminaux correspondants sont égales, ainsi ils seront fusionnée dans le BDD.



Les deux nœuds adjacents entourés par une ellipse ne sont pas identiques parce que les valeurs de leurs nœuds terminaux correspondants ne sont pas égales, d'où ils ne seront pas fusionnés dans le BDD.

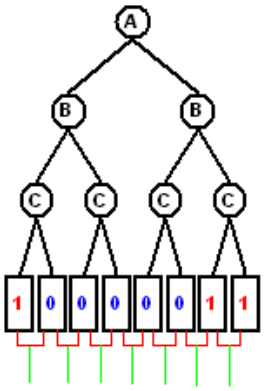


Après que la première partie soit accomplie le diagramme de décision binaire sera comme ceci :

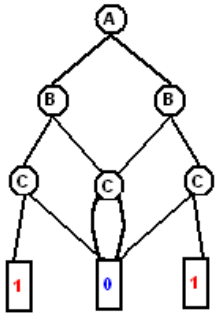


2. Fusion des nœuds terminaux

Dans la présente partie nous comparons chaque deux nœuds terminaux adjacents dans l'arbre binaire. Si les valeurs de ces nœuds terminaux sont égales alors ils sont fusionnés dans le BDD, autrement ils ne seront pas fusionnés.

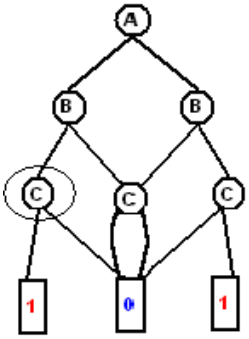


Après que la deuxième partie soit accomplie le diagramme de décision binaire sera comme ceci :

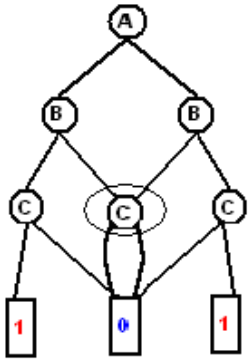


3. Elimination des nœuds non terminaux redondants :
    - Niveau 3 :

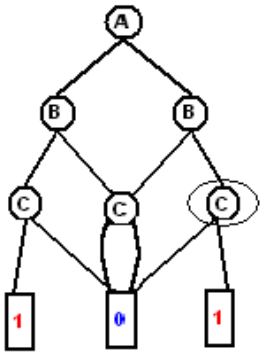
Le nœud non terminal entouré par une ellipse n'est pas un nœud redondant, parce que ses fils gauche et droit sont différents, d'où lui ne sera pas enlevé du BDD.



Le nœud non terminal entouré par une ellipse est un nœud redondant, parce que ses fils gauche et droit sont les mêmes, d'où lui sera enlevé du BDD.

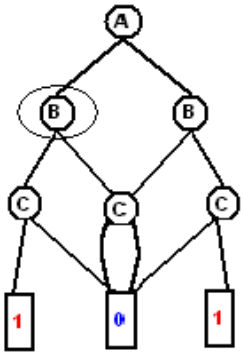


Le nœud non terminal entouré par une ellipse n'est pas un nœud redondant, parce que ses fils gauche et droit sont différents, d'où lui ne sera pas enlevé du BDD.

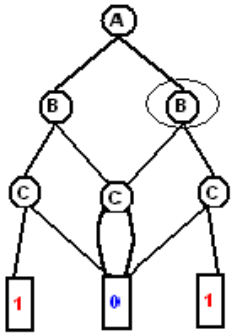


- Niveau 2 :

Le nœud non terminal entouré par une ellipse n'est pas un nœud redondant, parce que ses fils gauche et droit sont différents, d'où lui ne sera pas enlevé du BDD.

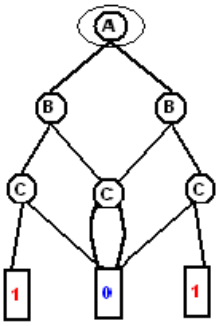


Le nœud non terminal entouré par une ellipse n'est pas un nœud redondant, parce que ses fils gauche et droit sont différents, d'où lui ne sera pas enlevé du BDD.

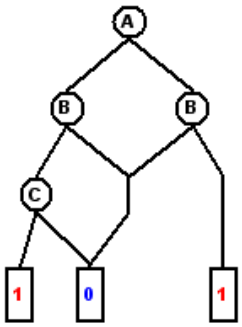


- Niveau 1 :

Le nœud non terminal entouré par une ellipse n'est pas un nœud redondant, parce que ses fils gauche et droit sont différents, d'où lui ne sera pas enlevé du BDD.



Après que la troisième partie soit accomplie le diagramme de décision binaire sera comme ceci : Résultat final



III.4 Conclusion

Dans ce chapitre, nous avions présenté un bref rappel sur l'algèbre de Boole et qui n'est donc en rien original puisqu'il est décrit dans toutes les monographies sur la sûreté de fonctionnement. Ainsi qu'un algorithme de génération de diagramme de décision binaire à partir de l'arbre de Shannon est présenté en partant de la formule logique interprétant l'arbre de défaillance, en passant par l'arbre de Shannon et en arrivant au diagramme de décision binaire correspondant, et ceci en passant par différentes étapes de fusion et de suppression des nœuds redondants dans l'arbre binaire.

Chapitre IV
Analyse qualitative & quantitative de la sûreté de fonctionnement des onduleurs

IV.1 Introduction

Dans ce chapitre, nous commençons par une généralité sur les convertisseurs statiques et la présentation de la chaine de conversion. Puis nous étendons l'étude à la sûreté de fonctionnement d'un onduleur à quatre bras à tolérance aux pannes. Pour cela nous considérons que les défauts dans la chaîne de conversion sont dus uniquement à un défaut d'ouverture ou de court-circuit d'interrupteur dans le convertisseur (ce qui n'est pas loin d'être le cas dans la réalité) [F.Khelifi, 2008]. L'application de l'AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités) et de l'AdD (Arbre de Défaillances) seront présentées par la suite.

IV.2 Généralité sur les convertisseurs statiques

Grâce aux progrès technologiques réalisés ces dernières années dans le domaine de l'électronique de puissance, les convertisseurs statiques voient progressivement leur champ d'applications s'élargir. Certaines de ces nouvelles applications, telles que le filtrage actif et la dépollution de réseaux électriques, ou l'alimentation de machines à courant alternatif pour des applications particulières, sont très exigeantes en terme de performances dynamiques, l'amplification de puissance (audio…). Pour de telles applications, il faudrait un convertisseur idéal, qui génèrerait une tension (ou un courant) de sortie rigoureusement identique à un signal de référence, à un facteur de proportionnalité près, y compris lorsque ce signal varie rapidement. Le transfert d'énergie entre la source et le récepteur serait alors idéalement contrôlé, et il n'y aurait pas d'harmoniques indésirables sur la tension (ou le courant) de sortie du convertisseur. En réalité, les convertisseurs statiques ne peuvent fournir qu'une tension (ou un courant) découpé(e), car « l'électronique de puissance ne peut être qu'une électronique de commutation » [M.Aimé, 2000]. Un grand nombre de procédés industriels s'appuient sur des entraînements électriques à vitesse variable. Dans beaucoup de ces applications, l'élément principal est un onduleur de tension qui alimente un moteur asynchrone. Les onduleurs sont des structures en pont constituées le plus souvent d'interrupteurs électroniques tel que les IGBT ou des transistors de puissance.

IV.3 Etude de cas : Onduleur de tension MLI dans un actionneur électro-hydrostatique (EHA) de surface de vol

Le système présenté ici est un *actionneur électro-hydrostatique* (EHA) de surface de vol. En effet, c'est une charge qui impose des contraintes particulièrement élevées au niveau de l'électronique de puissance : fonctionnement impulsionnel, performances dynamiques élevées, réversibilité en puissance et disponibilité élevée. Un EHA permet de commander le déplacement d'un vérin par un circuit hydraulique local à partir d'une puissance électrique. Classiquement, le réseau de bord triphasé alimente un bus continu à travers un pont de diodes double alternance. Ensuite, un onduleur de tension MLI autopilote une machine synchrone à aimants permanents qui entraîne une pompe hydraulique actionnant le vérin (figure **IV.1a, IV.1b**) [Mavier.J, 2007].

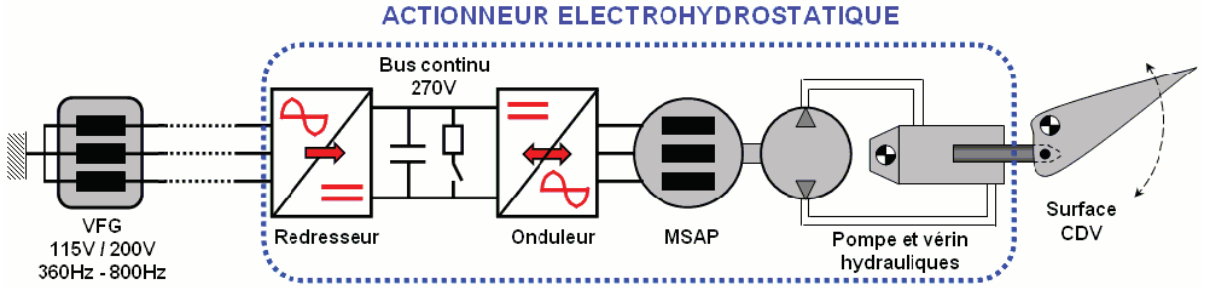


Fig. IV.1a – Synoptique d'un EHA sur réseau avion

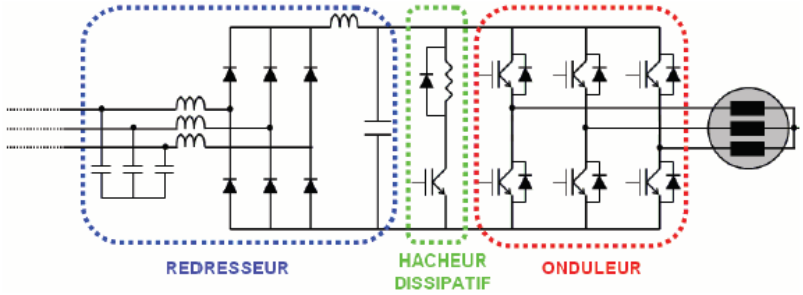


Fig. IV.1b – La chaine de conversion dans un EHA

Chaque bras de l'onduleur est composé d'une cellule de commutation comprenant deux interrupteurs. La Commande MLI permet d'obtenir des courants sinusoïdaux à partir de tension découpée comportant deux niveaux (figure **IV.2a**).

Chaque cellule de commutation est composée deux interrupteurs, chaque interrupteur est composé d'un transistor IGBT en antiparallèle avec une diode (figure **IV.2b**).

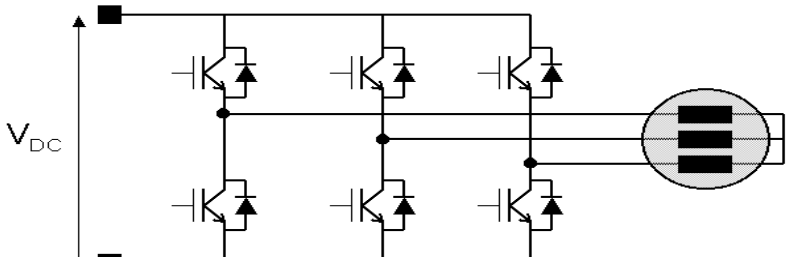


Fig. IV.2a – Structure de l'onduleur dans un EHA

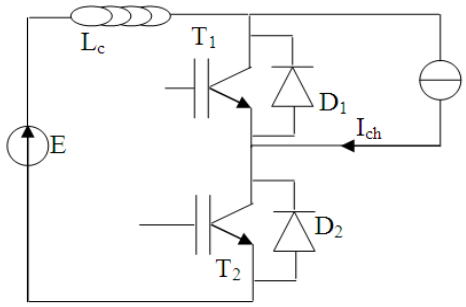


Fig. IV.2b – Structure d'une cellule de commutation

IV.4 Application de la méthode de l'Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticités (AMDEC)

L'AMDEC tient une place importante en sûreté de fonctionnement car c'est une méthode permettant de formaliser un grand nombre de connaissances sur les installations industrielles. La méthode AMDEC [Monchy, 2000] est une méthode qualitative et inductive visant à recenser les défaillances [Zwingelstein, 2009], puis à en estimer les risques (évaluation de la criticité).

L'AMDEC est vue comme un outil rentrant dans une stratégie d'amélioration permanente de la fiabilité et de la disponibilité des composants électroniques [Renson, 2003]. La méthode est basée sur quatre étapes séquentielles: *la décomposition fonctionnelle*, *l'analyse qualitative*, *l'analyse quantitative* et finalement *une proposition d'actions correctrices*.

Un rapport d'analyse ou grille AMDEC résume le bilan de l'étude de l'analyse des défaillances.

La *description fonctionnelle* adoptée est basée sur une décomposition arborescente et hiérarchique de la structure de l'équipement.

L'étape suivante *d'analyse qualitative* consiste à faire une identification des modes de défaillances, de leurs effets et des causes conduisant au dysfonctionnement d'un élément du système.

Ces trois notions sont liées par la relation suivante : Cause → Mode → Effet.

La troisième étape *d'analyse quantitative* permet le calcul de l'indice de criticité ***C***, déduit par le produit des trois indices nominaux ***O***, ***G*** et ***D***, représentant respectivement l'indice de l'***occurrence*** d'une défaillance due à une cause particulière, l'indice de ***gravité*** des effets de la défaillance et l'indice relatif à la possibilité de ***détecter*** la défaillance avant qu'elle ne produise l'effet.

La ***criticité C*** calculée permet de recenser les défaillances dont le niveau de criticité est supérieur à un seuil qui varie en fonction de critères de fiabilité [Monchy, 2000].

Après le calcul de la criticité ***C*** et sa comparaison avec le seuil établi, différentes actions correctrices sont menées qui permettent d'améliorer les valeurs des indices précédents et de définir les indices finaux ***O'***, ***G'*** et ***D'*** sur la même base que l'évaluation précédente. La nouvelle criticité ***C'*** calculée quantifie l'amélioration réalisée [Noureddine.M, 2010].

IV.4.1 Contraintes pouvant survenir dans une cellule de commutation de type onduleur

La cellule de commutation présentée dans la figure **V.1** est composée deux interrupteurs, chaque interrupteur est composé d'un transistor IGBT en antiparallèle avec une diode.

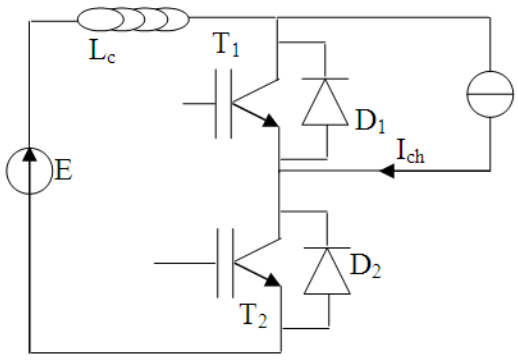


Fig. IV.3 – Cellule de commutation

Les contraintes pouvant survenir dans une cellule de commutation sont les suivantes :

1. Conduction en inverse ;
2. Conséquences suite aux gradients importants de tension ou de courant :

2.1. Commutation à la fermeture imposant un fort gradient de tension ;
2.2. Commutation à l'ouverture imposant un fort gradient de tension ;
2.3. Commutation à l'ouverture imposant un fort gradient de courant ;
2.4. Ouverture rapide sur une charge inductive (fort gradient de courant au blocage) ;
2.5. Commutation à la fermeture imposant un fort gradient de courant.

3. Défaut dans la diode de roue libre ;
4. Mise en avalanche dynamique dans la diode de roue libre ;
5. Défaut de commande d'origine de :
   5.1. Retard des commutations ;
   5.2. Défaut de driver (défaillance d'un composant électronique ou défaut d'alimentation) ;
   5.3. Défaut de commande éloignée.
6. Défaut de court circuit d'origine de :
   6.1. Défaut physique (interne au semi-conducteur) ;
   6.2. Défaut de commande (soit par l'ouverture de l'interrupteur soit par sa mise en court-circuit).
7. Comportement d'un transistor IGBT en court-circuit suite à un défaut de commande.

Le vecteur de test que nous avons choisi pour mener nos essais de fiabilité est une cellule de commutation, de type onduleur de tension. L'étude AMDEC va porter sur cette entité. Avant de présenter cette étude, un rappel sur les terminologies employées est nécessaire.

Le "*défaut*" traduit un fonctionnement anormal, généralement ponctuel et réparable d'un sous-système.

Selon son effet et la gestion qui en est faite, le défaut peut être confiné ou alors se transformer en "*défaillance*" traduisant la perte de fonctionnalité permanente et complète du composant ou du sous-système.

La défaillance est très souvent "*l'effet visible*" ou "*observable*" d'un défaut. On parle ainsi de "*mode de défaillance*".

IV.4.2 La démarche de l'AMDEC

La méthode s'inscrit dans une démarche en six étapes (figure **IV.4**). Comme dans plusieurs démarches, il y a une phase préparatoire qui consiste à une collecte de données pour réaliser l'étude.

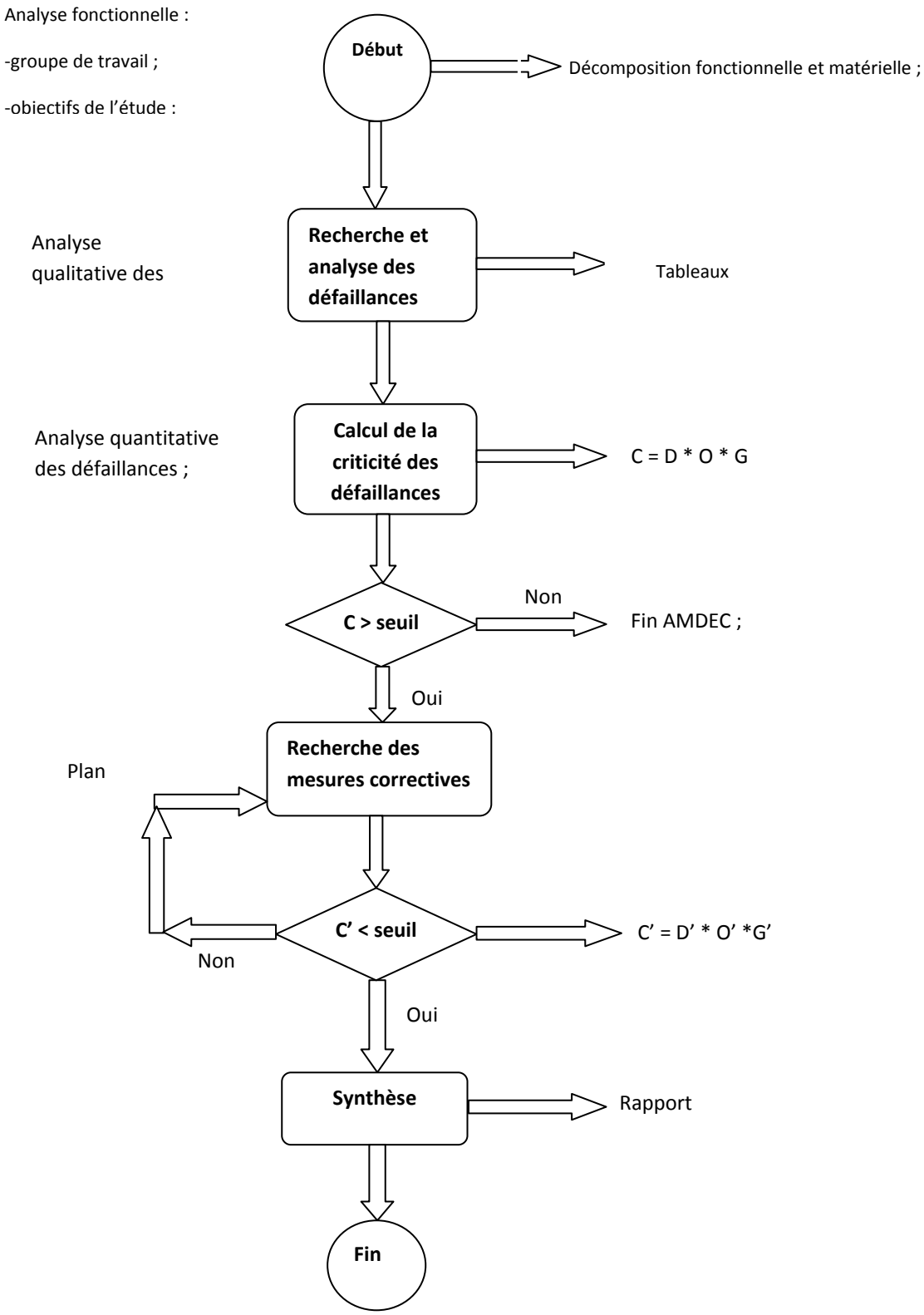


Fig. IV.4 – Synoptique générale de l'AMDEC

IV.4.3 Génération de l'AMDEC pour une cellule de commutation

Le résultat de l'étude AMDEC est présenté dans le tableau **IV.1** récapitulant tous les défauts pouvant apparaître sur cette cellule de commutation et tous les modes de défaillance qui en découlent. A chaque fois, le mode de défaillance est analysé (cause, effet, gravité et criticité).

Eléments	Fonction	Modes de défaillances	Causes de défaillance	Effets sur le système	Evaluation				Actions Préventives		Résultats			
					G	O	D	C	Recommandées	Prises effectivement	G’	O’	D’	C’
Transistor IGBT	interrupteur	Défaut d’ouverture	- échauffement - défaut de commande - vieillissement. - conduction en inverse. - fort dv/dt	Arrêt de l’onduleur										
		Défaut de fermeture	- défaut de commande - vieillissement. - échauffement. - dv/dt ou di/dt import	Arrêt de l’onduleur										
		Court circuit Physique	-surintensité -surtension - vieillissement	Arrêt de l’onduleur(Destruction de composant)										
		Claquage par avalanche	- surtension - claquage de la grille	Arrêt de l’onduleur(Destruction de composant)										
		Retournement statique ou latch up	- augmentation rapide de densité de	Arrêt de l’onduleur (Destruction de composant)										

			courant											
		Retournement et avalanche dynamique	- présence d'un fort courant et d'une forte tension simultanée dans la couche vitale du composant	Arrêt de l'onduleur (Destruction de composant)										
Diode de roue libre	Interrupteur	Court-circuit	- surtensions inverses	Arrêt de l'onduleur										
		Avalanche dynamique	Fort courant, forte tension, haute température et di/dt important sont réunies	Arrêt de l'onduleur										

Tab. IV.1: AMDEC d'une cellule de commutation

IV.5 Application de la méthode de l'Arbre de Défaillance (AdD)

La démarche consiste à s'appuyer sur la connaissance des éléments constitutifs du système étudié pour identifier tous les scénarios conduisant à l'événement redouté. La figure **IV.5** et tableau **IV.2** représentent l'arbre de défaillances résultant de l'AMDEC de la cellule de commutation dans un onduleur à MLI.

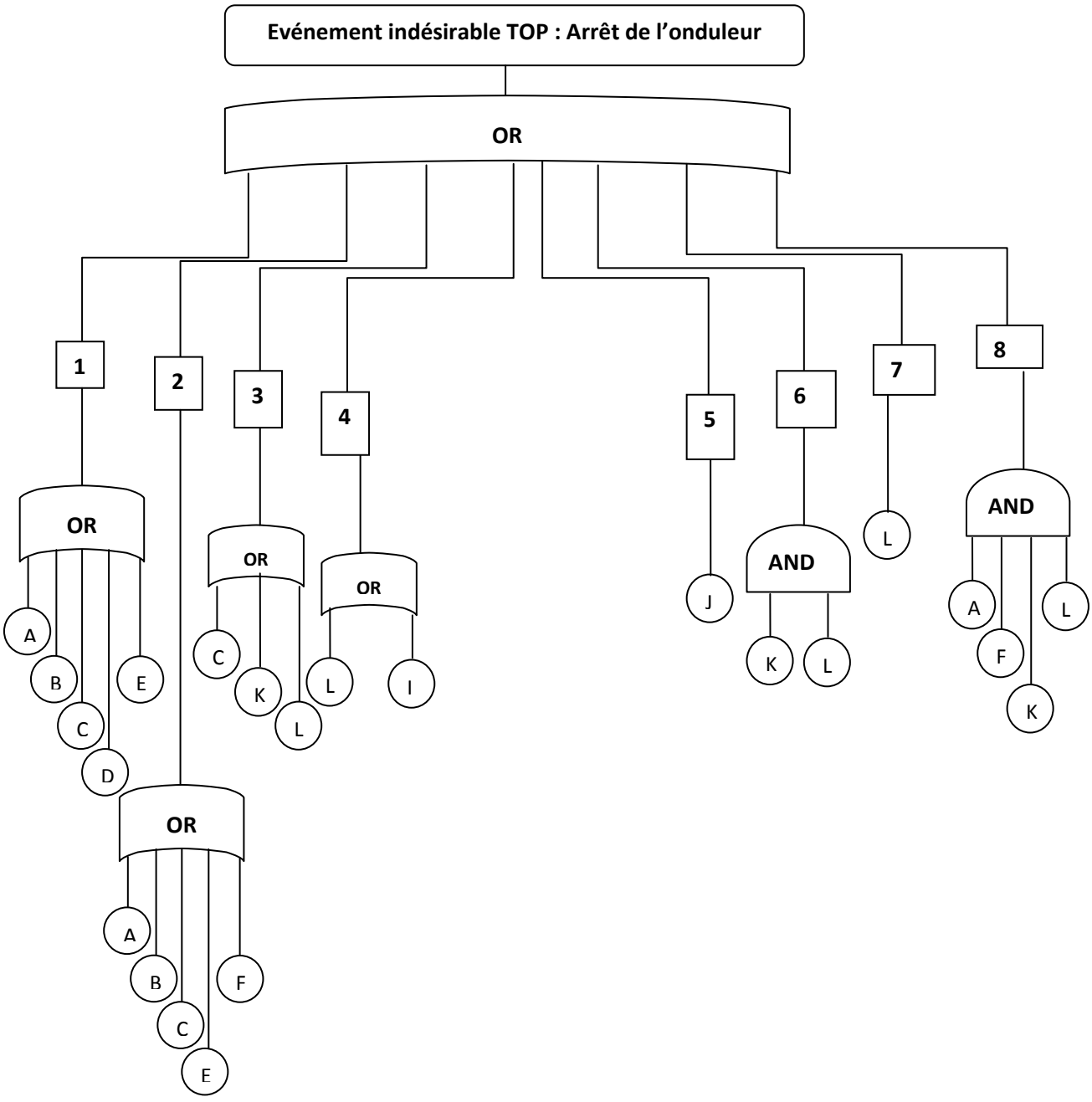


Fig. IV.5 – Arbre de défaillance d'une cellule de commutation dans un onduleur à MLI

Modes	**Causes**
1 : Défaut d'ouverture dans un transistor IGBT	**A :** Echauffement **B :** Défaut de commande **C :** Vieillissement **D :** Conduction en inverse **E :** Fort dv / dt
2 : Défaut de fermeture dans un transistor IGBT	**B :** Défaut de commande **C :** Vieillissement **A :** Echauffement **E :** Fort dv / dt **F :** di / dt important
3 : Court circuit physique dans un transistor IGBT	**K :** Surintensité **L :** Surtension **C :** Vieillissement
4 : Claquage par avalanche dans un transistor IGBT	**L :** Surtension **K :** Surintensité **C :** Vieillissement
5 : Latch Up dans un transistor IGBT	**J :** Augmentation de la densité de courant
6 : Retournement et avalanche dynamique dans un transistor IGBT	**K :** Fort courant **L :** Forte tension
7 : Court circuit dans la diode	**L :** Surtension
8 : Avalanche dynamique dans la diode	**K :** Fort courant **L :** Forte tension **A :** Echauffement **F :** di / dt important

Tab. IV.2: Tableau de l'arbre de défaillance de l'onduleur

IV.6 Application de la méthode de l'Arbre de Décision Binaire (BDD)

L'arbre de défaillance peut être interprété comme une représentation des relations booléennes entre les différents événements qui entraînent l'événement indésirable ; d'où l'importance de l'algèbre de Boole.

La formule logique interprétant l'arbre de défaillance représenté dans la figure **IV.5** est la suivante :

$Z = (A + B + C + D + E) + (A + B + C + E + F) + (C + K + L) + (L + I) + J + (K . L) + L + (A . F . K . L)$

Après simplification, la formule sera comme suit :

$Z = A + B + C + D + E + F + I + J + K + L + (K . L)$

Avec :

Z : est l'événement indésirable provoquant l'arrêt complet de l'onduleur.

A, B, C, D, E, F, I, J, K et L sont les événements élémentaires provoquant l'arrêt de l'onduleur (tableau **IV.2**).

L'arbre binaire complet se trouve dans l'annexe B.

En appliquant l'algorithme de génération de coupes minimales ainsi que les théorèmes de décomposition, l'arbre binaire réduit sera comme suit (figure **IV.6**):

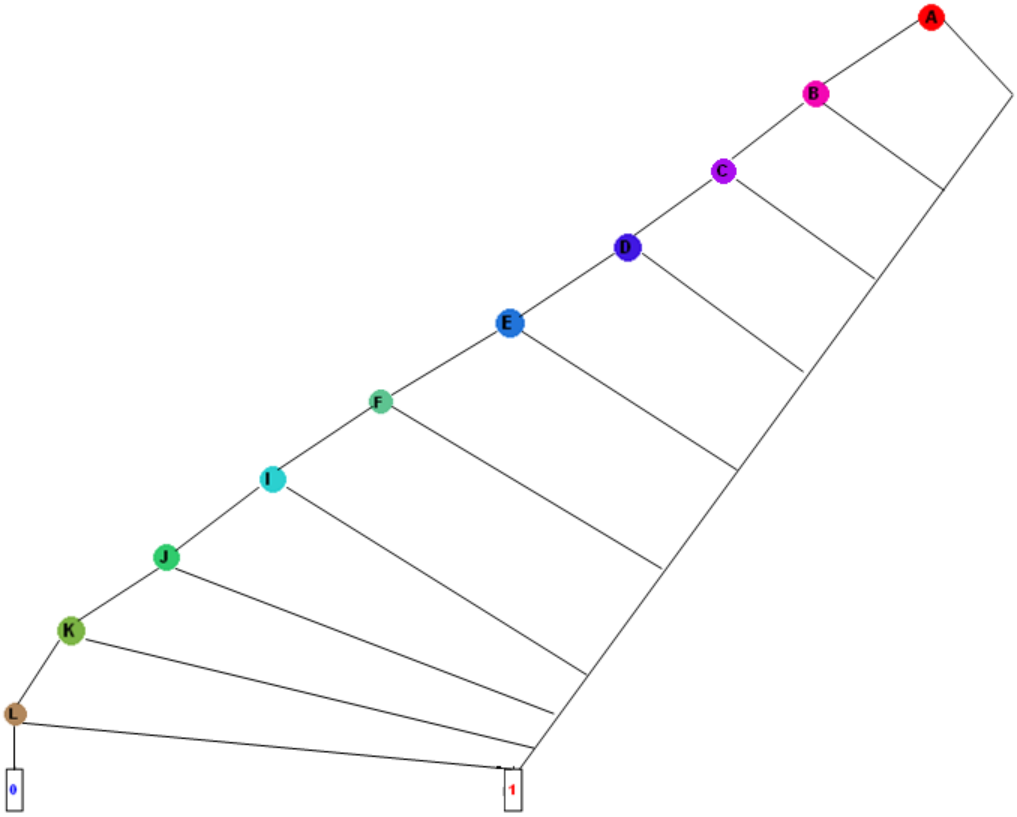


Fig. IV.6 – Arbre binaire réduit

IV.8 Conclusion

Ce chapitre nous a permis d'étudier la sûreté de fonctionnement de l'onduleur de tension. Des retours d'expérience prouvent que la majorité des défaillances sont dues à un défaut d'interrupteur.

L'Analyse des Modes de Défaillances, de leurs Effets et de leurs Criticités (AMDEC) est considérée comme un outil rentrant dans une stratégie d'amélioration permanente de la fiabilité et de la disponibilité des composants électroniques. L'Arbre de Défaillances (AdD) nous a permis une meilleure observation de l'enchaînement des défauts.

Ainsi, le passage de l'arbre de défaillance à l'arbre de décision binaire nous a également permis de mieux représenter les différentes relations booléennes entre les différents événements qui entraînent l'événement indésirable, et ceci en démarrant de la formule logique interprétant l'arbre de défaillance correspondant; d'où l'importance de l'algèbre de Boole.

L'analyse qualitative permet de déterminer les faiblesses du système dont le but est de proposer des modifications afin d'améliorer la fiabilité du système. Or que l'analyse quantitative nous a permis d'évaluer la criticité via l'AMDEC ainsi que la disponibilité à travers l'arbre de décision binaire.

Chapitre V Conception & réalisation d'une application informatique pour l'analyse de la Sûreté de Fonctionnement des Onduleurs

V.1 Introduction

Dans ce chapitre nous discuterons toutes les fonctions employées pour convertir des fonctions booléennes sous format texte en une structure de données adéquate qui est plus utile et plus facile pour le traitement. Puis nous expliquerons les fonctions et les structures de données employées pour stocker ces différentes fonctions.

Par la suite, les résultats obtenus lors de l'exécution de ces différentes fonctions de minimisation et d'optimisation des arbres binaires seront discutées. Ce qui suit montrera les avantages d'utilisation des BDDs, nous nous sommes focalisés sur ce principal point : *le temps* écoulé pour la génération des minitermes, ainsi que *l'exactitude* du fichier réponse surtout lorsqu'il s'agit du nombre de variables et de la minimisation du nombre des minitermes (coupes minimales).

Mais nous commençons d'abord par discuter les différentes fonctions utilisées dans l'élaboration de fiche AMDEC pour une analyse qualitative de sûreté de fonctionnement.

V.2 Avantages des BDDs

L'optimisation logique a été basée sur deux méthodes : méthodes algébriques et factorisation booléenne. Les méthodes algébriques ont réussi avant la factorisation booléenne et ont gagné un résultat presque optimal.

La factorisation booléenne pourrait fournir de meilleurs résultats, mais en raison de sa complexité informatique élevée, elle n'a pas concurrencé des méthodes algébriques.

Cet échec de la factorisation booléenne était dû à la structure de données inadéquate employée pour représenter des fonctions booléennes.

Les nouvelles techniques pour la décomposition logique basée sur la représentation de BDD sont apparues. Ces techniques sont effectuées par une décomposition itérative et la manipulation de BDD. Il a été prouvé que ces techniques sont très efficaces pour les deux fonctions intensives ET/OU et de XOR.

En conséquence pour tous les avantages mentionnés avant, la décomposition basée sur les BDDs a formé une alternative à l'optimisation existante de logique.

V.3 Objectifs

L'objectif principal de notre projet est de concevoir et réaliser une application qui permet de se servir des BDDs pour manipuler les fonctions logiques afin d'optimiser la sûreté de fonctionnement des systèmes électroniques non réparables.

Autres objectifs :

- Optimisation et représentation des fonctions logiques dans leurs formes minimales (minitermes) puis les représenter sous forme de BDDs ;
- Implémentation de quelques opérations sur les BDDs telles que ET et OU ;
- Génération d'arbres binaires complets et de BDDs.

V.4 Méthodologie de conception

V.4.1 Modules et outils du travail

Notre application comporte deux principaux modules:

- AMDECSys : qui permet de créer une fiche AMDEC pour une analyse qualitative de sûreté de fonctionnement des différents composants électroniques ;
- BDD Generator : qui permet de générer l'arbre binaire complet d'une fonction logique ainsi que son BDD.

Afin d'implémenter cette application, nous avons besoin de différents outils de travail matériel et logiciel :

- Le noyau ; ce qui inclut tous les algorithmes de base du programme tels que la minimisation de la fonction d'entrée, création de l'arbre binaire complet et création de BDD. Pour cela nous avons utilisé Microsoft Visual C++ 6 ;
- L'interface graphique (GUI : Graphical User Interface) ; ce qui inclut les fonctions employées pour montrer les sorties du noyau. Pour cela nous avons utilisé Microsoft Visual Basic 6 ;
- La base de données ; pour stocker les informations qui se trouvent dans les fiches AMDEC. Pour cela, nous avons opté pour un système de gestion de bases de données SGBD Microsoft SQL Server 2000 qui est assez robuste ;
- Un calculateur HP Compaq 610 doté :
  - ✓ D'un système d'exploitation Microsoft Windows XP Professional Ed, SP 3 ;

- ✓ D'un processeur d'Intel Core 2 duo T5870 d'une fréquence de 2 Ghz ;
- ✓ D'une mémoire de travail d'une capacité de 2 Go d'une fréquence de 2 Ghz.

V.4.2 Interfaces et fonctions

V.4.2.1 Module AMDECSys

A travers ce module nous avons la possibilité d'identifier le système à étudier, composé de différents sous systèmes ; chaque sous système comporte plusieurs unités ; ainsi, chaque unité est composée d'un ou plusieurs composants électroniques. Nous avons la possibilité de codifier chaque composant (entité « Code Composant ») et de définir sa fonction principale (entité « Fonction Composant ») (figure **V.1a** et **V.1b**).

Ainsi, les modes et causes de défaillance seront générés à partir d'une liste pré-alimentée comme elle se peut être enrichie en ajoutant d'autres modes et causes récemment constatés.

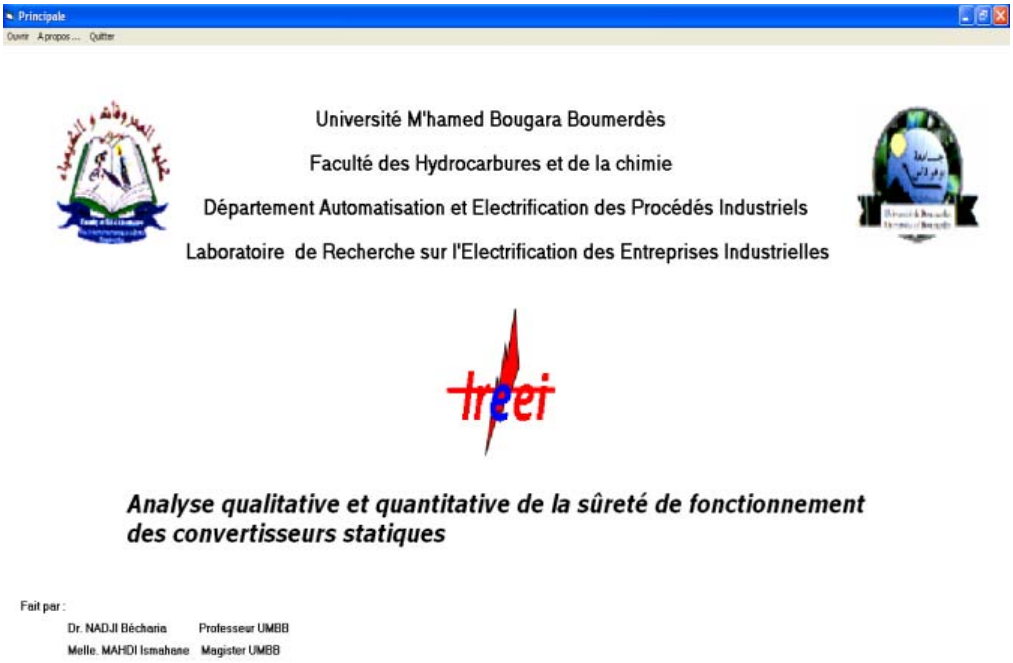


Fig. V.1a – Module AMDECSys.

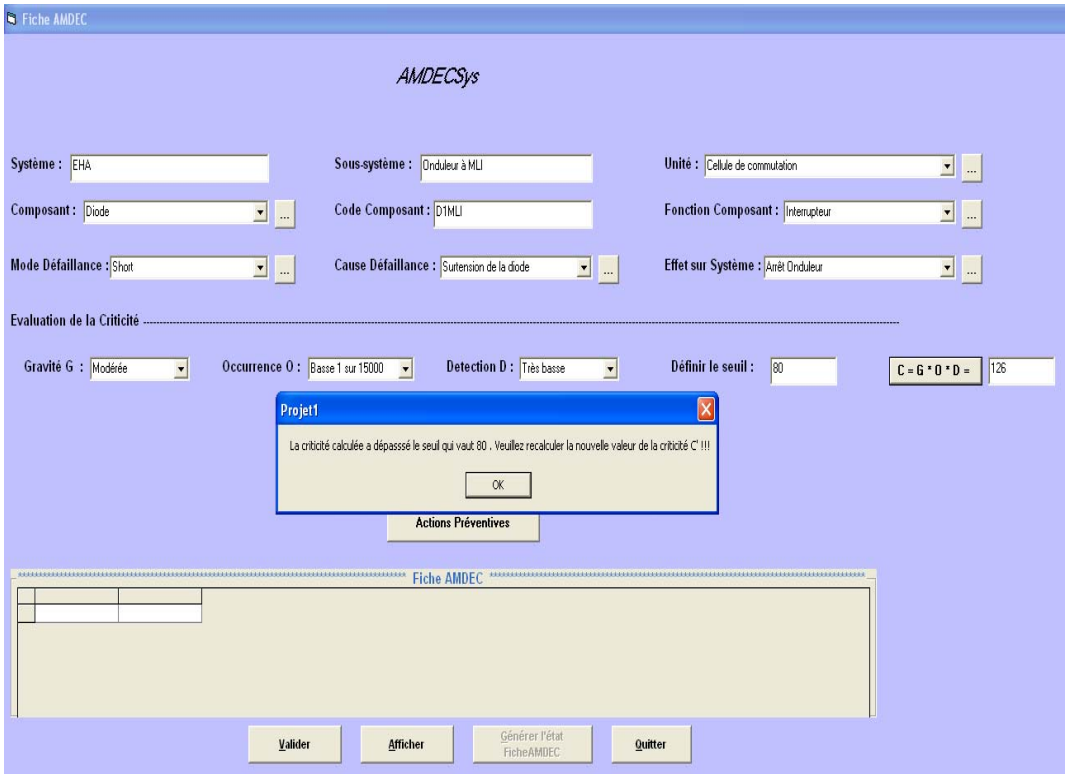


Fig. V.1b – Module AMDECSys.

1. Quelques fonctions ;

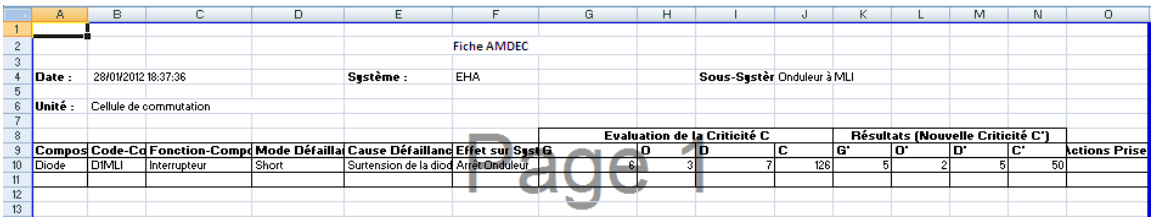
1.1. Bouton « Valider »

```
Private Sub Command3_Click()

If ((Text1.Text = "") Or (Text2.Text = "") Or (Text3.Text = "") Or (Combo1.Text = "") Or (Combo2.Text = "") Or (Combo3.Text = "") Or (Combo4.Text = "") Or (Combo5.Text = "") Or (Combo7.Text = "")) Then
MsgBox "Veuillez saisir tout les champs !!!"
Exit Sub
End If

If (rs1.State = 1) Then
   rs1.close
End If


Requete1 = " "
Requete1 = Requete1 & "insert into Composants  values ('" & Combo2.Text & "','" & Text3.Text & "','" & Combo7.Text & "','" & Combo3.Text & "','" & Combo4.Text & " ','" & Combo5.Text & "','" & Text4.Text & "')"

rs1.open Requete1, conn

End Sub
```

1.2. Bouton « Afficher »

```
Private Sub Command4_Click()

If (rs2.State = 1) Then
        rs2.close
End If

Requete2 = " "
Requete2 = Requete2 & "select * from Composants"

rs2.open Requete2, conn

Set DataGrid1.DataSource = rs2

For i = 0 To 6
       DataGrid1.Columns(i).Locked = True
Next

Command12.Enabled = True

End Sub
```

Après validation des informations saisies dans l'interface de la grille AMDEC, un état ou rapport AMDEC se génère sous format Excel (figure **V.2**). Toute fois, il est possible de sauvegarder tous les états AMDEC (que ce soit automatiquement ou intentionnellement).



					Fiche AMDEC									
Date :	28/01/2012 18:37:36			**Système :**	EHA			**Sous-Systèr**	Onduleur à MLI					
Unité :	Cellule de commutation													
						Evaluation de la Criticité C				**Résultats (Nouvelle Criticité C')**				
Compos	**Code-Co**	**Fonction-Compo**	**Mode Défailla**	**Cause Défaillanc**	**Effet sur Syst**	**G**	**D**	**D**	**C**	**G'**	**O'**	**D'**	**C'**	**ctions Prise**
Diode	D1MLI	Interrupteur	Short	Surtension de la diod	Arrêt Onduleur	8	3	7	128	5	2	5	50	

Page 1

Fig. V.2 – Aperçu de l'état AMDEC.

1.3. Bouton « Générer l'état Fiche AMDEC»

```
Private Sub Command12_Click()

Call PrintDoc("FicheAMDEC")
Set DataGrid1.DataSource = Nothing

End Sub


Public Sub PrintDoc(MySheetName As String)

   Set MyXls = New Excel.Application
   MyXls.Workbooks.Add
   MyXls.Caption = "Fiche_AMDEC"
   MyXls.Sheets.Add
```

```
Set MySheet = MyXls.ActiveSheet
MySheet.Name = MySheetName
MyXls.Vsible = True

If (rs2.RecordCount > 0) Then
   rs2.MoveFirst
x = rs2.RecordCount
y = rs2.Fields.Count

Set MyRange = MySheet.Range(MySheet.Cells(1, 1), MySheet.Cells(x + 1, y))
MyRange.Borders.LineStyle = xlContinuous
MyRange.Font.FontStyle = "italic"
MyRange.Font.Size = 10
MySheet.PageSetup.Orientation = xlLandscape
MyRange.Rows(1).Font.Bold = True
MyRange.Rows(1).Borders.LineStyle = xlDouble
MyXls.Vsible = True

'-------Entête
For i = 1 To y
MyRange(1, i) = rs2.Fields.Item(i - 1).Name
Next

'-------Données
For i = 2 To x + 1
For j = 1 To y
MyRange(i, j) = rs2.Fields(j - 1).Value
Next

rs2.MoveNext
Next

'-----Format des montants
'MyXls.ActiveSheet.Columns("AA:AF").NumberFormat = "#,##0.000"
'MyXls.ActiveSheet.Columns("A:A").NumberFormat = "0;[Red]0"
MySheet.Columns.AutoFit
End If

End Sub
```

1.4. Fonction de chargement du module AMDECSys

```
Private Sub Form_Load()

Form1.WindowState = vbMaximized
Call at_start.Show(1, Form1)

Set conn = New ADODB.Connection
conn.ConnectionString = "ProVder=SQLOLEDB.1;Persist Security Info=True;User ID=sa;Password=saJesse;Initial Catalog=AMDECNew;Data Source=(local)"
conn.CommandTimeout = 300
conn.open

Set rs1 = New ADODB.Recordset
rs1.CursorType = adOpenStatic
rs1.CursorLocation = adUseClient
rs1.LockType = adLockOptimistic
```

```
Set rs2 = New ADODB.Recordset
rs2.CursorType = adOpenStatic
rs2.CursorLocation = adUseClient
rs2.LockType = adLockOptimistic

Set MonRS = New ADODB.Recordset
MonRS.CursorType = adOpenStatic
MonRS.CursorLocation = adUseClient
MonRS.LockType = adLockOptimistic

Set AntRS = New ADODB.Recordset
AntRS.CursorType = adOpenStatic
AntRS.CursorLocation = adUseClient
AntRS.LockType = adLockOptimistic

Set EntRS = New ADODB.Recordset
EntRS.CursorType = adOpenStatic
EntRS.CursorLocation = adUseClient
EntRS.LockType = adLockOptimistic

Command12.Enabled = False

End Sub
```

2. Structure de données

```
Public conn As ADODB.Connection
Dim cmd As ADODB.Command
Public rs1, rs2, MonRS, AntRS, EntRS As ADODB.Recordset
Public Requete1, Requete2, Req1, Req2 As String
Public MyXls As Excel.Application
Public MyRange As Range
Public MySheet As Excel.Worksheet
Public i As Integer
```

V.4.2.2 Module BDD Generator

Ce module nous permet de générer l'arbre binaire complet (arbre de Shannon) ainsi que son BDD d'une formule logique interprétant l'arbre de défaillance du système (figure **V.3**).

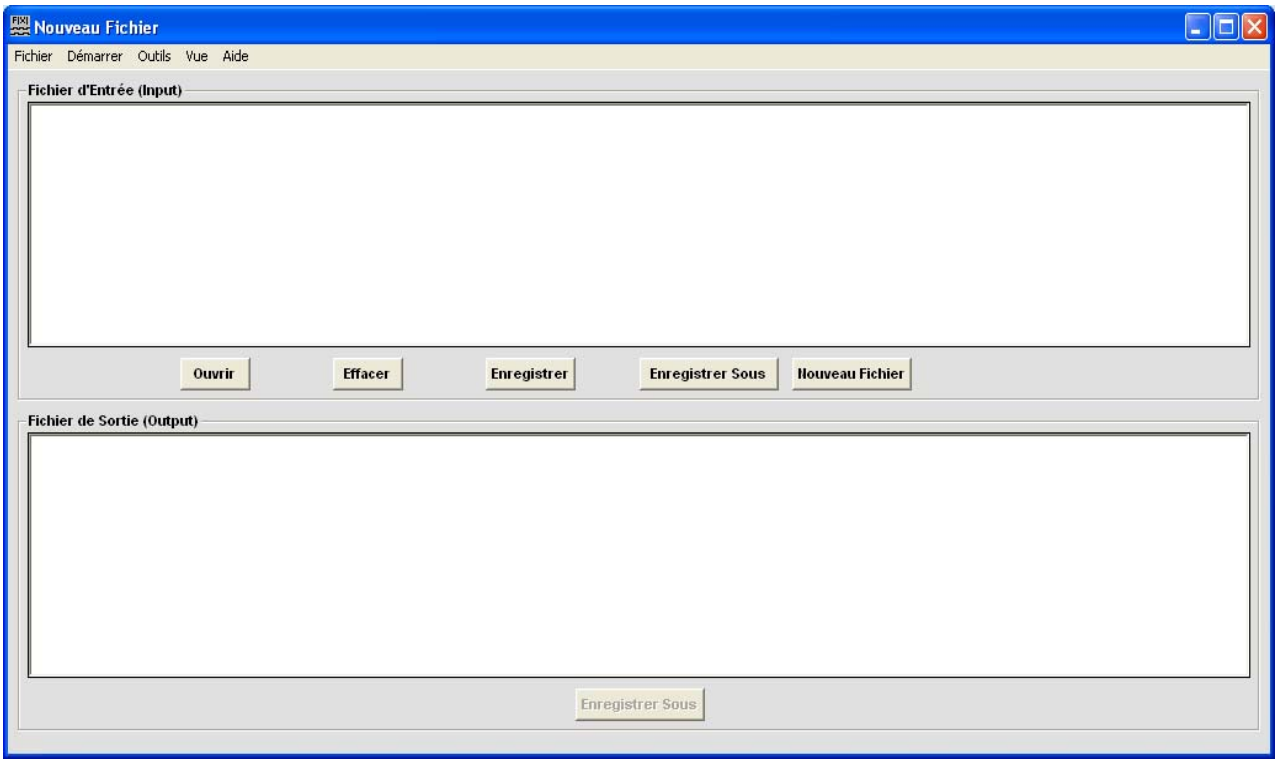


Fig. V.3 – Module BDD Generator.

1. Fonctions et structures de données ;

   AdD = Portes logiques + Evénements

   Evénements = Evénement top + Evénements bas

   Evénement top = Output = Evénement indésirable

   Evénements bas = Inputs = Entrées des différentes portes logiques

   Portes logiques = AND, OR, NOT, XOR, …

   Nous avons opté pour une classe orientée objet afin de générer le code de l'arbre binaire :

```
class binarytree{

friend short int reorder_linked_list(char order_of_variable[],binarytree &tree);

public:

binarytree(indices_node *temp_first,indices_node *temp_last,short int temp_max_number_of_variable);
    ~binarytree();
    void buildtree();
    void filltree();
    unsigned int findvalue(char []);
    void findvalue(char [],char [],binarytree_node *,short int);
    void displaytree();
    binarytree_node *get_root();
    binarytree_node *get_bdd_root();
    indices_node *get_first();
    indices_node *get_reorder_first();
```

```
        indices_node *get_last();
        indices_node *get_reorder_last();
        short int get_number_of_variable();
        char * order_of_variable();
        short int set_order_of_variable(char order[]);

    private:

        void resettree(binarytree_node *start_from);
        void refill_variable_name(unsigned short int level_num,binarytree_node *level_ptr);
        binarytree_node *root;
        indices_node *first;
        indices_node *last;
        indices_node *reorder_first;
        indices_node *reorder_last;
        short int max_number_of_variable;
        short int level;
        char array_order_of_variable[27];
        binarytree_node *createleft(short int );
        binarytree_node *createright(short int );
    };
```

Fonction	**Valeur retournée**	**Description**
binarytree	-	Constructeur.
~binarytree	-	Destructeur.
buildtree	short int	Fonction utilisée pour construire l'arbre binaire et qui sera accessible à partir de la variable *root*.
filltree	Void	Fonction utilisée pour remplir l'arbre.
findpath	unsigned int	Fonction utilisée pour trouver si le chemin indiqué dans le tableau est 1 ou 0.
displaytree	Void	Fonction utilisée pour imprimer tous les chemins et leurs résultats.
get_root	binarytree_node *	Fonction qui retourne un pointeur à l'arbre binaire.
get_bdd_root	binarytree_node *	Fonction qui retourne un pointeur au BDD.
get_first	indices_node *	Fonction qui retourne un pointeur au 1er nœud.
get_last	indices_node *	Fonction qui retourne un pointeur au dernier nœud.
get_reorder_first	indices_node *	Fonction qui retourne un pointeur au 1er nœud dans la liste enregistrée.
get_reorder_last	indices_node *	Fonction qui retourne un pointeur au dernier nœud dans la liste enregistrée.
get_order_of_variable	char *	Fonction qui retourne un pointeur au tableau contenant l'ordre des variables.

set_order_of_variable	short int	Recalculer les indices stockées pour le nouveau ordre de variables.
get_number_of_variable	Short int	Fonction qui retourne le nombre de variables utilisées dans la création de l'arbre binaire.
resettree	Void	Fonction utilisée pour réinitialiser (mise à 0) toutes les valeurs des feuilles dans l'arbre binaire.
refill_variable_name	Void	Recharger les niveaux de l'arbre binaire avec le nouvel ordre.
createleft	binarytree_node *	Fonction utilisée pour créer le sous-arbre gauche pour un nœud déjà créé.
createright	binarytree_node *	Fonction utilisée pour créer le sous-arbre droit pour un nœud déjà créé.

Tab. V.1: Fonctions de la classe

Variable	**Type**	**Description**
root	binarytree_node *	Le pointeur se pointe au 1er nœud de l'arbre binaire.
first	indices_node *	Le pointeur se pointe au 1er nœud de la liste des liaisons par défaut.
last	indices_node *	Le pointeur se pointe au dernier nœud de la liste des liaisons par défaut.
reorder_first	indices_node *	Le pointeur se pointe au 1er nœud de la liste des liaisons réordonnées.
reorder_last	indices_node *	Le pointeur se pointe au dernier nœud de la liste des liaisons réordonnées.
max_number_of_variable	short int	Variable utilisée pour stocker le nombre de variables dans l'arbre binaire.
level	short int	Une variable locale utilisée dans des fonctions récursives.
array_order_of_variable [27]	Char	Tableau de type Char pour stocker l'ordre des variables utilisé pour remplir l'arbre binaire et construire la liste des liaisons réordonnées.

Tab. V.2: Variables de la classe

La classe orientée objet qui permet de générer le code des nœuds l'arbre binaire est la suivante :

```
class binarytree_node{
friend binarytree_node * Logical_Operation(char op,char inv,binarytree_node *array_bdd1,binarytree_node
*array_bdd2,binarytree_node **root);
```

```
friend short int Binary_Dission_Diagram(const short int NV,binarytree_node *root1,unsigned int &no_of_nodes,unsigned int &no_of_edges,char *BDD_output_file,binarytree_node **bdd_root);
friend short int Binary_Dission_Diagram(char *order,const short int NV,binarytree_node *root1,unsigned int &no_of_nodes,unsigned int &no_of_edges,binarytree_node **bdd_root);
    friend short int reorder_linked_list(char order_of_variable[],binarytree &tree);
friend short int Binary_Dission_Diagram(char *order,const short int NV,binarytree_node *root1,unsigned int &no_of_nodes,unsigned int &no_of_edges,binarytree_node **bdd_root,char *BDD_output_file);
friend short int Binary_Dission_Diagram(char *order,const short int NV,binarytree_node *root1,unsigned int &no_of_nodes,unsigned int &no_of_edges,binarytree_node **bdd_root,unsigned int&no_of_ones);
friend short int Binary_Dission_Diagram(const short int NV,binarytree_node *root1,unsigned int &no_of_nodes,unsigned int &no_of_edges,binarytree_node **bdd_root);
    friend class binarytree;

public:
    binarytree_node();
    ~binarytree_node();
binarytree_node( unsigned short int , char , binarytree_node * , binarytree_node * );
    void setvalue( unsigned short int );
    void setname( char );
    binarytree_node *leftptr;
    binarytree_node *rightptr;
    unsigned short int readvalue();
    char readname();

private:
    unsigned short int value;
    char variable;

};
```

Fonction	**Valeur retournée**	**Description**
binarytree_node	-	Constructeur de paramètres Vide.
~ binarytree_node	-	Destructeur.
setvalue	void	Fonction utilisée pour fixer la valeur du nœud soit 1 ou bien 0.
setname	void	Fonction utilisée pour attribuer un nom au nœud.
readvalue	char	Fonction utilisée pour lire la valeur du nœud.
readname	unsigned short int	Fonction utilisée pour lire le nom du nœud.

Tab. V.3: Fonctions de la classe.

Variable	**Type**	**Description**
value	unsigned short int	Variable pour stocker la valeur du nœud (elle n'est pas utilisée si le nœud n'est pas une feuille).
variable	char	Variable pour stocker le nom du nœud (elle prend la valeur 'v' si le nœud est une feuille).

Tab. V.4: Variables de la classe

La classe orientée objet qui permet de générer le code des indices des différents nœuds l'arbre binaire est la suivante :

```
struct indices_node{
    unsigned int calculated_indices ;
    char term[53];
    indices_node *next_node;
    indices_node *pre_node;
    indices_node(unsigned int);
};

struct file_node{
    unsigned short int number_of_variable_in_the_term;
    char term[53];
    file_node *next_file_node;
    file_node(unsigned short int,file_node *);};
```

Fonction	**Valeur retournée**	**Description**
indices_node	-	Constructeur.
file_node	-	Constructeur.

Tab. V.5: Fonctions de la classe.

Variable	**Type**	**Description**
calculated_indices	unsigned int	Variable pour stocker l'indexe du terme.
term[53]	char	Variable pour stocker le terme.
next_node	indices_node *	Pointeur qui pointe au prochain nœud dans la liste des liaisons.
pre_node	indices_node *	Pointeur qui pointe au nœud précédent dans la liste des liaisons.
number_of_variable_in_the_term	unsigned short int	Variable utilisée pour stocker le nombre de variables utilisées dans le terme.

Tab. V.6: Variables de la classe

2. Aperçus et interfaces ;

2.1. Créer un nouveau fichier Input : Fichier => Nouveau Fichier, Fichier => Enregistrer Sous (figure **V.4**);

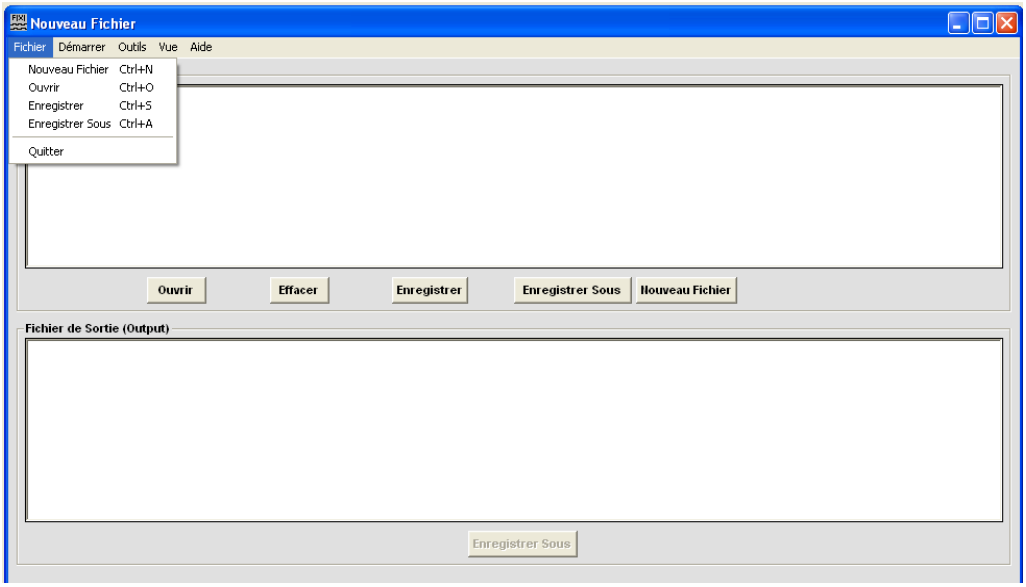


Fig. V.4 – Menu Fichier

2.2. Pour ouvrir un fichier existant : Fichier => Ouvrir ;

2.3. Formats possibles d'Input : peut être la somme des produits (forme disjonctive) ou bien le produit des sommes (forme conjonctive) (figure **V.5**);

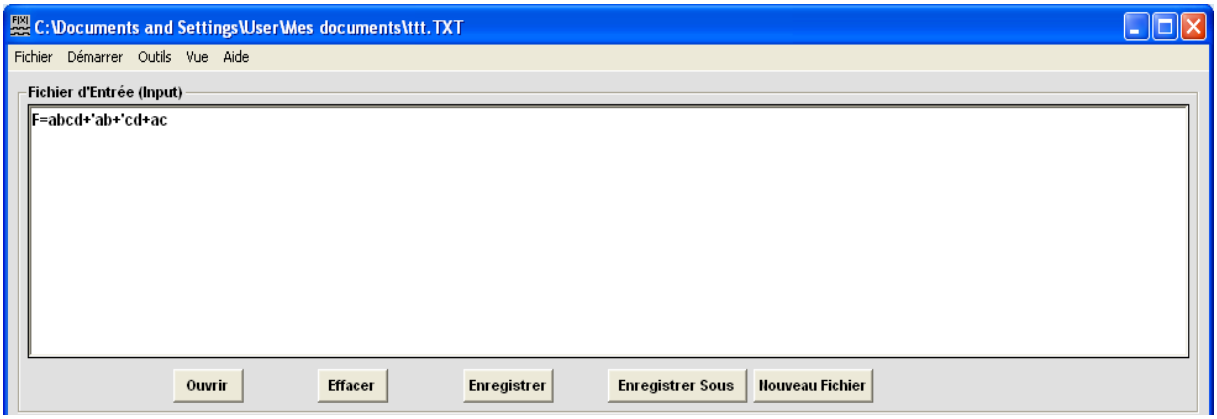


Fig. V.5 – Espace pour saisir la fonction booléenne (Input)

2.4. Pour exécuter le programme : Outils => Démarrer, ou bien F2 (figure **V.6**);

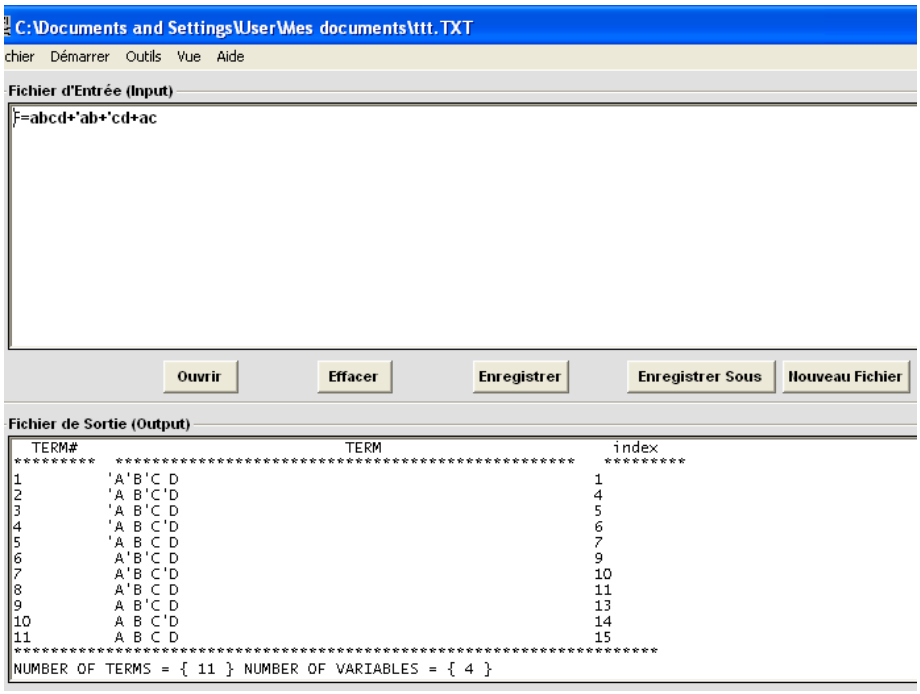


Fig. V.6 – Fenêtre du Output (Résultat)

2.5. Pour générer l'arbre binaire complet : Vue => Arbre Complet (Ctrl + F) (figure **V.7**);

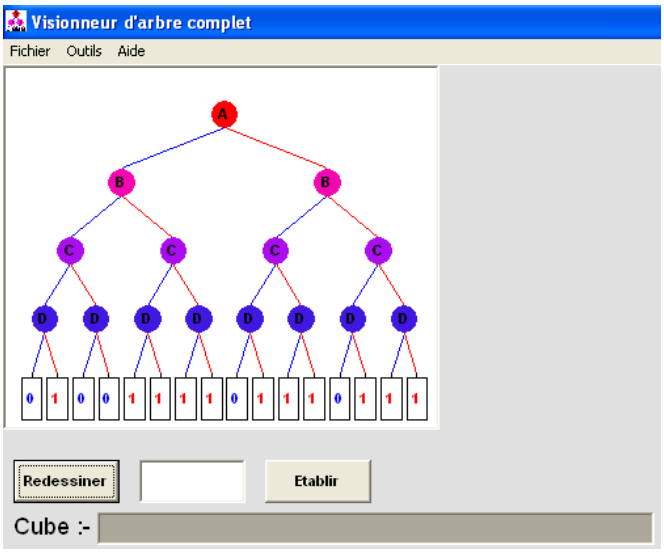


Fig. V.7 – Arbre binaire complet

2.6. Le bouton « Etablir » est utilisé pour trouver un ou plusieurs chemins dans l'arbre, le chemin trouvé apparaît avec une couleur différente (en vert). Pour assigner un chemin, nous choisissons la valeur de l'Input (0 ou bien 1) (figure **V.8**);

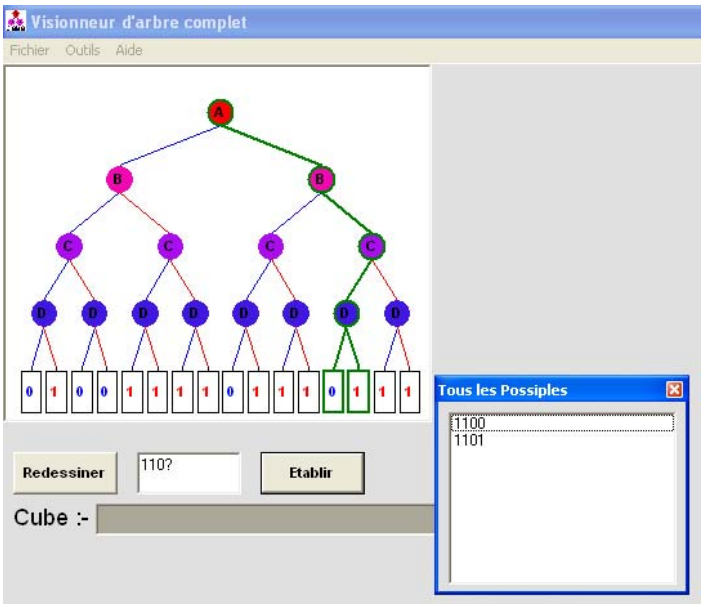


Fig. V.8 – Chemins possibles pour le cas 110 ?

2.7. Pour générer le BDD : Vue => BDD (Ctrl + B) (figure **V.9**);

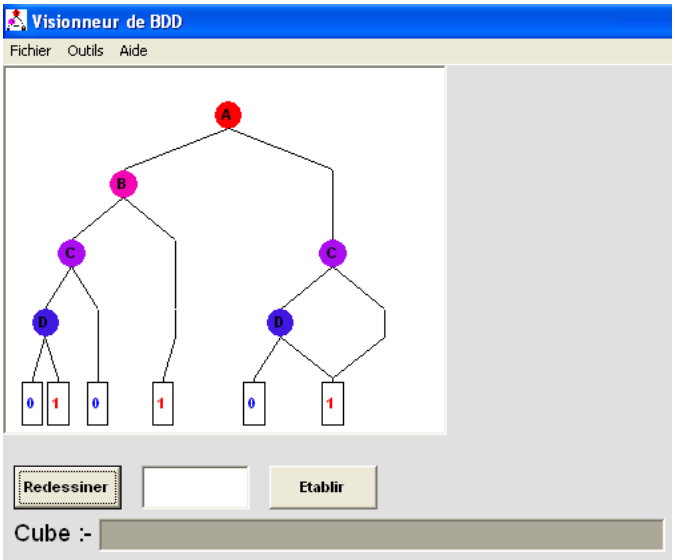


Fig. V.9 – Diagramme de décision binaire (BDD)

V.5 Etude et discussion des résultats

V.5.1 Le temps

Afin de mesurer le temps consommé, nous devons accentuer les fonctions principales qui ont été employées :

- Utiliser les fonctions « *take_and_analyze* » et « *find_term_has_miss_terms* » pour entrer la fonction désirée pour l'optimisation, et l'examiner pour déceler les erreurs, et créer la liste chaînée qui contient les minitermes de début de la fonction ;
- Créer l'instance de la classe « *binarytree* » et utiliser ses fonctions « *buildtree()* » et « *filltree()* » pour la création de l'arbre binaire ;
- Utiliser la fonction « *binary_dission_diagram* » pour générer le diagramme de décision binaire.

Les facteurs qui affectent le temps écoulé dans le traitement sont nombreux; nous mentionnons les plus importants par ordre d'importance :

- Le nombre de variables dans la fonction: plus le nombre (de variables) est grand, plus le temps nécessaire pour le traitement est important (le graphe prend la forme exponentielle);
- Le nombre de minitermes dont laquelle la fonction est composée : plus le nombre (de minitermes) est grand, plus le temps nécessaire pour le traitement est important (le graphe prend la forme linéaire);
- Le matériel utilisé : la fréquence du processeur, la capacité mémoire, …etc.
- Le type de minitermes en Input : quelques minitermes prennent plus de temps que d'autres, ceci dépend de combien de manières différentes ont été utilisés (les minitermes ABC et AB'C prennent moins de temps « différence dans une seule variable ABC et AB'C » que les minitermes 'ABC et AB'C qui prennent plus de temps « différence dans deux variables 'ABC et AB'C ».

Nous étudierons les deux premiers facteurs, puisqu'ils affectent le plus ; le troisième facteur pouvant être récapitulé en décrivant le matériel utilisé ayant les caractéristiques suivantes :

- ✓ un système d'exploitation Microsoft Windows XP Professional Ed, SP 3 ;
- ✓ un processeur d'Intel Core 2 duo T5870 d'une fréquence de 2 Ghz ;
- ✓ une mémoire de travail d'une capacité de 2 Go d'une fréquence de 2 Ghz.

Le tableau **V.7** et la figure **V.10** montrent l'effet du premier facteur dans l'optimisation des BDDs suivant le nombre de variables employées et en fonction du temps :

Nb Vars	3,00	4,00	5,00	6,00	7,00	8,00	9,00	10,00
Temps (sec) X 10	0,03	0,04	0,05	0,12	0,14	0,20	0,97	7,26

Tab. V.7 : Effet du nombre de variables en fonction du temps

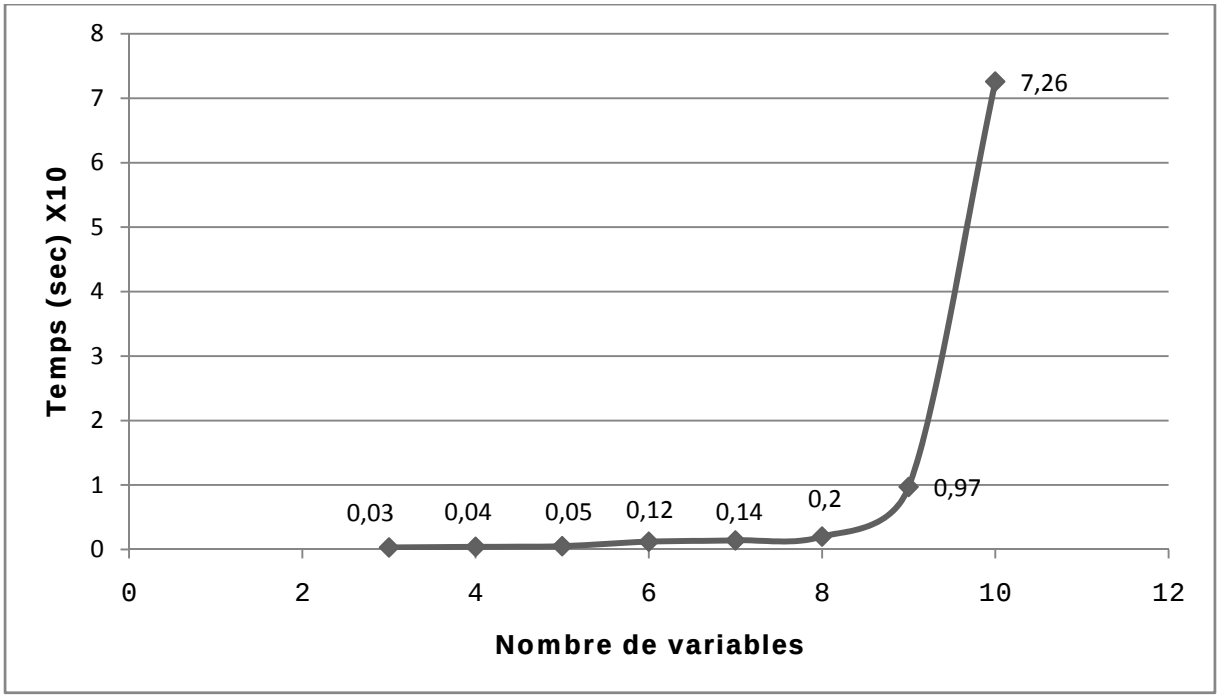


Fig. V.10 – Effet du nombre de variables en fonction du temps

Le tableau **V.8** et la figure **V.11** montrent l'effet du deuxième facteur dans l'optimisation des BDDs suivant le nombre de minitermes dont la fonction est composée :

Nb Vars	3,00	4,00	5,00	6,00	7,00	8,00	9,00	10,00
Nombre de minitermes	5	6	22	34	98	226	482	934

Tab. V.8 : Effet du nombre de minitermes

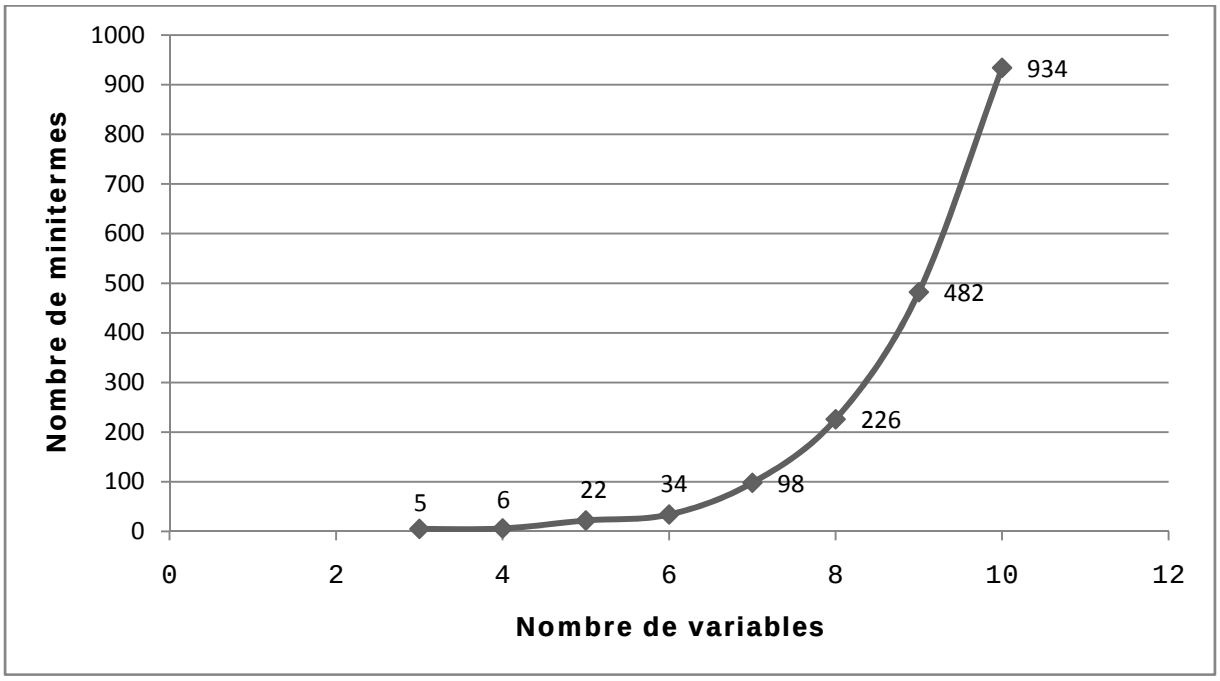


Fig. V.11 – Effet du nombre de minitermes

V.5.2 L'exactitude

Il n'y a aucune règle pour savoir si le résultat de l'optimisation utilisant BDD est optimal ou pas, à moins qu'avoir le résultat de l'optimisation sans employer BDD pour rivaliser avec. En général, plus le nombre de variables employées est important, plus la possibilité d'inexactitude est probable, bien que l'inexactitude peut se produire qu'avec trois variables seulement, ceci dépendra des minitermes de la fonction.

Si les minitermes sont semblables (la différence entre chaque miniterme et l'autre est une variable), l'optimisation sans employer BDD tend à être optimale. Les deux exemples suivants clarifieront plus, le premier exemple est une fonction de trois variables, qui cause l'inexactitude, le deuxième exemple est une fonction de dix variables mais la solution est optimale.

Exemple 1

On considère la fonction F à trois variables :

F = A'B'C + A'BC + ABC + AB'C + 'A'BC + 'ABC

La solution optimale, sans utilisation des BDDS, est (A + C), mais on utilisant le BDD le résultat optimal est (A + 'AC) comme le BDD le montre dans la figure **V.12**:

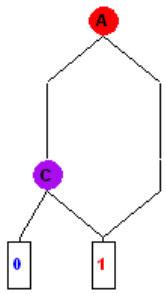


Fig. V.12 – BDD de la fonction F à trois variables

Exemple 2

On considère la fonction K à dix variables :

K = ABCDEFGHIJ + ABCDEFGHI'J + ABCDEFGH'IJ + ABCDEFGH'I'J

La solution optimale, avec et sans utilisation des BDDs, est (ABCDEFGH). Ceci est un exemple d'exactitude utilisant dix variables.

V.5.3 Réarrangement des variables

Le réarrangement diminue le nombre de nœuds employés pour représenter un BDD, ainsi moins de minitermes peuvent se produire, par conséquent, on obtient une solution plus optimale.

Mais le réarrangement consomme le temps, et si le nombre de variables est au-dessus de cinq à six variables, le temps consommé dans le réarrangement peut dépasser la durée de l'optimisation sans utilisation des BDDs, et la solution peut ne pas être optimale, d'où on n'opte pas pour le réarrangement.

La figure **V.13** illustre un exemple de réarrangement de variables, la partie droite de la figure représente un BDD sans réarrangement de variables, alors que la partie gauche de la même figure représente un BDD avec réarrangement de variables (l'ordre est BAC). Dans l'ordre BAC, la fonction optimisée est (BC + A'B + 'A'B'C), alors qu'avec l'ordre ABC, la fonction

optimisée est (ABC + A'B + 'ABC + 'A'B'C), comme nous le constatons, le nombre de minitermes est réduit par un, et le nombre de variables dans chaque minitermes est moins.

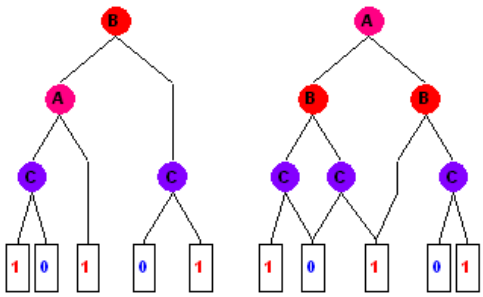


Fig. V.13 – Deux BDDs : à gauche avec réarrangement, à droite sans réarrangement

V.6 Conclusion

A travers l'exécution de l'application que nous avons implémentée, plusieurs fichiers de sortie (Output) peuvent être enregistrés et utilisés pour d'autres applications (fiche AMDEC, fichier de sortie des minitermes, arbre binaire complet et diagramme de décision binaire). Ainsi, la sauvegarde de la base de données en cas de besoin de restauration.

Pour obtenir un BDD optimal, nous avons opté pour le réarrangement des variables, la fonction que le réarrangement mis en application dépend de tous les ordres possibles des variables ainsi le nombre de nœuds produits dans chaque cas; le BDD avec un nombre minimum de nœuds est considéré comme le BDD optimal. Il a été constaté que plus de six variables, le temps de traitement deviendra important.

Les résultats obtenus ci-dessus nous affirment que plusieurs facteurs peuvent affecter le temps écoulé dans le traitement, tels que : le nombre de variables et minitermes dont laquelle la fonction est composée, le type de minitermes en Input ainsi que la configuration matérielle utilisée (fréquence du processeur, capacité mémoire, …etc.).

Il a été constaté aussi qu'aucune règle nous permette de savoir si le résultat de l'optimisation utilisant BDD est optimal ou pas. En général, plus le nombre de variables employées est important, plus la possibilité d'inexactitude est probable.

Conclusion Générale

Conclusion Générale

Notre travail traite la sûreté de fonctionnement et la fiabilité des circuits de l'électronique de puissance en particulier les onduleurs. Il apporte des éléments de caractérisation et des solutions visant à optimiser la fiabilité globale et la sécurité des circuits de puissance.

Après avoir présenté les concepts et les méthodes fondamentales de la sûreté de fonctionnement, ainsi que les principales lois associées à la fiabilité ; nous avions cité les origines de défauts dans un interrupteur et particulièrement dans une cellule de commutation que ce soit au niveau de la diode ou bien au niveau du transistor IGBT.

L'analyse AMDEC nous a permis de mettre en évidence les différents modes de défaillance pouvant apparaître au sein d'une cellule de commutation. Ces modes de défaillance sont les suivants :

- ✓ défaillance de type circuit ouvert (IGBT et diode) ;
- ✓ défaillance de type court-circuit type I (défaut de commande) ;
- ✓ défaillance de type II (défaillance de puce).

Ainsi, le passage de l'arbre de défaillance à l'arbre de décision binaire nous a également permis de mieux représenter les différentes relations booléennes entre les différents événements qui entraînent l'événement indésirable, et ceci en démarrant de la formule logique interprétant l'arbre de défaillance correspondant; d'où l'importance de l'algèbre de Boole.

A travers la plateforme informatique que nous avions conçue, plusieurs fichiers de sortie (Output) peuvent être enregistrés et utilisés par la suite pour d'autres applications (fiche AMDEC, fichier de sortie des minitermes, arbre binaire complet et diagramme de décision binaire). Ainsi que la sauvegarde de la base de données en cas de besoin de restauration. Ceci peut être utile pour le retour de l'expérience REX.

Enfin, les résultats obtenus nous ont permis de mieux cerner les aspects : temps, exactitude et ré-ordonnancement (réarrangement) afin d'optimiser le diagramme de décision binaire pour une meilleure analyse de sûreté de fonctionnement (plus rapide et plus exacte).

Annexe A

Annexe A

Méthodes et outils pour évaluer la fiabilité

A.1 Arbre d'évènement (AE)

L'analyse par Arbre d'Evènements (AE) permet de déterminer les évènements qui découlent suite à une défaillance d'un composant ou d'une partie du système. La méthode de l'AE permet également d'étudier les conséquences de la défaillance des dispositifs de détection, d'alarme,...

La méthode AE comporte quatre étapes :

– définition de l'événement initiateur ;

– identification des fonctions de sécurité prévues ;

– construction de l'arbre ;

– description et exploitation des séquences d'évènements identifiées.

Le principal avantage de l'AE est de permettre l'examen, à partir d'un événement initiateur, de l'enchaînement des évènements pouvant conduire ou non à un accident potentiel.

Cependant, cette méthode peut s'avérer rapidement lourde à mettre en œuvre. En effet, il faut définir avec discernement l'événement initiateur qui fera l'objet de cette analyse.

A.2 Méthode des Combinaisons de Pannes Résumées (MCPR)

La Méthode des Combinaisons de Pannes Résumées (MCPR) [Villemeur, 1988], [Zwingelstein, 1996] permet de déterminer de manière inductive des combinaisons de défaillances qui aboutissent à des événements indésirables, après réalisation de l'AMDE.

L'ensemble des fonctionnements anormaux - ou modes de défaillance - d'un système ou d'un ensemble de systèmes élémentaires est alors obtenu.

Après la réalisation de l'AMDE, la MCPR peut être utilisée pour l'analyse d'un système ou pour l'analyse d'un ensemble de systèmes élémentaires en interaction. La MCPR peut être appliquée simultanément à tous les systèmes élémentaires (thermohydraulique, électrique, mécanique...). Les principales caractéristiques de la méthode sont le regroupement des pannes ayant les mêmes effets et les critères de sélection des combinaisons de pannes.

On distingue quatre principales étapes dans l'application de la méthode :

– décomposition du système élémentaire en composants ;

– élaboration des «pannes résumées internes» ;

– élaboration des «pannes résumées externes» ;

– élaboration des «pannes résumées globales» [Villemeur, 1988].

Le principal avantage de cette méthode est qu'elle se prête à des analyses systématiques, sur les composants du système.

A.3 Méthode Diagramme Causes-Conséquences (MDCC)

La méthode du Diagramme Causes-Conséquences (MDCC) [Villemeur, 1988], [Zwingelstein, 1996] est destinée à l'analyse déductive des causes et des conséquences d'un événement initiateur que l'on redoute de voir survenir dans un système.

L'analyse par Diagramme Causes-Conséquences est une combinaison, d'une part, de l'analyse déductive de l'AdD et, d'autre part, de l'analyse inductive de l'AE. La représentation de l'analyse utilise les symboles de l'arbre des causes, ainsi que des symboles spécifiques à cette méthode. On distingue généralement deux parties dans une MDCC :

– une partie 'causes' représentant les causes d'un ou de plusieurs événements 'sommets' des AdD inacceptables dans certaines conditions ;

– une partie 'conséquences' représentant l'étendue des conséquences envisageables lorsque se réalisent les événements 'sommets', en tenant compte des diverses conditions de fonctionnement ou de défaillance de composants, ou de sous-systèmes [Villemeur, 1988].

Le principal avantage de cette méthode est qu'elle est intéressante pour l'analyse des systèmes où l'ordre dans lequel surviennent les défaillances est important.

A.4 Table de Vérité (TV)

La méthode consiste à recenser toutes les combinaisons d'états (états de fonctionnement, états de panne) des composants, les unes après les autres et à en étudier les effets [Villemeur, 1988].

La méthode de la Table de Vérité nécessite l'étude de toutes les combinaisons d'états de fonctionnement et de panne des composants.

Elle n'est plus applicable dès que le nombre de composants est grand, étant donné le nombre alors très important de combinaisons à considérer.

Après la décomposition du système, il est nécessaire d'examiner les modes de défaillance des composants et les états de panne. Chaque composant est caractérisé par un état de

fonctionnement et par un état de panne. Un vecteur des états est une combinaison d'états des composants, chaque composant étant représenté par son état de fonctionnement ou son état de panne.

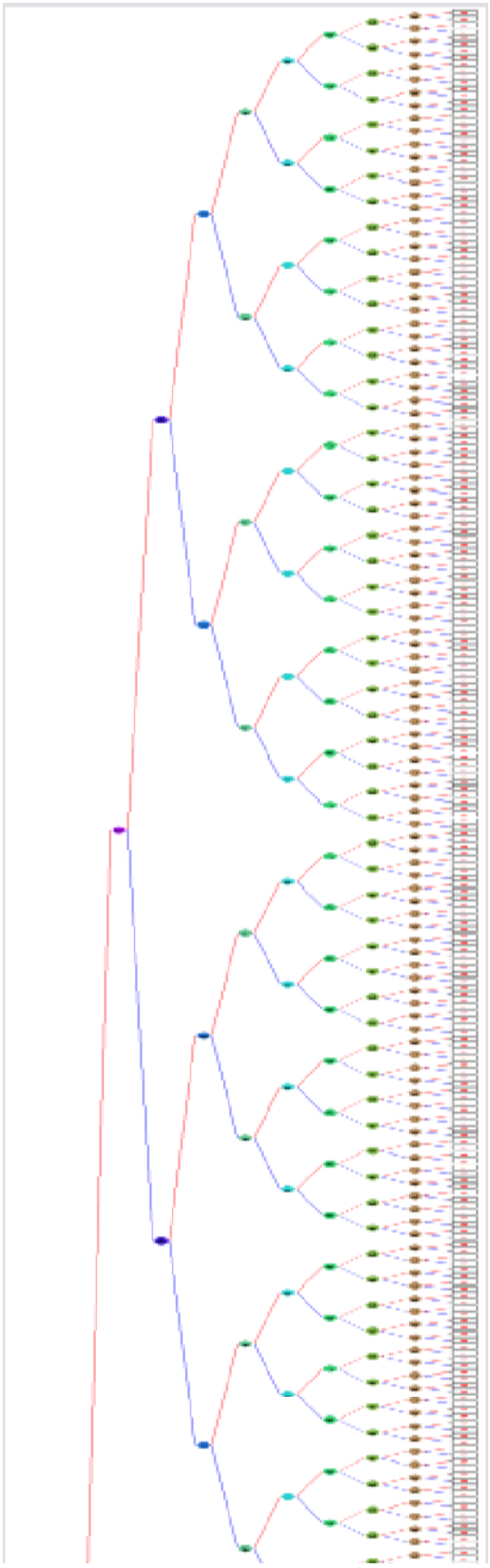
Etablir la table de vérité du système consiste à analyser les effets de tous les vecteurs des états des composants ; l'étude de ces effets permet de recenser tous les fonctionnements anormaux du système.

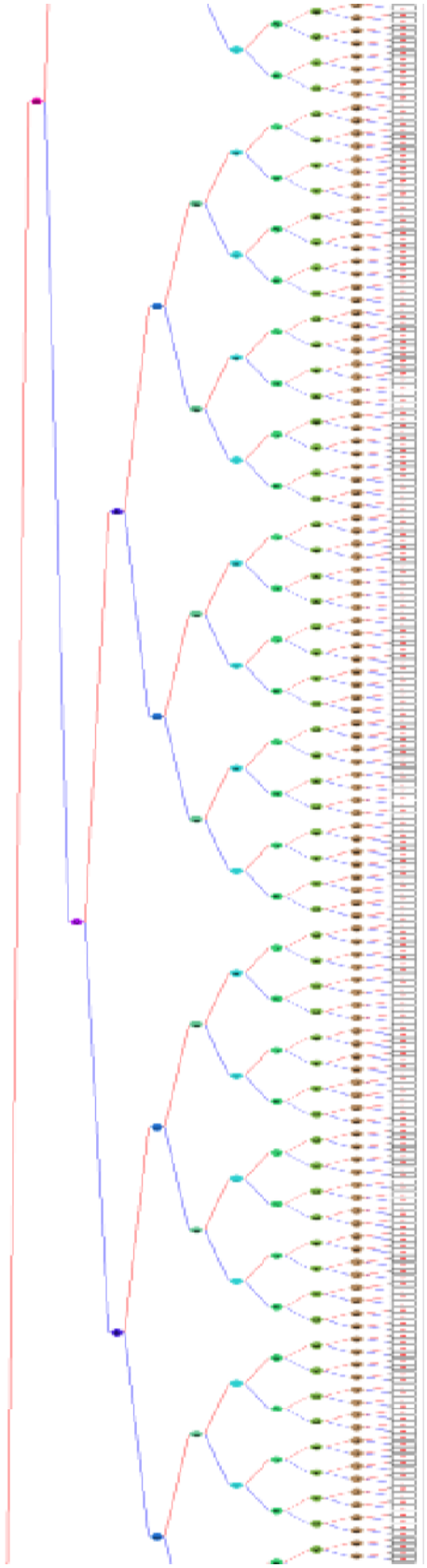
Les résultats sont présentés sous forme d'un tableau où 0 est l'état de fonctionnement et est 1 l'état de panne.

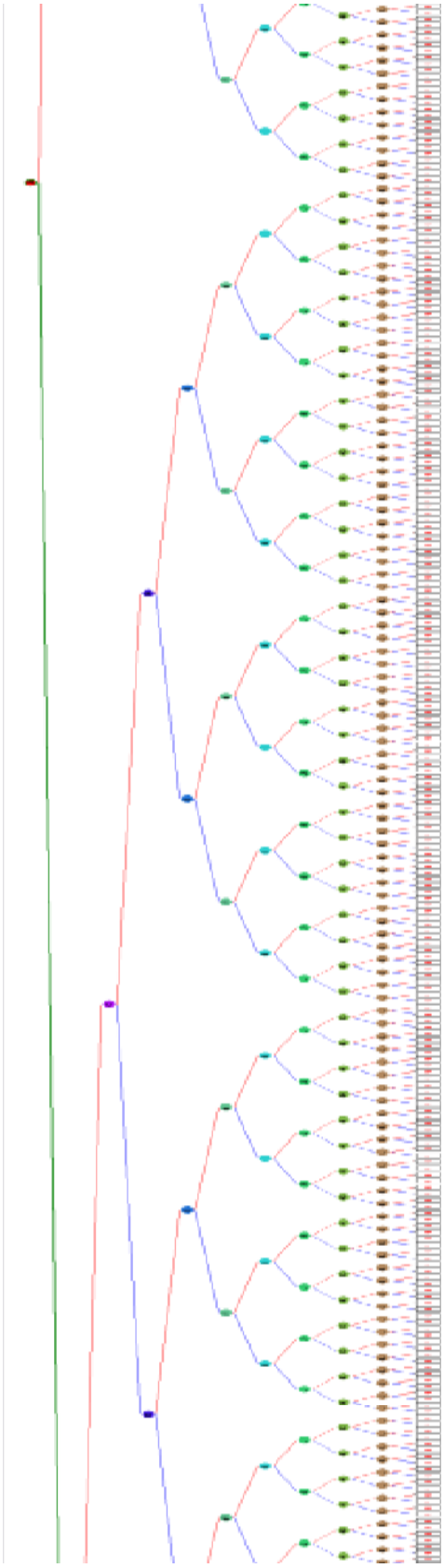
Le principal avantage de cette méthode est la rigueur. Pourtant, la méthode se révèle rapidement inutilisable pour l'analyse manuelle de grands systèmes.

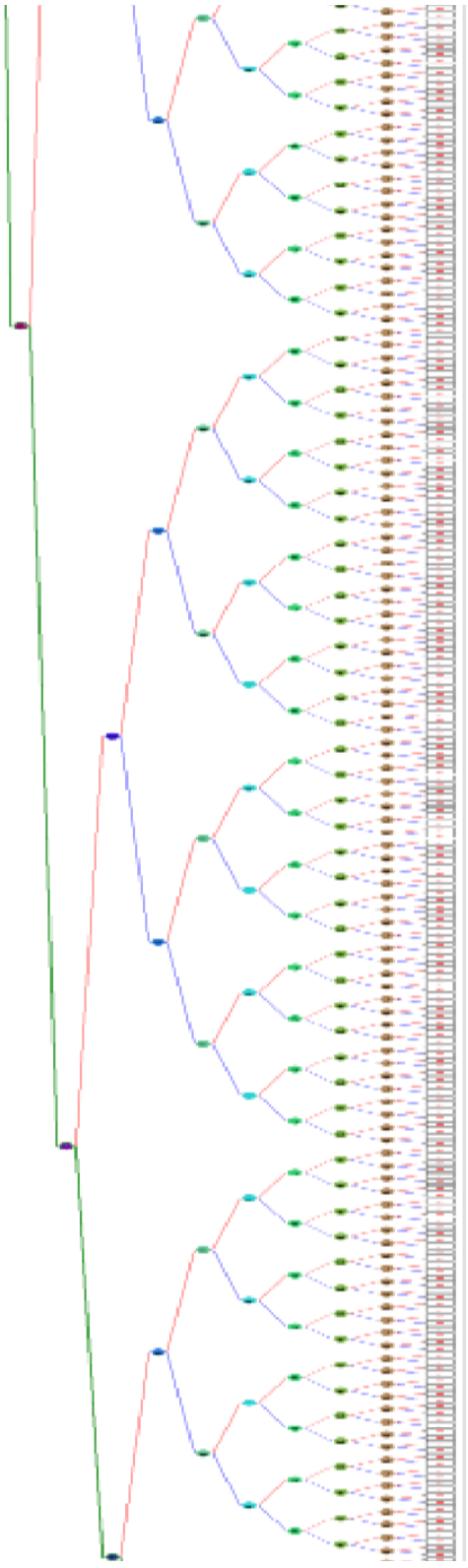
Annexe B

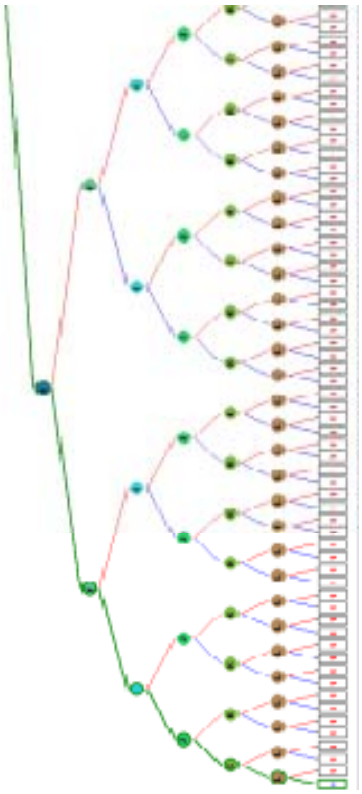
Annexe B











Références Bibliographiques

Références bibliographiques

[AYYUB, MCCUEN, 1997] : AYYUB, B.MCCUEN *"Solutions manual for probability, statistics and reliability for engineers"*, 1997

[A.BLOCH, 1982]: A.BLOCH *"Murphy's law and other reasons why things go wrong publishers*", Los Angeles 1982

[A.PAGES, M.GONDRAN, 1980] : A.PAGES, M.GONDRAN « *Fiabilité des systèmes* », Eyrolles, Paris 1980

[BARGER et al., 2002] : P.Barger, J.M.Thiriet et M.Robert *« Performance and dependability evaluation of distributed dynamical systems."*, In ESREL, Nancy 2002

[BARREAU et al., 2003] : M.Barreau, A.Todoskoff, J.Y.Morel, F.Guerin et A.Mihalache *« Dependability analysis of complex mechatronic systems. »*, In 5th IFAC Symposium Fault Detection, Supervision and Safety, Safeprocess2003, Washington, USA

[BECHOU et al., 2007] : L.Bechou, Y.Danto, J.Deletage et F.Verdier *« Nouvelles approches d'évaluation de la fiabilité : Perspectives pour les nanotechnologies. »*, In Journées scientifiques du CNFRS "NANOSCIENCES ET RADIOÉLECTRICITÉ", Paris 2007

[BELHADAOUI.H, 2011]: BELHADAOUI Hicham, *"Conception sûre des systèmes mécatroniques intelligents pour des applications critiques"*, Thèse de doctorat à l'INPL France, Janvier 2011

[BIROLINI, 1997] : A.Birolini *«Reliability engineering: theory and practice»*, 1997

[BORREL, 1996] : Borrel «*Interactions entre composants matériel et logiciel de systèmes tolérants aux fautes*», PhD thesis, LAAS, Toulouse 1996

[C.GRELLET, 2000] : C.Grellet « *Actionneurs électriques* », Ed Eyrolles 2000

[CHU, 1995] : F.Chu *« Conception des systèmes de production à l'aide des réseaux de Petri : vérification incrémentale des propriétés qualitatives. »*, PhD thesis, Faculté des Sciences UFR Mathématiques, Informatique, Mécanique, Université de Metz 1995

[COCOZZA-THIVENT, 1997] : Cocozza-Thivent *«Processus stochastiques et fiabilité des systèmes»*, Springer 1997

[DEMARCO, 1982] : T.DeMarco *« Controlling Software Projects : Management, Measurement and Estimation."*, Prentice Hall PTR 1982

[DUTUIT et al., 1997] : Y.Dutuit *«Dependability modelling and evaluation by using stochastic Petri nets: application to two test cases"*, 1997

[DUTUIT, RAUZY, 1997] : Y. Dutuit and A. Rauzy. "*Exact and Truncated Computations of Prime Implicants of Coherent and non-Coherent Fault Trees within Aralia*." Reliability Engineering and System Safety, 58:127–144, 1997

[F.KHELIFI, 2008] : F.Khelifi « *Sûreté de fonctionnement et fiabilité des circuits de l'électronique de puissance* », thèse de Magistère UMBB 2008

[F.P.LEES, 1980]: F.P.LEES "*Loss prevention in the process industries*", Butterworks 1980

[FIDES, 2004]: *"Méthodologie de fiabilité pour les systèmes électroniques"*, DGA-DM/STTC/CO/477, FIDES (2004).

[G.GALABRESE, 1947]: G.GALABRESE "*Generating reserve capacity determined by the probability method*", AIEE Transaction, Vol. 66, 1947 p. 1493-50

[I.BAZOVSKY, 1961]: I.BAZOVSKY "*Reliability theory and practice*", Prentice Hall 1961

[LADET, 1989] : P.Ladet *«Flow modelling for a class of hybrid systems"*, IEEE 1989

[LAPRIE et al., 1995] : LAPRIE et al. *«Guide de la sûreté de fonctionnement »* , 1995

[LEDOUX, 1993] : J.Ledoux *« Modèles markoviens : sur la caractérisation de l'agrégation faible et sur les modèles structurels pour l'évaluation de la sûreté de fonctionnement du logiciel. »*, PhD thesis, Université de Rennes I 1993

[LIMNIOS, 1991] : N.Limnios "Arbres de défaillance. », Hermes, Paris 1991

[LINDEMANN, 1998] : C.Lindemann *"Performance Modelling with Deterministic and Stochastic Petri Nets."*, Wiley 1998

[LITTLEWOOD, 1980] : B.Littlewood *"Theories of software reliability : How good are they and how can they be improved ?"*, IEEE Trans. Software Eng., 6(5) :489–500 1980

[M.AIME, 2000] : M.Aimé « *Évaluation et optimisation de la bande passante des convertisseurs statiques Application aux nouvelles structures multicellulaires* », thèse de l'INPT 2000

[M.SCHWOB, G.PEYRACHE, 1969]: M.SCHWOB, G.PEYRACHE "*Traité de fiabilité*", Masson et Cie Edition 1969

[MARCOVICI et LIGERON, 1974] : C.Marcovici et J.C.Ligeron *« Utilisation des techniques de fiabilité en mécanique. »*, Technique et documentation 1974

[MAVIER.J, 2007] : Mavier.J *« Convertisseurs génériques à tolérance de panne. Applications pour le domaine aéronautique »*. Thèse de doctorat, 2007

[MAZOUNI M.H, 2008] : Mazouni M.H *« Pour une meilleure approche du management des risques: de la modélisation otologique du processus accidentel au système interactif d'aide à la décision »*. Thèse de doctorat à l'INPL France, Novembre 2008

[MERLE.G et ROUSSEL.J.M, 2007]: Merle Guillaume et Roussel Jean-Marc, *"Modélisation algébrique des arbres de défaillances temprels"*; Journées doctorales du GDR MACS, Reims 2007

[MIL-HDBK, 1995]: Department of Defense, Washington DC, *"MIL-HDBK-217F – "Reliability prediction of electronic equipment"* (+ notice 1and 2) (dernière révision en 1995).

[MIL-STD 781]: U.S. Military Standard "*Reliability demonstration, exponential distribution*", MIL-STD 781

[MINATO, 1993] : S. Minato. "*Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems*." In Proceedings of the 30th ACM/IEEE Design Automation Conference, DAC'93, pages 272–277, 1993

[MONCHY.F, 2000]: Monchy F., *« Maintenance méthodes et organisation »*, Edition Dunod, 2000

[MOREL et al., 2002] : J.Y.Morel, M.Barreau et A.Todoskoff *« Petri nets : a tool adapted to computer system dependability and safety."*, In IEEE ESREL, Lyon 2002

[NELSON, 1990] : W.Nelson *« Accelerated Testing : Statistical Models, Test Plans and Data Analyses."*, John Wiley & Sons 1990

[NOUREDDINE.M, 2010]: Noureddine M., *« Application de l'AMDEC à un satellite en phase active »*, CIM 2010.

[O'CONNOR, 2002]: O'CONNOR "*Practical reliability engineering*", 2002

[P.DT O'CONNOR, 1981]: P.DT O'CONNOR "*Practical reliability engineering*", London 1981

[PAGES, GONDRAN, 1980]: PAGES et GONDRAN "*les cahiers de sécurité: les différentes méthodes d'analyse de sécurité*», Mars 1980

[PETRI, 1962] : C.A.Petri *« Commnication with Automatas. »*, PhD thesis, Universitat Bonn 1962

[PEREZ.A, 2009]: Perèz Antonio, *"Evaluation par simulation de la SdF des systèmes en contexte dynamique hybride"*, Thèse de doctorat à l'INPL France, Mars 2009

[PROCACCIA, MORILHAT, 1996] : H.PROCACCIA, C.MORILHAT *«Fiabilité des structures des installations»*, Eyrolles 1996

[RAUZY, 2001a] : A. Rauzy. "*Mathematical Foundations of Minimal Cutsets.*" IEEE Transactions on Reliability 2001

[RENSON.P, 2003]: Renson P., *« L'apport de l'AMDEC en fiabilisation des installations industrielles »*, 1er Colloque francophone Performances et Nouvelles Technologies en Maintenance PENTOM 2003, Valenciennes (France), CD-Rom 10 pages, 26-28 Mars 2003

[RIEUNEAU, 1993] : F.Rieuneau *« Sûreté de fonctionnement en phase de développement des systèmes embarqués automobiles. »*, In Integrated Logistics & Concurrent Engineering, Montpellier 1993

[S.LEVINE, F.STETSON, 1984]: S.LEVINE, F.STETSON « *Applying the lessons of PRA (Probabilistic Risk Assessment), an American perspective Nuclear Ingineering International* », Jan.1984

[T.ELCH-Heb, J.P.HAUTIER, 1993] : T.Elch-Heb, J.P.Hautier, "*Remedial strategy for inverter -induction machine system faults using two-phase operation*", EPE, Brighton 1993

[VALLON, 2003] : Vallon.J : *« Introduction à l'étude de la fiabilité des cellules de commutation à IGBT sous fortes contraintes »*, Thèse de doctorat, 2003

[VALETTE, 2002] : R.Valette "*Moyens de la sûreté de fonctionnement»*, Hermès 2002

[VILLEMEUR, 1988]: A.Villemeur, « *Sûreté de fonctionnement des systèmes industriels* », 1988

[YE et al., 2003] : H.Ye, C.Basaran et D.C.Hopkins *« Mechanical degradation of microelectronics solder joints under current stressing."*, International Journal of Solids and Structures, 40 :7269–7284 2003

[ZWINGELSTEIN, 1996]: G.ZWINGELSTEIN "*Diagnostic des défaillances*", Hermès, Paris 1995

[ZWINGELSTEIN, 2009]: Zwingelstein G., *« Sûreté de fonctionnement des systèmes industriels complexes »*, Techniques de l'Ingénieur, S 8 250v2 & S 8 251, 10 Juin 2009