Évaluation quantitative de séquences d'événements en sûreté de fonctionnement à l'aide de la théorie des langages probabilistes

Dorina-Romina Ionescu

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Evaluation quantitative de séquences d'événements en sûreté de fonctionnement à l'aide de la théorie des langages probabilistes

THÈSE

présentée et soutenue publiquement le 21/11/2016

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention Automatique, Traitement du Signal et Génie Informatique)

par

Dorina-Romina IONESCU

Composition du jury

Président :	Frédéric Kratz	Professeur, INSA Centre Val de Loire
Rapporteurs :	Antoine Grall	Professeur, Université de Technologie de Troyes
	Eric Niel	Professeur, INSA Lyon
Examinateur :	Alain Richard	Professeur, Université de Lorraine
Invité :	Gilles Deleuze	Chercheur senior, EDF R&D
Directeur de thèse :	Jean-François Pétin	Professeur, Université de Lorraine
Co-Directeur de thèse :	Nicolae Brînzei	Maître de conférences, Université de Lorraine

Centre de Recherche en Automatique de Nancy — UMR 7039

Mis en page avec la classe thesul.

A la mémoire de ma maman, devenu ange depuis 2004

Remerciements

Je tiens à adresser mes plus sincères remerciements et ma pleine reconnaissance à mon directeur et à mon co-directeur de thèse : M. Jean-François Pétin pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses multiples conseils et pour toutes les heures qu'il a consacrées à diriger cette recherche et M. Nicolae Brînzei pour toute son aide à travers le long travail de thèse, sa patience et ses indications précises et détaillées. Votre soutien constant m'a beaucoup encouragé pour atteindre l'objectif de cette thèse. Je veux remercier également les membres du Jury, qui ont accepté d'évaluer mon travail : M. Frédéric Kratz pour m'avoir fait l'honneur de présider la soutenance et d'examiner mon travail, M. Antoine Grall, M. Eric Niel, M. Alain Richard et M. Gilles Deleuze pour avoir accepté d'examiner en profondeur mon travail, ainsi que pour tous leurs commentaires constructifs.

J'exprime ma reconnaissance aux professeurs d'IUT Nancy Brabois : M. Christian Saunal pour sa manière extraordinaire de travailler et son apport dans mon expérience de moniteur et M. Franck Joly, pour sa patience et ses conseils concernant les Travaux Dirigées de Mathématiques. Un grand merci à toute l'équipe de l'IUT qui m'a donné l'opportunité de percevoir différemment l'enseignement dans l'environnement académique.

Je veux remercier Mme. Valerie Louis-Dorr pour son soutien pendant la thèse et également avant la soutenance, c'était d'une grande importance pour mon état d'esprit.

Un grand remercie à Christine Pierson pour le temps qu'elle m'a accordé depuis ma première journée dans le laboratoire, pour toutes les informations qu'elle m'a données du point de vue administratif et pour son aide dans différentes situations.

Je voudrais remercier Mme. Emilia Petrisor pour le fait de m'avoir soutenu dans l'idée de faire une thèse, pour son aide et pour la confiance qu'elle m'a accordée tout au long du travail de la recherche. Je veux remercier également Mme. Cosmina Neag pour son investissement concernant les cours de Mathématiques pendant que j'étais en Roumanie.

Je voudrais remercier aussi tous mes amis et collègues de CRAN - ENSEM. Chacun a apporté sa touche personnelle pendant les années de la thèse : Gaetan, Gundars, Vairis, Sharib, Marcos, Vitor, Julien, Michele, Maria, William, Claudia, Thomas, Prisca, Harry et Pierre.

Un grand merci à mon cher collègue de bureau Gundars, qui m'a encouragé et m'a accompagné pendant les trois années de thèse. Je l'apprécie pour toutes ses qualités humaines et professionnelles. Un grand merci également à Grégory pour les conversations constructives que nous avons eu et pour ses conseils concernant l'évolution professionnelle.

Je tiens à remercier également tous mes amis avec qui j'ai partagé cette expérience et qui m'ont encouragé tout au long du chemin : Estelle, Inga, Antoine, Gigi, Viorel, Daniela, Elena, Evelyne, Amandine, Jean-Pierre, Alina, Adi, Laurentiu, Andreea, Andra, Luminita, Andrei, Katy, Roxana, Costi, Alexandra, Catalin, Larisa, Loredana, Rovana, Nelica, Gina, Dana et Felica.

Je veux remercier très spécialement à Veronica pour son écoute, pour son amour, pour la confiance qu'elle m'a toujours accordée et pour sa présence constante dans ma vie. Un grand merci à Simona avec qui j'ai partagé les plus beaux moments à Nancy et les plus belles expériences pendant la thèse et à Benjamin pour son soutien vers la fin de mes travaux de recherche. Je ne vais jamais oublier tout ce que vous avez fait pour moi.

Pour finir, je voudrais exprimer ma profonde et sincère gratitude à ma famille : vous êtes toujours dans mon cœur partout dans le monde et je vous aime pour toute l'aide que vous m'avez donnée jusqu'à aujourd'hui. Merci beaucoup à maman d'avoir investi tout son temps dans mon éducation, de m'avoir donné une direction dans la vie et la motivation d'évoluer sur tous les plans. Merci papa d'être à coté de moi dans toutes les étapes difficiles de la vie et de m'aimer inconditionnellement. Merci à tous les autres membres pour leur amour et leur soutien.

Table des matières

Glossaire

- **SdF** : Sûreté de Fonctionnement
- **FMDS** : Fiabilité, Maintenabilité, Disponibilité, Sûreté
- **RAMS** : Reliability, Availability, Maintainability, Safety
- **MTTF** : Mean Time To Failure
- **MTBF** : Mean Time Between Failure
- **APR** : Analyse Préliminaire de Risques
- **AMDEC** : Analyse des Modes de Défaillances, de leurs Effets et Criticités
- **SIS** : Système Instrumenté de Sécurité
- **AdE** : Arbres d'Evénements
- **AdED** : Arbres d'Evénements Dynamiques
- **AdEDD** : Arbres d'Evénements Dynamiques Discrets
- **DYLAM** : Dynamic Logical Analytical Methodology
- **DETAM** : Dynamic Event Tree Analysis Method
- **ADS** : Accident Dynamic Simulator
- **AdD** : Arbres de Défaillances
- **ER** : Événement Redouté
- **BDD** : Binary Decision Diagram
- **AdDD** : Arbres de Défaillances Dynamiques
- **DFT** : Dynamic Fault Trees
- **BDMP** : Boolean logic Driven Process
- **AEF** : Automate à Etats Finis
- **SED** : Système à Événements Discrets
- **SCM** : ensemble des Séquences de Coupe Minimales
- **CdM** : Chaîne de Markov
- **ASH** : Automate Stochastique Hybride
- **RdP** : Réseaux de Petri Stochastiques
- **CMTC** : Chaîne de Markov à Temps Continu
- **PI** : contrôleur de type Proportionnel-Intégral
- **PID** : contrôleur de type Proportionnel-Intégral-Dérivée
- **TOR** : contrôleur de type Tout Ou Rien
- **DCC** : Défaillance de Cause Commune
- **GV** : Générateur de Vapeur
- **REP** : Réacteur à Eau Pressurisée
- **AAR** : Arrêt Automatique du Réacteur
- **ADG** : Bâche alimentaire et dégazeur
- **ABP** : Réchauffeurs Basse Pression
- **AHP** : Réchauffeurs Haute Pression
- **CEX** : Pompes d'extraction

- **ARE** : Régulation Débit Eau Alimentaire
- **TPA** : Turbo Pompe Alimentaire.
- **ADG** : Bâche alimentaire et dégazeur
- **ABP** : Réchauffeurs Basse Pression
- **AHP** : Réchauffeurs Haute Pression
- **VVP** : Barillet

Table des figures

Introduction générale

Classiquement, les analyses de Sûreté de Fonctionnement (SdF) visent à caractériser de manière qualitative ou quantitative le comportement dysfonctionnel d'un système. Elles peuvent se classifier en deux grands familles. Une première famille a pour objectif l'évaluation des indicateurs globaux de la SdF, comme les indicateurs probabilistes FMDS (Fiabilité, Maintenabilité, Disponibilité, Sûreté) ou des indicateurs de temps moyens tels que MTTF (Mean Time To Failure), MTBF (Mean Time Between Failure), *etc.* La deuxième famille concerne les analyses qui permettent de comprendre (qualitativement) et d'évaluer (quantitativement) le comportement d'un système par l'étude des scénarios qui le conduisent dans un état non désiré. Ce mémoire s'inscrit dans la deuxième famille d'analyses de SdF et porte sur l'identification des séquences d'événements conduisant le système dans un état de panne (analyse qualitative) et leur quantification probabiliste (analyse quantitative).

Pour mener à bien ces études qualitatives et quantitatives, les approches existantes dans la littérature sont fortement dépendantes des caractéristiques des systèmes considérés. Pour un système réparable, un scénario dysfonctionnel sera défini sous la forme d'une séquence d'événements constituée par des événements de défaillance mais aussi de réparation conduisant à un état global de panne du système. Lorsque l'on considère des systèmes dynamiques, la seule connaissance des composants défaillants reste insuffisante pour statuer sur l'état de panne ou non du système. Enfin, pour les systèmes reconfigurables (qui présentent des stratégies variables de commande et de maintenance), multi-états (qui présentent plusieurs modes de défaillance ou de fonctionnement) ou multi-phases (présentant plusieurs profils de mission), l'analyse des scénarios dysfonctionnels ne peut plus être basée sur les modèles statiques booléens de la SdF tels que les arbres d'événement, les arbres de défaillance ou encore les blocs diagramme de fiabilité.

Les modèles permettant d'aborder l'analyse de scénarios dysfonctionnels pour des systèmes dynamiques réparables, reconfigurables, multi-états, multi-phases (tels que les arbres d'événements dynamiques [Acosta et Siu, 1993], les Boolean logic Driven Markov Process (BDMP) [Bouissou et Bon, 2003], les chaînes de Markov [Howard, 1971b, Rozanov, 1975, Kemeny et Snell, 1976, Iosifescu *et al.*, 2010], les réseaux de Petri stochastiques [Molloy, 1982, Marsan *et al.*, 1984, Chiola *et al.*, 1993a], *etc.*), reposent de manière sous-jacente (explicitement ou implicitement, de manière plus ou moins formalisée) sur des descriptions de type états-transitions, couplées dans la plupart des cas à une description des séquences d'événements. En ce sens, la théorie des langages probabilistes, présentée par Garg, Kumar et Marcus [Garg *et al.*, 1999] comme une extension de la théorie des langages rationnels pour l'étude des Systèmes à Événements Discrets (SED) stochastiques, offre un cadre formel prometteur pour notre étude. En effet, à l'instar de la théorie des langages déterministes, une équivalence peut être définie, sous certaines hypothèses, entre les représentations sous forme d'automates (pour la description des comportements du système) et sous la forme de langages (pour l'analyse des séquences d'événements). D'autre part, si

la théorie des langages probabilistes a fait l'objet de travaux dans le domaine de la synthèse de la commande [Kumar et Garg, 2001,Wang et Ray, 2004,Pantelic et Lawford, 2012], son exploitation dans le domaine de la Sûreté de Fonctionnement reste un champ d'application ouvert.

Ce mémoire a donc pour objectif de définir un cadre formel, basé sur la théorie des langages probabilistes, pour, d'une part, l'identification et la quantification de séquences d'événements critiques de systèmes dynamiques, réparables et reconfigurables et, d'autre part, la mise en œuvre d'une approche compositionnelle permettant l'étude de systèmes complexes, en particulier multi-états et multi-phases.

Le premier chapitre présente la problématique scientifique faisant l'objet de la thèse. Après un bref rappel des concepts et définitions associés à la Sûreté de Fonctionnement, la première partie introduit les différentes caractéristiques des systèmes et leur impact sur les types d'analyses à effectuer. Pour les systèmes dynamiques, réparables, reconfigurables, multi-états et multi-phases, cette partie met évidence l'intérêt des notions de séquences d'événements et de scénarios dysfonctionnels ainsi que la nécessité de disposer de modèles dynamiques pour conduire des études qualitatives et quantitatives de SdF. Une synthèse des objectifs de nos travaux sera effectuée avant de dresser, dans une deuxième partie, un état de l'art du domaine. Deux catégories de modèles permettant l'analyse de scénarios dysfonctionnels sont présentés en soulignant leurs forces et leurs limitations :

- les approches standard et leurs extensions dynamiques, basées sur des équations booléennes pour caractériser les relations entre l'état du système et l'état de ses composants (arbres d'événements, arbres de défaillances, BDMP, ...),
- les modèles états-transitions permettant de modéliser les comportements fonctionnels et dysfonctionnels du système et de ses différentes configurations ou phases (automates à états finis et théorie des langages, chaînes de Markov, automates stochastiques hybrides, réseaux de Petri stochastiques et langages probabilistes).

La dernière section de ce chapitre positionne et justifie le choix de la théorie des langages probabilistes comme support à notre étude.

Le chapitre deux pose un cadre formel basé sur les langages probabilistes pour l'identification et l'évaluation quantitative des séquences d'événements, en régime asymptotique (distribution stationnaire de probabilités) et transitoire. Après une présentation détaillée de la théorie des langages probabilistes, nous soulignons que la connaissance des séquences sous la forme de langages probabilistes constitue très souvent le point de départ de la modélisation dans le cadre de cette théorie, alors que dans le domaine de la SdF, la modélisation débute plutôt par une description comportementale fonctionnelle et dysfonctionnelle sous la forme d'un automate.

L'approche proposée se déroule en trois étapes principales : modélisation du système par un automate à états finis, identification de l'ensemble de séquences (sous-langage) amenant le système dans un état quelconque (absorbant ou non), par la théorie des langages rationnels (lemme d'Arden) et quantification des séquences en régime asymptotique et transitoire. Le calcul de probabilités des séquences en régime asymptotique repose sur la chaîne de Markov à temps discret immergée dans un processus stochastique à temps continu. Ces probabilités peuvent être obtenues de manière exacte par un calcul analytique. A partir des valeurs obtenues pour les probabilités des séquences, une analyse de criticité de séquences sera effectué en se basant sur leur coût global et sur leur longueur. Une méthode de validation analytique et numérique, basée sur les calculs de probabilités d'état en utilisant la théorie classique des chaînes de Markov, est proposée. Le calcul de probabilités des séquences en régime transitoire repose, quant à lui, sur la

détermination du temps de premier passage dans un état, à l'aide d'une transformée de Laplace. On peut en déduire la densité de probabilité et la fonction de répartition qui correspond à la probabilité des séquences associées au premier passage d'un état initial vers un état cible.

Ces contributions sont illustrées sur un cas d'étude constitué d'un système de contrôle-commande de la température d'un four par un régulateur principal de type proportionnel-intégral (PI) et par un régulateur de secours de type tout ou rien (TOR). L'approche proposée donne des résultats satisfaisants pour le cas d'étude du four mais laisse apparaître des limites relatives à la modélisation et l'exploitation des expressions régulières pour des systèmes plus complexes ou de plus grande taille.

Ainsi, l'objectif du troisième chapitre est d'étendre le cadre formel, proposé au chapitre précédent, par une approche compositionnelle basée sur des opérateurs définis sur les langages probabilistes. L'intérêt est de pouvoir calculer les probabilités d'occurrence des séquences critiques pour des systèmes caractérisés par plusieurs modes de fonctionnement ou de défaillance. L'évaluation quantitative globale des séquences est obtenue à partir des évaluations locales, mode par mode, à l'aide d'un opérateur de choix modélisant la sélection non déterministe entre différents modes et d'un opérateur de concaténation modélisant une séquence de transitions d'un mode à l'autre. Cette approche modulaire est illustrée sur le cas d'étude du chapitre deux en intégrant différents modes aux caractéristiques variées : deux modes de défaillance (défaillances indépendantes et défaillance de cause commune) et deux politiques de maintenance (maintenance parfaite et imparfaite). Une généralisation de ces opérateurs pour n modes différents de fonctionnement ou de défaillance est présentée en fin de chapitre.

Cette approche modulaire est particulièrement pertinente pour les systèmes de grandes taille et/ou présentant de multiples modes de défaillances ou de fonctionnement car elle permet le calcul d'une probabilité de séquences de manière analytique à partir de modèles locaux sans nécessiter la construction, souvent difficile, d'un modèle global intégrant ces différents modes.

Le dernier chapitre présente l'ensemble de la démarche d'identification et d'évaluation quantitative de séquences d'événements sur un cas d'étude industriel. Il applique les principes de calcul des probabilités en régime asymptotique et transitoire présentés au chapitre deux et l'approche compositionnelle présentée au chapitre trois. Le cas d'étude est un système de régulation du niveau d'eau dans un Générateur de Vapeur (GV) d'un Réacteur à Eau Pressurisée (REP) comprenant notamment des pompes d'extraction, des turbopompes et des vannes réglantes. Ce cas d'étude, fourni par EDF, a été abordé dans un projet précédent de notre équipe de recherche, le projet APPRODYN [Aubry *et al.*, 2012a]. La conduite de ce système est réalisée selon trois phases : démarrage et montée en puissance, régime nominal en puissance et baisse de puissance. Chaque phase peut être caractérisée par un ou plusieurs modes de fonctionnement et de défaillance qui seront, chacun, représentés par un automate probabiliste. La phase nominale où le système est stabilisé est représenté par un mode unique de fonctionnement et de défaillance pour lequel quelques séquences caractéristiques seront évaluées par un calcul de probabilité en régime asymptotique. Pour les autres modes, transitoires par nature, un calcul de probabilité en régime transitoire sera effectué. Enfin, les opérateurs de choix et de concaténation pour évaluer les probabilités des séquences impliquant respectivement une sélection ou un changement de mode. Compte tenu des caractéristiques intrinsèques de ces changements ou sélection de mode, le calcul sera effectué en régime transitoire.

Le mémoire se conclue en présentant les principales contributions de ce travail et en proposant des perspectives de recherche en termes d'extension du cadre formel proposé.

Chapitre 1

Positionnement scientifique

1.1 Introduction

L'objectif de ce mémoire est l'identification et la quantification probabiliste de séquences d'événements, qui caractérisent le comportement d'un système, et l'évaluation de leur criticité, dans le cadre des études de Sûreté de Fonctionnement (SdF). Ce chapitre débute par une présentation des notions scientifiques élémentaires de la Sûreté de Fonctionnement, en particulier dans le contexte des systèmes dynamiques, réparables, reconfigurables, multi-états et multi-phases qui constituent l'objet d'étude de cette thèse. La seconde partie de ce chapitre dresse un état de l'art des modèles sur lesquels les études de SdF peuvent se baser. Les modèles dans lesquels la défaillance du système s'exprime au travers de la fonction de structure booléenne seront présentés dans la section 1.3.1, les modèles basés sur l'espace d'états seront décrits dans la section 1.3.2. Enfin, le chapitre se conclut par une synthèse argumentée qui justifie le choix des langages probabilistes dans le cadre de notre étude.

1.2 Contexte

1.2.1 Sûreté de fonctionnement (SdF)

Au sens le plus strict, la Sûreté de Fonctionnement est l'aptitude d'une entité à assumer une ou plusieurs fonctions requises dans des conditions données [CEI, 1990a]. De manière plus large, la Sûreté de Fonctionnement est définie par [Villemeur, 1988], comme la science des défaillances.

Conformément à la norme CEI 61508 [CEI, 2000], une *défaillance* est définie comme étant la cessation de l'aptitude d'une unité fonctionnelle à accomplir une fonction requise. Ainsi, elle représente le passage d'un état de bon fonctionnement à un état d'inaptitude à accomplir la fonction. Une défaillance d'un système est observée lorsque le service délivré dévie du service spécifié. On parlera alors de comportement dysfonctionnel.

La défaillance est la conséquence d'un comportement erroné du (ou d'une partie du) système [Kombé, 2011]. Une *erreur* est définie, par la norme CEI 61508 [CEI, 2000], comme l'écart ou la discordance entre une valeur ou une condition calculée, observée ou mesurée et la valeur ou condition vraie, prescrite ou théoriquement correcte. Cette définition (erreur de mesure) relie la notion d'erreur à la mesure d'une grandeur physique plutôt qu'au processus conduisant à un dysfonctionnement du système. Dans le travaux de [Kombé, 2011], une erreur est vue comme une composante valuée de l'état du système susceptible d'entraîner une défaillance. La cause adjugée ou supposée d'une erreur est une faute dont l'origine est due à un phénomène physique ou à un

comportement humain erroné. Une erreur est alors la manifestation d'une faute dans le système, alors qu'une défaillance est l'effet d'une erreur sur le service.

Dans le cadre de nos travaux, le concept de défaillance est abordé à partir de deux points de vue :

- *état défaillant*, que l'on appellera état de panne, est l'état (où les états) dans lequel le système n'accomplit plus sa fonction ou la fonction délivrée ne coïncide pas avec celle spécifiée ;
- *événement de défaillance* qui amène, suite à son occurrence, les composants d'un système (ou même le système) dans un état de panne.

Pour mettre en évidence les deux sens d'utilisation du concept de défaillance, on considère comme exemple un régulateur de température d'un système quelconque. L'occurrence d'un événement de défaillance amène le régulateur dans un état dysfonctionnel ou il n'accomplit plus sa fonction. Dès que la température n'est plus contrôlée, le système va se retrouver dans un état de panne qui peut être critique pour les humains mais aussi pour l'environnement.

[Laprie *et al.*, 1995] enrichit la définition de la sûreté de fonctionnement d'un système en la considérant comme la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre. Cette notion de confiance est fondamentale, dans la mesure où tous les systèmes matériel/logiciel peuvent subir des dysfonctionnements, qu'ils soient dus à des erreurs systématiques introduites lors des phases de conception ou à l'occurrence de défaillances. Cette confiance est classiquement évaluée au travers des quatre indicateurs que sont la fiabilité, la maintenabilité, la disponibilité et la sécurité (FMDS). L'équivalent Anglo-Saxon utilisé pour désigner la Sûreté de Fonctionnement est le terme *dependability* (*reliability*, *availability*, *maintainability*), souvent désigné par l'acronyme RAM ou RAMS si l'on y adjoint la sécurité (*safety*).

La *fiabilité* est définie comme l'aptitude d'une entité à accomplir une fonction requise dans des conditions données pendant une durée donnée [Villemeur, 1988], [CEI, 1990a]. Les conditions données concernent les modes d'utilisation de l'entité ayant un impact sur la fiabilité, tels que les modes de fonctionnement, les conditions environnementales ou la maintenance. Autrement dit, la fiabilité peut être mesurée par la probabilité qu'une entité accomplisse une fonction requise dans les conditions données pendant l'intervalle de temps $[0, t]$. Selon la nature des entités considérées, il y a différents moyens pour évaluer cette probabilité.

La *maintenabilité* est la capacité d'une entité à être maintenue ou rétablie dans un état dans lequel elle peut accomplir une fonction requise lorsque la maintenance est accomplie dans des conditions données avec des procédures et des moyens prescrits [Villemeur, 1988], [CEI, 1990a]. Cet indicateur de SdF est mesuré par la probabilité que la maintenance d'une entité s'achève à l'instant t, sachant que l'entité est défaillante à l'instant $t = 0$. L'évaluation de cette probabilité est liée à la manière dont la remise en état de fonctionnement de l'entité est effectuée.

La *disponibilité* est l'aptitude d'une entité à être en état d'accomplir une fonction requise dans des conditions données et à un instant donné, ou pendant un intervalle de temps donné [Villemeur, 1988], [CEI, 1990a]. La mesure de la disponibilité est donnée par la probabilité qu'une entité soit en état d'accomplir une fonction requise dans des conditions données à l'instant t. L'inverse de la disponibilité est l'indisponibilité.

Le dernier indicateur SdF est la *sécurité* ; selon [Laprie *et al.*, 1995], on parle de "sécurité innocuité", définie comme la capacité d'une entité à éviter de faire apparaître des événements critiques ou catastrophiques c'est-à-dire pouvant affecter les personnes ou les équipements.

Les préoccupations dites de sécurité sont très présentes dans le monde de l'énergie nucléaire, des transports ou dans les domaines aéronautique et spatial. Dans les installations de production manufacturière ou batch, les préoccupations sont plutôt liées à la disponibilité. Dès lors que la

sécurité ou la disponibilité d'un système est mise en défaut, on incrimine sa fiabilité. Enfin, en cas de dysfonctionnement, il convient de remettre le système en conditions de fonctionnement initial : c'est là qu'intervient la maintenabilité. Ces quatre caractéristiques constituent la Sûreté de Fonctionnement d'un dispositif.

Enfin, la sûreté de fonctionnement peut-être considérée comme l'ensemble des moyens d'ingénierie déployés pour spécifier, concevoir, réaliser et exploiter des systèmes où la défaillance est naturelle mais reste tolérable au regard des indicateurs FMDS. Une définition alternative donnée par [Laprie, 2004] de la SdF est l'aptitude à éviter des défaillances du service délivré plus fréquentes ou plus graves qu'acceptable.

Il existe plusieurs méthodes pour améliorer les indicateurs de SdF. Parmi les plus classiques, le principe de redondance consiste à mettre à disposition de multiples ressources pour réaliser une même fonction ou une même tâche. En ingénierie, la redondance de systèmes au sein d'une machine vise à améliorer la fiabilité de cette dernière. En multipliant les composants du système on prévient la défaillance de l'un d'entre eux. Un système redondant améliorant la sûreté de fonctionnement d'un système donné est basé sur des structures parallèles : un système constitué de n éléments redondants est fiable ou disponible si au moins un de ses éléments fonctionne. Deux types de redondance peuvent être mis en œuvre :

- la redondance active (ou chaude) caractérisée par le fait que les composants redondants du système fonctionnent simultanément ;
- la redondance passive (ou froide) qui consiste à n'utiliser qu'un seul des composants à la fois et à ne démarrer la second que lorsque le premier tombe en panne.

1.2.2 Analyses de Sûreté de Fonctionnement

De manière générale, les analyses de SdF ont pour objectif de caractériser de manière qualitative ou quantitative le comportement dysfonctionnel d'un système. Elles se décomposent généralement en deux grandes familles :

- les analyses visant à évaluer quantitativement les indicateurs caractéristiques de la SdF, tels que les indicateurs probabilistes FMDS ou encore des indicateurs de temps moyens tels que MTTF (Mean Time To Failure), MTBF (Mean Time Between Failure), *etc.* Cette évaluation peut être prévisionnelle pour permettre, par comparaison aux objectifs, d'identifier les actions de conception ou d'amélioration de l'entité étudiée, mais peut également être réalisée pour suivre les performances d'un système en exploitation.
- les analyses permettant de comprendre (qualitativement) et d'évaluer (quantitativement), les comportements dysfonctionnels d'une entité au travers de l'étude de scénarios conduisant l'entité dans un état non désiré.

Dans le cadre de ce mémoire, nous nous intéressons à la seconde famille d'analyse de SdF portant sur l'identification et la compréhension des séquences d'événements conduisant le système dans un état non désiré (analyse qualitative) et leur quantification en termes de probabilités (analyse quantitative).

1.2.2.1 Analyses qualitatives vs analyses quantitatives

En connaissant la structure du système à analyser et le modèle fonctionnel qui le représente, l'objectif des analyses qualitatives de scénarios dysfonctionnels est de comprendre les mécanismes, les risques et les combinaisons ou séquences d'événements conduisant le système à des états non désirés.

Les méthodes qualitatives les plus industriellement diffusées pour parvenir à ces fins sont l'Analyse Préliminiare des Risques (APR) et l'Analyse des Modes de Défaillances, de leurs Effets et Criticités (AMDEC) qui fait l'objet de normes internationales (X 60-510, CEI 812 et MIL-STD-1629A). Ces analyses suivent un processus inductif qui démarre du niveau le plus bas pour lequel on dispose d'information sur de possibles défaillances (composants, sous-ensemble) pour en analyser les effets au niveau le plus haut (système, produit).

Plus précisément pour l'AMDEC, l'objectif est de référencer les différents modes de défaillance d'un composant ou d'une fonction, d'identifier les causes primaires ou secondaires associées à chaque mode de défaillance ainsi que les conséquences sur le composant ou la réalisation de la fonction et enfin d'estimer leur criticité. Cette analyse peut être enrichie par un ensemble de préconisations permettant de réduire les risques et la fréquence d'occurrence des modes de défaillance : ces préconisations peuvent être établies de manière prévisionnelle en phase de conception ou sur la base d'observations des défaillances associées à des mesures (statistiques-rendement).

Même s'il existe théoriquement des liens entre causes, modes et conséquences (un mode de défaillance à un niveau donné peut, par exemple, être considéré comme une cause à un autre niveau), il demeure difficile de déterminer de manière exhaustive les combinaisons ou les séquences de défaillance entraînant la défaillance du système à l'aide d'une analyse de type APR ou AMDEC. De plus, ces approches informelles ne garantissent aucune exhaustivité quant à l'identification des scénarios conduisant à la défaillance du système.

D'autres approches ont donc été développées pour identifier ou déterminer, de manière formelle, des combinaisons ou séquences d'événements (critiques) susceptibles de conduire à la défaillance globale du système (ou à un état non désiré dans le cadre d'une étude donnée). De plus, le niveau de formalisation de ces approches permet d'effectuer des calculs d'indicateurs probabilistes de SdF (probabilité d'occurrence de scénarios notamment), sous réserve que chaque événement élémentaire du système puisse être probabilisé à partir d'une loi soigneusement paramétrée et de la connaissance du temps de mission associé à l'événement redouté et/ou à l'aide de données issues du retour d'expérience.

Les hypothèses de modélisation sous-jacentes à ces approches ainsi que la nature des résultats obtenus dépendent fortement des caractéristiques intrinsèques des systèmes étudiés et de leurs propriétés. Avant de présenter un éventail de ces différentes approches formelles en section 1.3, les sections suivantes précisent quelques caractéristiques principales des systèmes et leurs impacts sur les analyses formelles de SdF.

1.2.2.2 Coupes vs séquences d'événements

Selon Z.W. Birnbaum [Birnbaum *et al.*, 1961], un système statique est un système dont la défaillance à chaque instant est uniquement déterminée par la défaillance de ses composants au même instant. Dans ce contexte, il est alors possible de définir une fonction de structure [Husson *et al.*, 2007] qui représente la relation entre la défaillance du système et la défaillance de ses composants. Sur la base de cette fonction, une coupe sera définie comme une combinaison d'événements (défaillances de composants) qui entraîne l'événement indésirable. Une coupe sera dite minimale si elle ne contient aucune autre coupe du système.

La principale conséquence de ces définitions est que, pour les systèmes statiques, la connaissance de la fonction de structure est suffisante pour déduire l'état global de défaillance (ou non) du système à partir de l'état de chacun de ses composants. De plus, l'ordre d'occurrence des événements de défaillance des composants est sans effet sur l'état global du système. L'analyse des scénarios de défaillance est donc basée sur l'analyse des coupes minimales. De ce fait, la

coupe couvre l'ensemble de toutes les séquences ordonnées d'événements qu'elle contient et sa probabilité d'occurrence représente la somme des probabilités de toutes ces séquences.

Pour le cas des systèmes dynamiques, objets de notre étude, l'utilisation des coupes dans l'évaluation des indicateurs de SdF présente quelques limites. En effet, un système dynamique est un système pour lequel la fonction de structure est insuffisante pour déduire l'état de panne (ou non) du système global. Les raisons pour lesquelles un système ne peut être qualifié comme statique au sens de ces définitions sont diverses :

- la structure du système est fixe, mais l'ordre d'occurrence des événements de défaillance de ces composants a une influence sur l'état global de défaillance (ou non) du système. Par exemple, considérons une installation industrielle avec son système instrumenté de sécurité (*SIS*). L'événement e_1 représente une défaillance dangereuse de l'installation dont le SIS doit annihiler les effets ; l'événement e_2 représente la défaillance du SIS. Si la défaillance du SIS se produit avant la défaillance l'installation, le système va se retrouver dans un état critique puisque le SIS n'aura pu jouer son rôle ; en revanche, si la défaillance de l'installation survient avant la défaillance du SIS, celui aura eu la possibilité d'exercer son action après occurrence de e_1. Comme le montre la figure (1.1), une séquence d'événements peut conduire le système dans un état dangereux alors que les mêmes événements se produisant dans un ordre différent n'y conduisent pas ;

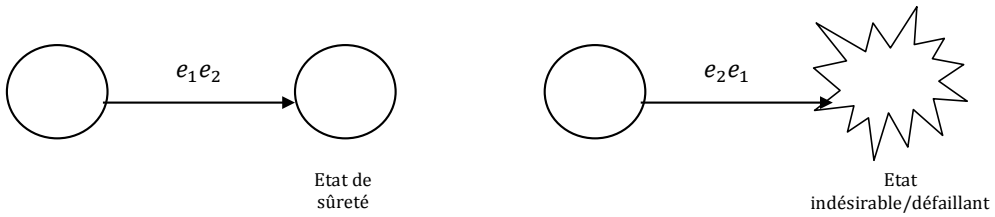


FIGURE 1.1 – Installation industrielle dans deux états différents

- l'occurrence d'un événement de défaillance ou de réparation peut dépendre de l'occurrence antérieure d'autres événements (dans les systèmes automatisés, l'occurrence d'un événement peut être interdite par le système de commande en fonction de l'occurrence précédente, ou non, de certains événements) ;
- la structure du système évolue au cours du temps, un tel système est alors qualifié comme système multiphasé [Alam et Al-Saggaf, 1986].

Si dans les systèmes statiques les études de SdF peuvent être basées sur l'analyse des coupes, dans les modèles dynamiques cette analyse n'est plus suffisante. En effet, l'utilisation des coupes conduirait à une surévaluation des paramètres de SdF puisque seules certaines séquences peuvent conduire à la défaillance du système. Pour les systèmes dynamiques, l'analyse des scénarios de défaillance doit donc se baser sur l'analyse de séquences d'événements qui permet de décrire l'évolution de certains composants du système à partir d'un état de bon fonctionnement jusqu'à l'occurrence de l'événement redouté tout en tenant compte du comportement du système.

1.2.2.3 Scénarios de défaillance vs scénarios dysfonctionnels

Un système est considéré *réparable* si au moins un de ses composants est réparable [Crow, 1975]. Un composant est réparable s'il peut, dans des conditions données après une défaillance, être remis dans un état lui permettant de fonctionner tel que requis [CEI, 1990b]. Les comportements dysfonctionnels de ces systèmes seront donc caractérisés par deux familles d'événements

complémentaires : les événements de défaillance et les événements de réparation (événement faisant passer un équipement d'un état de dysfonctionnement à un état de bon fonctionnement).

Un *scénario de défaillance* sera donc défini comme une séquence d'événements de défaillance, comportant un ordre d'occurrence précis, qui décrit la façon dont un système quitte le bon fonctionnement pour évoluer vers un fonctionnement défini comme dangereux. Le scénario de défaillance doit décrire cette évolution de manière précise pour la compréhension et de manière concise. Un scénario est ainsi vu comme une description du système sous la forme d'un changement d'état (état initial vers état final) et d'une suite d'événements qui mènent à l'état critique ou dangereux.

Pour un système réparable, un *scénario dysfonctionnel* sera donc défini comme une séquence d'événements comprenant des événements de défaillance mais aussi de réparation conduisant à un état global de défaillance du système. En d'autres termes un scénario dysfonctionnel commence à partir de causes (si elles sont connues) conduisant à un effet final particulier (l'état de panne du système) en passant éventuellement par des phases de réparation. L'évaluation de scénarios dysfonctionnels constitue un moyen efficace d'accroître la précision d'évaluation des performances de SdF en limitant le nombre de séquences possibles [Clarhaut, 2009].

1.2.2.4 Modèles statiques vs modèles dynamiques

Au delà des caractéristiques abordées dans les sections précédentes (système dynamique, réparable), d'autres propriétés relatives aux comportements fonctionnels et dysfonctionnels des systèmes peuvent avoir un impact sur les modèles sous-jacents aux études de SdF.

Un *système "multi-phases"* doit réaliser une mission qui aura été décomposée en plusieurs phases. Le passage d'une phase à l'autre induit une modification de la structure du système et/ou de son comportement dysfonctionnel et/ou de ses critères de succès [Burdick *et al.*, 1977,Meshkat *et al.*, 2003]. Les systèmes périodiquement testés où les phases sont constituées par les intervalles entre tests successifs en sont un exemple [Signoret, 2005]. Dans certains cas, les composants d'un système multi-phases sont *multi-états*. Cela signifie que l'état de panne d'un composant peut ne pas être binaire mais caractérisé par différents modes de défaillance (nominal, dégradé, défaillant par exemple). Les deux aspects (multi-phases, multi-états) peuvent se combiner par exemple dans le cas où un changement de phase de mission se traduit par un changement de mode d'opération de certains composants.

Enfin, la dernière propriété que nous considérons est relative à la notion de *reconfiguration* des systèmes. Cette notion est associée dans la littérature à des thématiques de contrôle des SED, notamment pour concevoir un contrôle tolérant aux fautes [Faraut *et al.*, 2010] et [Paoli *et al.*, 2011]. En effet, selon [Kanso et Berruet, 2010], les systèmes reconfigurables offrent les flexibilités nécessaires qu'ils exploitent de façon à proposer, et à maintenir une qualité de service adéquate malgré la présence de perturbations (pannes altérant le fonctionnement, intégration d'une nouvelle machine, variation de la demande, etc.).

Dans sa thèse, P.-Y. Piriou définit un système reconfigurable comme tout système qui dispose de mécanismes de commutation permettant de modifier dynamiquement sa structure et/ou le comportement de ses composants (*e.g.* changement de mode d'opération) [Piriou, 2015]. La mise en place d'une reconfiguration d'un système peut être justifiée pour des raisons :

- fonctionnelles (après un changement de phase) ;
- de maintenance (pour réaliser des tests périodiques) ;
- de sécurité (afin de conduire le système dans un état sauf) ;
- de production (pour s'adapter aux aléas du carnet de commande) ;
- de tolérance aux fautes (par exemple la redondance passive).

L'impact de la mise en œuvre des mécanismes de commutation sur la SdF du système est remarqué sur plusieurs aspects :

- les redondances passives sont mises en place pour améliorer la sûreté de fonctionnement ;
- le changement de mode d'opération d'un composant implique souvent une modification de son comportement dysfonctionnel (*e.g.* les taux de défaillances d'un composant peuvent augmenter si on le surcharge pour s'adapter aux aléas du carnet de commande) ;
- un changement de phase peut modifier les critères de succès ;
- les commutations augmentent les risques de défaillance à la sollicitation.

Pour qualifier et quantifier les impacts d'une stratégie de reconfiguration d'un système, il est nécessaire de prendre en compte les mécanismes de commutation et leurs effets dans les modèles de SdF. Dans ce contexte, l'évaluation de SdF de systèmes reconfigurables multi-phases et multi-états, comme par exemple un système caractérisé par plusieurs modes de défaillance ou par de multiples politiques de maintenance, met en évidence la nécessité de disposer de modèles de représentation dynamiques.

Ces représentations devront caractériser les différentes variations possibles du point de vue du comportement fonctionnel et dysfonctionnel du système dans le temps. Par opposition à un modèle statique, le comportement dynamique sera caractérisé par des équations de sorties (en fonction des entrées et des états du système) et par des équations d'état (donnant l'état du système en fonction de ses états précédents et des entrées). Le modèle mathématique de ces équations peut être un système d'équations différentielles ou récurrentes. Dans le contexte de SdF, les équations de Chapman Kolmogorov, par exemple, donnent la relation entre les probabilités pour le système (sorties) d'être dans l'un de ses états (n'importe lequel) en connaissant les taux de transition probabilistes (défaillances ou réparations de ses composants) entre ces états.

1.2.2.5 Positionnement de l'étude

Dans le cadre de ce mémoire, nous nous intéressons à l'étude de la sûreté de fonctionnement de systèmes dynamiques, réparables, reconfigurables, multi-états et multi-phases. L'analyse de ces propriétés dans les sections précédentes permet de fixer le cadre scientifique de notre étude :

- l'analyse de SdF sera basée sur l'étude des séquences d'événements plutôt que sur celle des coupes (systèmes dynamiques),
- les séquences d'événements considérés contiennent des événements de défaillance et de réparation (systèmes réparables),
- l'analyse de SdF reposera sur des modèles dynamiques comportementaux des systèmes étudiés (systèmes reconfigurables, multi-états et multi-phases).
- l'analyse des séquences d'événements sera qualitative (identification des séquences) et quantitative (évaluation d'indicateurs de SdF de ces séquences, notamment leur probabilité d'occurrence).

1.3 Modèles pour l'analyse des scénarios dysfonctionnels

L'état de l'art relatif aux approches permettant la détermination et l'évaluation quantitative de séquences d'événements est présenté en classant les modèles en deux grandes catégories :

- les modèles booléens dans lesquels la défaillance du système s'exprime en fonction de la défaillance de ses composants à l'aide d'une fonction de structure booléenne,
- les modèles basés sur une représentation états/transitions décrivant l'évolution du comportement fonctionnel et dysfonctionnel du système.

1.3.1 Modèles booléens

Les approches appartenant à cette première catégorie sont caractérisées par le fait que l'état du système est donné en fonction de l'état de ses composants par une équation booléenne, ce qui veut dire que les termes de l'équation sont des produits des variables aléatoires booléennes associées aux composants ou aux modes de défaillance des composants.

Plusieurs types de modèles ont été développés en se basant sur cette approche booléenne. Parmi tous ces modèles, nous allons présenter les modèles les plus répandus et plus utilisés dans le domaine de la sûreté de fonctionnement, à la fois en recherche et en industrie.

1.3.1.1 Arbres d'événements (AdE)

L'analyse par arbre d'événements a été développée au début des années 1970 pour être utilisée essentiellement dans le contexte des études probabilistes de sûreté des centrales nucléaires [Cepin, 2011, Bouissou, 2008]. Le fait que cet outil soit employé dans ce domaine s'explique sans doute par le fait qu'il est bien adapté à la description d'un ensemble de parades (fonctions de sécurité) qui peuvent être opposées au développement d'un incident (événement) initiateur survenant sur une installation munie de divers dispositifs de sécurité.

A l'instar de l'analyse par arbre des défaillances dont elle s'inspire, elle permet d'estimer les probabilités d'occurrence de séquences accidentelles. Cette méthode est particulièrement utilisée dans le domaine de l'analyse après accidents en vue d'expliquer les conséquences observées résultant d'une défaillance du système. La suite de ce paragraphe rappelle les principes fondamentaux des arbres d'événements puis présente une extension dynamique de ces derniers, connue sous l'appellation "arbres d'événements dynamiques".

Arbres d'événements

Les arbres d'événements sont des modèles graphiques sous forme d'arbres qui sont le résultat d'une analyse inductive du système et qui permettent de modéliser et d'analyser des séquences complexes d'événements [Papazoglou, 1998]. Ces séquences d'événements peuvent être discrétisées en fonction de leurs effets possibles, ou, en fonction de leur distinction, dans une série d'événements simples. Malgré le caractère généraliste des arbres d'événements, ils ont été utilisés principalement en référence à des modèles logiques décrivant les performances d'un système complexe.

L'analyse utilisant les arbres d'événements est une technique utilisée pour définir les séquences d'un accident potentiel associé à un événement initiateur particulier ou à un ensemble d'événements initiateurs. L'analyse des séquences dans le système montre la réussite et l'échec de la mission du système (en général une mission de sécurité) et les actions (parades) disponibles. Selon l'échec ou la réussite des parades, on peut aboutir à plusieurs catégories de conséquences : c'est ce point qui joue un rôle déterminant dans le choix d'un arbre d'événements plutôt qu'un arbre de défaillances.

Les séquences d'événements analysées contiennent un événement initiateur et les échecs ou les réussites des fonctions de sécurité. L'événement initiateur est un événement qui pourrait compromettre la sécurité si les systèmes de sécurité n'empêchent pas les conséquences indésirables.

L'évaluation par arbres d'événements peut être qualitative et/ou quantitative, tout comme l'analyse utilisant les arbres de défaillance. L'évaluation se fait de manière que chaque séquence décrivant un accident potentiel est évaluée séparément. Lorsque les séquences de tous les accidents sont évaluées, l'évaluation de l'arbre d'événements est finie.

L'expression qui nous donne la possibilité d'effectuer l'évaluation quantitative d'un arbre d'événements est :

$$\mathbb{P}(s_i) = \mathbb{P}(e_1) \times \mathbb{P}(e_2|e_1) \times \mathbb{P}(e_3|e_1 \cap e_2) \times ... \times \mathbb{P}(e_m|e_1 \cap e_2 \cap ... \cap e_{m-1}), \quad (1.1)$$

où $\mathbb{P}(s_i)$ est la probabilité d'occurrence d'une séquence s_i, m est le nombre d'événements de base dans la séquence et $\mathbb{P}(e_1), ..., \mathbb{P}(e_m)$ représentent les probabilités de défaillance des événements de base $e_1, ..., e_m$.

Sous l'hypothèse que les événements de base sont mutuellement indépendants, on peut réécrire l'équation (1.1) comme :

$$\mathbb{P}(s_i) = \prod_{j=1}^{m} \mathbb{P}(e_j) \quad (1.2)$$

La figure suivante présente un exemple simple d'arbre d'événements. L'événement initiateur est noté par « initiateur », les parades sont A, B, C et c_1, c_2, c_3 sont des conséquences. Si la parade A fonctionne, les parades B et C ne sont pas déclenchées et les conséquences de l'initiateur sont acceptables (c_1). Sinon il faut que B et C fonctionnent pour éviter des conséquences inacceptables, ce qui signifie l'arrivée à un état dégradé du système (conséquence c_2). Si A échoue et au moins une des deux parades B ou C échoue on aura des conséquences inacceptables (c_3).

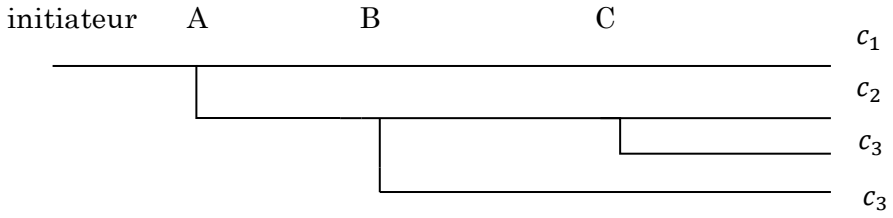


FIGURE 1.2 – Un exemple d'arbre d'événements

On note avec $\mathbb{P}(A)$, $\mathbb{P}(B)$ et $\mathbb{P}(C)$ les probabilités d'échec des parades A, B et C. Si les échecs des parades sont des événements indépendants, on peut calculer la probabilité de chaque type de conséquences en considérant que chaque extrémité d'une branche de l'arbre peut être associée au produit des probabilités des embranchements qui amènent à cette extrémité (équation 1.2). Les probabilités d'occurrence pour les trois conséquences sont alors :

$$\mathbb{P}(c_1) = 1 - \mathbb{P}(A),$$

$$\mathbb{P}(c_2) = \mathbb{P}(A) \cdot (1 - \mathbb{P}(B)) \cdot (1 - \mathbb{P}(C)),$$

$$\mathbb{P}(c_3) = \mathbb{P}(A) \cdot [(1 - \mathbb{P}(B)) \cdot \mathbb{P}(C) + \mathbb{P}(B)].$$

Les séquences d'événements, analysées et évaluées indépendamment les unes des autres à partir d'un arbre d'événements, peuvent en outre être combinées dans des groupes de séquences d'accidents similaires et les résultats communs peuvent contribuer à l'étude globale de sûreté de fonctionnement.

Arbres d'événements dynamiques (AdED)

Les AdED sont définis comme une extension des arbres d'événements, par le fait que les ramifications sont autorisées à se produire à différents points dans le temps [Acosta et Siu, 1993].

Une ramification de l'arbre peut être causée par des événements de défaillance des composants aussi bien que par le franchissement du seuil d'une variable décrivant l'évolution physique d'un paramètre du système au fil du temps. Les taux de défaillance des composants peuvent avoir des valeurs constantes aussi bien qu'exponentielles ou distribuées selon toute autre loi. En plus ces lois peuvent dépendre de l'évolution des variables physiques [Cojazzi, 1996].

Quelle que soit la déclinaison du modèle, il est consacré à l'évaluation par simulation (les arbres d'événements dynamiques discrets (AdEDD) représentent une méthode spécifique pour gérer le temps au cours de la simulation). Toutes les variables dépendantes du temps sont déterminées périodiquement aux même instants pour évaluer la possibilité d'occurrence d'un événement. Les séquences d'événements sont générées par des règles au fur et à mesure que l'analyse de l'arbre progresse. A noter que les séquences ne sont pas spécifiées dans leur intégralité en début d'analyse mais que les règles de génération doivent l'être.

Pratiquement, ce modèle a été implémenté dans plusieurs outils de simulation (Dynamic logical analytical methodology-DYLAM [Cojazzi, 1996], Dynamic event tree analysis method-DETAM [Kermisch et Labeau, 2000, Acosta et Siu, 1993] et Accident dynamic simulator-ADS [Kermisch et Labeau, 2000, Hsueh et Mosleh, 1996]). Ces méthodes sont différentes du point de vue de l'implémentation de la technique de ramification, de la mémorisation de l'arbre et de la modélisation des interactions.

La figure (1.3) représente un AdED simple, contenant deux systèmes binaires, le système A et le système B. Il y a trois aspects intéressants à remarquer dans la figure : toutes les combinaisons possibles des états du système doivent être considérées à chaque point de ramification, les ramifications se réalisent à des moment arbitraires (et discrets) du temps et le nombre de séquences d'événements peut rapidement atteindre une taille difficile à gérer si diverses approximations visant à limiter le problème ne sont pas appliquées.

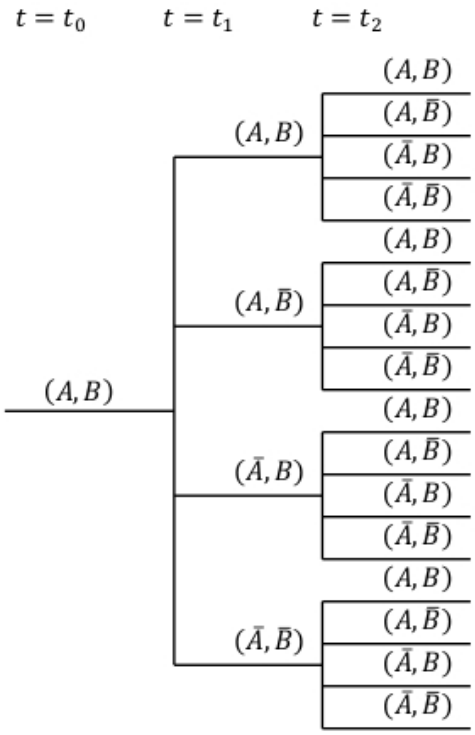


FIGURE 1.3 – Un exemple d'arbre d'événements dynamique

Pour formaliser le concept d'AdED, cinq ensembles sont définis [Acosta et Siu, 1991] :

- l'ensemble de ramifications
- l'ensemble de variables définissant l'état du système
- les règles de ramification
- les règles d'expansion des séquences
- les outils de quantification

L'ensemble des ramifications est l'ensemble des variables utilisé pour déterminer les parades (les séquences d'événements) existantes par rapport à chaque nœud de l'arbre. Dans l'exemple qui a été pris dans la figure (1.3) les ramifications sont déterminées par l'état commun des systèmes A et B ; l'ensemble des ramifications peut alors être écrit comme $\{X_a X_b\}$, où X_a est l'indicateur binaire pour l'état du système A ($X_a = 1$ si le système A est fonctionnel et 0 s'il est défaillant) et X_b est l'indicateur pour le système B.

L'état global du système est défini par les variables qui influent sur l'affectation de la fréquence pour les différentes ramifications. De plus, l'état du système peut aussi être exprimé à partir de variables caractérisant des fonctions déterministes de la séquence d'événements courante.

Les règles de ramifications sont des règles utilisées pour déterminer quand une ramification (parade) devrait avoir lieu. Dans sa forme la plus simple, l'ensemble de règles de ramification est un ensemble d'instants de ramification (ou une constante Δ_t) établi avant de commencer l'analyse.

Les règles d'expansion des séquences sont utilisées pour limiter le nombre de séquences et, par conséquent, l'expansion de l'arbre. Ces règles devraient impliquer, au minimum, qu'une séquence s'arrête quand le temps de simulation maximum s'est écoulé ou bien lorsque qu'un état appartenant à un groupe d'états absorbants défini par l'utilisateur est atteint ou encore lorsque la fréquence de la séquence est inférieure à une limite inférieure spécifiée par l'utilisateur.

Les outils de quantification sont utilisés pour calculer les variables déterministes d'état mais aussi les fréquences des ramifications.

Des choix spécifiques pour chacun de ces cinq ensembles définissent une application particulière de la notion d'arbre d'événement dynamique.

L'analyse par arbre d'événements est une méthode qui permet d'examiner, à partir d'un événement initiateur, l'enchaînement des événements (les séquences d'événements) pouvant conduire ou non à un accident potentiel. Elle trouve ainsi une utilité toute particulière pour l'étude de l'architecture des moyens de sécurité (prévention, protection, intervention) existants ou pouvant être envisagés sur un site. A ce titre, elle peut être utilisée pour l'analyse d'accidents a posteriori. Cette méthode peut s'avérer rapidement lourde à mettre en œuvre. En conséquence, il faut définir avec discernement l'événement initiateur qui fera l'objet de cette analyse.

1.3.1.2 Arbres de défaillances (AdD)

Les arbres des défaillances sont un autre modèle booléen utilisé de manière fréquente dans les études de Sûreté de Fonctionnement. A l'origine ils ont été développée en 1962 aux BELL Laboratories par H.A. Watson, sur une demande de l'U.S. Air Force pour évaluer le Système de commande de Lancement du missile balistique intercontinental Minuteman7. L'utilisation des arbres de défaillance s'est très largement répandue chez les experts de fiabilité, notamment pour l'analyse de SdF d'applications industrielles critiques [Hanley et Kumamoto, 1981, Leveson, 1995].

Un arbre de défaillances (aussi appelé arbre de pannes ou arbre de fautes) est une technique déductive d'ingénierie très utilisée dans les études de sécurité et de fiabilité des systèmes statiques

(un système statique est un système dont la défaillance ne dépend pas de l'ordre de défaillance de ses composants). Cette méthode consiste à représenter graphiquement les combinaisons possibles d'événements qui provoquent l'occurrence d'un événement indésirable prédéfini appelé « événement redouté ». Une telle représentation graphique met donc en évidence les relations de cause à effet. Associer une probabilité d'occurrence aux événements de défaillance simples permet également de quantifier la probabilité d'occurrence d'un événement indésirable. En fonction des éléments constitutifs utilisés dans la construction d'un arbre (ses portes), on peut distinguer les arbres de défaillance et les arbres de défaillance dynamiques.

Arbres de défaillances

Le point de départ de la construction d'un arbre de défaillance est l'événement redouté lui-même (également appelé événement sommet). Il est essentiel qu'il soit unique et bien identifié. À partir de là, le principe est de définir des niveaux successifs d'événements tels que chacun est une conséquence d'un ou plusieurs événements du niveau inférieur. La démarche est la suivante : pour chaque événement d'un niveau donné, le but est d'identifier l'ensemble des événements nécessaires et suffisants à sa réalisation. Des opérateurs logiques (ou portes) permettent de définir précisément les liens entre les événements des différents niveaux.

Dans la construction des arbres de défaillance, les portes logiques utilisés sont ET, OU et K/N. La porte OU a comme signification le fait que l'événement en sortie survient si au moins un des événements en entrée est présent. Quant à la porte ET, l'événement en sortie survient seulement si tous les événements en entrée sont présents. La dernière porte utilisée est K/N et correspond au fait que l'événement en sortie survient si au moins K événements en entrée parmi N sont présents. Cette porte généralise les deux précédentes : une porte OU est une porte $1/N$ et une porte ET est une porte N/N.

Le processus déductif est poursuivi niveau par niveau jusqu'à ce que les spécialistes concernés ne jugent pas nécessaire de décomposer des événements en combinaisons d'événements de niveau inférieur, notamment parce qu'ils disposent d'une valeur de la probabilité d'occurrence de l'événement analysé. Ces événements non décomposés de l'arbre sont appelés événements élémentaires (ou événements de base).

La probabilité d'occurrence de l'événement redouté $\mathbb{P}(ER)$ est la probabilité qu'au moins une coupe minimale (un ensemble d'événements conduisant a l'événement redouté qui ne contient pas un sous-ensemble d'événements qui soit aussi une coupe) se produise. Pour calculer cette probabilité, le théorème de Sylvester-Poincaré peut être utilisé [Villemeur, 1997] :

$$\mathbb{P}(ER) = \sum_{i=1}^{n} \mathbb{P}(C_j) - \sum_{j=2}^{n} \sum_{i=1}^{j-2} \mathbb{P}(C_i \cap C_j) + ... + (-1)^n \cdot \mathbb{P}(C_i \cap C_j \cap ... \cap C_n) \tag{1.3}$$

où C_i représente une coupe minimale et n est le nombre des coupes minimales. Cette équation correspondant à une suite alternée décroissante, il est alors possible d'obtenir, par approximation, un encadrement de la probabilité de l'événement redouté $\mathbb{P}(ER)$:

$$\sum_{i=1}^{n} \mathbb{P}(C_j) - \sum_{j=2}^{n} \sum_{i=1}^{j-2} \mathbb{P}(C_i \cap C_j) \leq \mathbb{P}(ER) \leq \sum_{i=1}^{n} \mathbb{P}(C_j) \tag{1.4}$$

La probabilité d'une coupe étant égale au produit des probabilités de chacun de ses événements élémentaires $\mathbb{P}(e_{k_i})$, la probabilité d'occurrence de l'ER est donc borné par :

$$\mathbb{P}(ER) \leq \sum_{i=1}^{n} \prod_{k_i=k_1}^{k_i=k_m} \mathbb{P}(e_{k_i}) \tag{1.5}$$

Les arbres de défaillances permettent facilement de déterminer les coupes d'événements, de calculer leur probabilité d'occurrence et d'en déduire la probabilité d'occurrence de l'événement redouté. Une coupe (combinaison des événements de base) représente (contient) plusieurs séquences d'événements (en fonction de l'ordre d'occurrence des événements présents dans la coupe), mais l'arbre de défaillances ne permet ni d'identifier les séquences appartenant à une coupe, ni leur probabilité d'occurrence.

Une nouvelle algorithmique concernant les arbres de défaillance a été proposée par Coudert et Madre [Coudert et Madre, 1992,Coudert et Madre, 1994] et désignée sous le terme de *diagrammes de décision binaires* (l'équivalent en anglais est le terme *binary decision diagram* (BDD)). Les BDD sont des structures de données utilisées pour représenter des fonctions booléennes. Les algorithmes basés sur les BDD sont très rapides et ont permis une réduction très importante (de quelques heures ou même quelques jours à quelques secondes) pour le traitement de cas tests complexes et de grande taille. Ils se basent sur une réduction de l'arbre de factorisation de Shannon mise au point dans le traitement de fonctions booléennes par Akers [Akers, 1978], et plus tard par Bryant [Bryant, 1987, Bryant, 1992].

Partant de l'arbre développé de Shannon, sa réduction est effectuée des feuilles vers la racine en supprimant les nœuds inutiles. Un nœud peut-être supprimé dans les cas suivants :

- lorsque ses deux fils se dirigent vers le même nœud,
- lorsqu'il est équivalent à un autre nœud.

Deux nœuds sont dits équivalents s'ils portent la même variable et si leurs fils (1) et (0) se dirigent vers deux nœuds équivalents respectivement. Cette définition s'applique récursivement du bas vers le haut de l'arbre. Lorsque deux ou plusieurs nœuds sont équivalents, un seul est conservé et les autres seront supprimés en dirigeant leurs entrées vers le nœud conservé.

L'arbre de Shannon (développé ou réduit) n'est pas unique, il dépend de l'ordre de variables. Cependant à un arbre développé correspond un arbre réduit unique. C'est-à-dire que lorsque l'ordre des variables est fixé, le BDD obtenu est unique. Néanmoins, tous les BDD obtenus en adoptant différents ordres de variables sont équivalents et implicitement les expressions des fonctions booléennes le seront également. Le BDD peut être obtenu également d'une manière formelle sans passer par les graphes, le formalisme étant décrit en [Limnios, 2005]. Le calcul probabiliste qui sera effectué à partir de BDD donnera la même valeur dans tous les cas.

Étant donné qu'un inconvénient majeur de l'analyse par des arbres de défaillance est l'incapacité de prendre en considération l'ordre dans lequel les événements se produisent dans un système et implicitement l'impossibilité d'analyser les séquences d'événements, les arbres de défaillance dynamiques ont été introduits.

Arbres de défaillances dynamiques (AdDD)

Les arbres de défaillance dynamiques (AdDD) (ou Dynamic Fault Trees (DFT)) ont été introduits dans le milieu des années 1970 [Fussel *et al.*, 1976], et développés / implémentés dans plusieurs outils à la fin des années 1990.

Les AdDD représentent un modèle de fiabilité pour les systèmes dynamiques de composants binaires non réparables, initialement décrits par [Dugan *et al.*, 1992,Dugan *et al.*, 2000]. Basés sur les arbres de défaillances, ils permettent la représentation non plus d'un système statique, mais dynamique. Cette capacité à représenter la relation entre l'ordre d'occurrence des défaillances de

composants et la défaillance du système ou d'un sous-système provient de l'extension du modèle par l'ajout de portes dynamiques ($PAND$, $SPARE$, $FDEP$, SEQ) en plus des portes statiques existantes (ET, OU, K/N). La valeur de sortie de ces nouvelles portes n'est plus seulement fonction de l'ensemble des composants défaillants en entrée de celles-ci, mais également de l'ordre dans lequel ces défaillances sont produites.

La porte *priority-AND* ($PAND$) peut être considérée avec des événements e_1, ..., e_k comme entrées. La sortie de la porte est « vrai » si tous les événements e_1, ..., e_k se produisent exactement dans cet ordre [Rauzy, 2011].

La porte *Spare* ($SPARE$) a trois déclinaisons : Cold (CSP), Warm (WSP) and Hot (HSP). Ces portes sont utilisées pour modéliser un composant qui peut être remplacé dès qu'il tombe en panne par un composant de secours qui est disponible et qui a les mêmes fonctionnalités. Il est possible de considérer plusieurs composants de secours pouvant être sollicités dans un ordre prédéfini et partagés avec d'autres portes ($SPARE$). La "température" de la porte indique le type de vieillissement des composants de secours avant d'être actif : qui ne défaillent pas (cold), qui défaillent avec leur taux de défaillance (hot) et qui défaillent avec leur taux de défaillance réduite par le facteur de dormance (warm). La sortie de la porte est "vrai" quand toutes les entrées sont "vrai", c'est à dire que le composant principal et tous les composants de secours sont défaillants (ou indisponibles) [Aubry et Brînzei, 2015].

La porte $FDEP$ (*Functional-Dependency Gate*) est caractérisée par une entrée de déclenchement (représentant soit un événement de base, soit la sortie d'une autre porte dans l'arbre), une sortie non-dépendent (reflétant l'état de l'événement de déclenchement) et un ou plusieurs événements de base dépendants.

Une dernière porte présentée ici est SEQ (*Sequence-Enforcing Gate*) ; les événements d'entrée sont contraints à se produire dans l'ordre de gauche à droite dans lequel ils apparaissent sous la porte. Cette porte peut être comparée avec la porte $PAND$ par le fait que $PAND$ détecte si les événements se produisent dans un ordre particulier, alors que la porte *sequence enforcing gate* permet que les événements se produisent seulement dans un ordre spécifié.

Dans l'exemple de la figure (1.4) on a un arbre de défaillance dynamique avec ER représentant l'événement redouté, e_1, e_2, e_3 sont les événements de base et la porte $PAND$ signifie qu'à sa sortie on a l'événement « vrai » si deux événements se produisent, mais seulement si le premier (e_2) apparaît avant le deuxième (e_3).

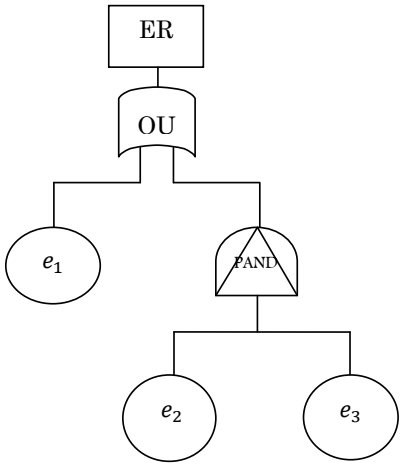


FIGURE 1.4 – Arbre de défaillances dynamique

Dans la littérature scientifique traitant des AdDD plusieurs sémantiques (interprétations) des portes nouvellement introduites existent [Cepin et Mavko, 2002]. En fonction de ces sémantiques, les formules de calcul de la probabilité d'occurrence de ces portes sont différentes les unes des autres, dans certains cas une résolution analytique étant possible, sinon on fait appel à la simulation de Monte-Carlo. Une sémantique assez habituelle pour la porte $PAND$, permet de calculer la probabilité de l'événement redouté de l'AdDD de la figure (1.4) par l'expression suivante :

$$\mathbb{P}(ER) = \mathbb{P}(e_1) + \mathbb{P}(e_3 \mid e_2)$$

Par la définition et l'introduction des nouvelles portes, un AdDD permet de modéliser certains types de séquences d'événements (en fonction des portes utilisées dans leur construction) et de calculer leurs probabilités d'occurrence, notamment pour des systèmes constitués de composants binaires non-réparables. Néanmoins, les séquences d'événements relatives à des systèmes réparables ne peuvent pas être modélisées et évaluées par des arbres de défaillances dynamiques. Pour combler ce manque et permettre la modélisation et l'évaluation de systèmes possédant des politiques de réparation complexes, les arbres de défaillance réparables (ou Repairable Fault Tree) [Flammini, 2006] ont été introduits. Ils préservent la simplicité de la modélisation d'AdD et se base sur l'expressivité des réseaux de Petri pour modéliser les politiques de réparation (sans se focaliser sur des séquences d'événements). Cette approche, consistant à enrichir un modèle booléen basé sur les arbres de défaillances par un modèle de type états-transitions pour prendre en compte des politiques de réparation ou de reconfiguration, est également à l'origine du développement des BDMP couplant arbres de défaillances et chaînes de Markov.

1.3.1.3 Boolean logic Driven Markov Process (BDMP)

Pour modéliser et analyser les systèmes réparables en prenant en compte les séquences d'événements, un nouveau type de modèle a été défini. Il s'agit des Boolean logic Driven Markov Process (BDMP) définis par M. Bouissou dans [Bouissou et Bon, 2003]. Partant du modèle de l'arbre de défaillances [Bouissou, 2008], on peut donner le principe simplifié du formalisme des BDMP en disant qu'il remplace :

- les modèles simples des événements de base (ou feuilles) d'un arbre de défaillances, par des chaînes de Markov. Les états de ces chaînes de Markov sont classés en deux catégories (marche ou panne). Suivant la catégorie à laquelle appartient l'état d'une feuille à un instant donné, « l'événement » correspondant à cette feuille est considéré comme *vrai* ou *faux*.
- l'indépendance totale des feuilles d'un arbre de défaillances par *des dépendances simples*. Chaque feuille a deux modes « sollicité » et « non sollicité », correspondant à deux chaînes de Markov différentes. Le choix du mode dans lequel une feuille se trouve à un instant donné est déterminé par la valeur (*vrai* ou *faux*) d'un sous-ensemble de feuilles. Les transitions entre ces deux modes définissent éventuellement des états instantanés dans lesquels on peut déclencher des transitions instantanées probabilisées. Le choix du mode d'une feuille donnée en fonction de l'état d'un sous-ensemble de feuilles et représenté par « *une gâchette* » qui active la feuille donnée lorsque le sous-ensemble des autres feuilles a la valeur *vrai*.

Un BDMP sans gâchettes équivaut à un arbre de défaillances. Dans un BDMP avec des gâchettes, les défaillances des composants ne sont pas toutes possibles dans l'état initial : seules celles des feuilles sollicitées le sont.

La figure (1.5) présente un exemple de BDMP qui aide à mieux comprendre ce formalisme. Dans cet exemple, ER signifie l'événement redouté (qui représente la défaillance du système)

et e_1, e_2, e_3 représentent les comportements dysfonctionnels des composants du système. Un premier niveau de redondance est présent entre e_1 et le sous-système représenté par la porte ET ; il est représenté par la gâchette. Tant que l'événement e_1 ne se produit pas, le deuxième sous-système (e_2 et e_3) n'est pas sollicité (il est dormant). Lorsque l'événement e_1 se produit (défaillance du premier composant), il active par la gâchette le deuxième sous-système (la porte ET entre e_2 et e_3). Une deuxième redondance est présente entre e_2 et e_3. A chaque feuille du BDMP (e_1, e_2, e_3) une chaîne de Markov sous-jacente est associée pour décrire le comportement fonctionnel et/ou dysfonctionnel du composant.

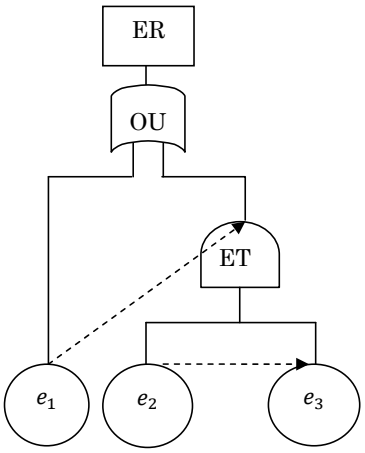


FIGURE 1.5 – Exemple de BDMP

Les BDMP réunissent ainsi, dans le même paradigme, des concepts issus des arbres de défaillances et des chaînes de Markov. Cette idée d'association était déjà incluse dans les arbres de disponibilité [Villemeur, 1988], où les événements de défaillance de base dans les arbres de défaillance ont été remplacés par l'événement d'indisponibilité, mais en supposant que chaque composant peut être réparé à n'importe quel moment. En supprimant cette restriction, les BDMP permettent de modéliser un grand nombre de redondances, des politiques de réparation, des modes de défaillance multiples, *etc.* et facilitent la construction et la résolution des modèles markoviens de grande taille.

En termes d'évaluation quantitative, le modèle sous-jacent au BDMP n'est plus une équation booléenne, mais une chaîne de Markov, voir une chaîne de Markov multi-phases dans le cas le plus général avec des gâchettes. Par conséquent, la probabilité d'occurrence de l'événement redouté ne pourra plus être calculée par une formule simple (sommes et/ou produits des probabilités des événements de base) mais nécessitera de faire appel à la théorie de chaînes de Markov qui sera présentée par la suite. De même, la détermination des séquences d'événements et le calcul de leurs probabilités d'occurrence devront faire appel aux modèles de type états-transitions.

Les BDMP constituent donc un outil efficace pour les études de SdF, proche des arbres de défaillance qui peuvent être considérés comme un sous ensemble de BDMP [Bouissou, 2007, Chaux, 2013]. Sous certaines hypothèses, le langage *Figaro* [Bouissou *et al.*, 1991,Bouissou, 1993] sert à transformer un BDMP dans un processus de Markov en temps continu, donnant ainsi accès à l'évaluation quantitative. Les BDMP sont implémentés dans un ensemble d'outils développés par EDF (*KB3*) ; un grand nombre des études ont été réalisées avec ces outils, particulièrement dans le domaine des systèmes électriques réparables. Néanmoins, les gâchettes ne considèrent qu'une stratégie de reconfiguration simple (commutation vers un mode de secours sur condition

booléenne et retour dès que possible). Ceci a justifié l'extension proposée par Piriou [Piriou, 2015] (GBDMP - Generalized BDMP) qui introduit des machines de Moore pour représenter des stratégies de reconfiguration complexe. En revanche, cette extension ne permet pas l'évaluation quantitative des indicateurs de SdF et nécessite une transformation des modèles vers des chaînes de Markov.

1.3.2 Modèles états-transitions

Ce type de modèle a comme caractéristique principale de prendre en compte les différents états du système et de ses configurations (fonctionnel, dégradé, défaillant, *etc.*) ainsi que les transitions qui les relient. Les évolutions entre états sont provoquées par l'occurrence d'événements associés aux transitions. Cette section présente d'abord un modèle déterministe permettant d'extraire les séquences d'événements mais pas de les quantifier (automates à états finis et théorie des langages) puis quatre modèles probabilistes (chaînes de Markov, automates stochastiques hybrides, réseaux de Petri stochastiques et langages probabilistes).

1.3.2.1 Automates à états finis (AEF) & Théorie des langages

La notion de langage [Cassandras et Lafortune, 2008] est utilisée pour décrire l'ensemble des séquences d'occurrence d'événements (on dit séquences d'événements pour simplifier) possibles dans un système à événements discrets (SED). Cette notion permettra de définir également les automates à états finis qui en sont une représentation condensée.

Chaque événement devant être pris en compte dans l'étude d'un système est considéré comme un symbole faisant partie d'un ensemble fini appelé *alphabet* Σ. Sur cet *alphabet*, des *mots* sont définis pour représenter les séquences ordonnées d'événements qui décrivent les différentes trajectoires possibles du système. Un *langage* est un sous-ensemble de mots formé sur l'alphabet Σ et représente ainsi un sous-ensemble de séquences d'événements ayant les mêmes caractéristiques (par exemple, toutes les séquences d'événements amenant à un état de panne). Plusieurs types d'opérations (réunion, produit, concaténation, itération ou fermeture de Kleene, clôture en préfixe, *etc.*) sont définis [Cassandras et Lafortune, 2008] sur les langages déterminés par l'alphabet Σ, incluant aussi le mot vide ε toujours présent dans le système. Ainsi, l'ensemble de toutes les séquences finies définies sur Σ (incluant le mot ε) est représenté par la fermeture de Kleene de tout l'alphabet Σ^*.

Définition 1. *Un automate fini A est un quintuplet $A = (X, \Sigma, f, x_0, X_f)$ où :*

- *X est un ensemble fini d'états ;*
- *Σ est un ensemble fini d'événements appelé alphabet ;*
- *$f : X \times \Sigma \to X$ est la fonction de transition qui associe à chaque état de départ et à chaque événement un état d'arrivée ;*
- *x_0 est l'état initial ;*
- *$X_f \subseteq X$, un sous-ensemble d'états de X, identifiant les états terminaux ou marqués.*

Une séquence s est reconnue par un automate A si la fonction de transition étendue $f(x_0, s)$ est définie. L'ensemble des séquences reconnues par un automate forme le langage généré par cet automate. La séquence s appartiendra au langage marqué si $f(x_0, s) \in X_f$. Les états, définis comme « marqués » ou « terminaux », représentent des états spécifiques (par exemple, état de panne, mais aussi un état pouvant représenter la fin de l'exécution d'une tâche demandée au système). De manière plus générale, un état du système est défini comme « terminal » par

rapport à l'objectif de l'étude (un état défaillant dans une étude de fiabilité ou d'indisponibilité, un état non défaillant dans une étude de disponibilité ou de maintenabilité, *etc.*).

Le mémoire de thèse de P. Y. Chaux [Chaux, 2013] propose un cadre formel basé sur la théorie des langages et les automates à états finis pour l'identification de séquences d'événements critiques, pour des systèmes cohérents, à partir d'un noyau du langage défaillant et d'un ensemble de séquences minimales. Ce cadre formel repose sur :

- la définition de la cohérence des systèmes dynamiques et réparables à partir des langages générés et marqués par l'automate fini,
- 5 règles de cohérence qui sont utilisées ensuite pour définir le noyau du langage défaillant et les séquences de coupe minimales,
- la définition du noyau du langage défaillant et des séquences de coupe minimales comme représentations minimales respectivement du langage défaillant et de l'ensemble des séquences de coupe.

L'hypothèse principale retenue dans cette étude est que les scénarios d'événements de défaillance et de réparation décrits par un modèle représentant la défaillance globale du système peuvent toujours être représentés à l'aide d'un automate à états finis. Tous les scénarios (les séquences) d'événements existant dans un système forment alors un langage qui est reconnu par l'automate qui représente le système.

L'identification des séquences repose sur la construction de plusieurs langages dont les relations d'inclusion sont présentés schématiquement sur la figure (1.6) :

- Le langage $\mathcal{L}_D$ est nommé dysfonctionnel et les séquences décrivent tous les scénarios de défaillance et de réparation possibles dans le système. Dans le cas de systèmes réparables, ce langage est de taille infinie.
- $\mathcal{L}_F$ est le langage défaillant qui décrit, parmi les séquences du langage défaillant, celles qui terminent dans un état où le système est défaillant. Comme pour le langage dysfonctionnel $\mathcal{L}_D$, pour les systèmes réparables ce langage est de taille infinie.
- Les langages $\mathcal{L}_D$ et $\mathcal{L}_F$ sont nécessaires à la définition de la fonction de structure dynamique, ils permettent alors de définir l'état de panne du système en fonction d'événements de défaillance et de réparation de ses composants.
- Dans le langage défaillant, le langage $\mathcal{L}_{SC}$ représente le sous ensemble des séquences de coupe. Pour les systèmes réparables et dynamiques, ce langage est de taille infinie. Les séquences de coupe représentent les scénarios qui permettent d'arriver à la première défaillance du système depuis l'état initial.
- $\mathcal{L}_F^{NL}$ est la réduction du langage défaillant $\mathcal{L}_F$ au langage fini qui ne contient que les séquences ne passant pas deux fois par un même état.
- $Kern\,(\mathcal{L}_F)$ est l'ensemble fini des séquences du langage défaillant permettant la représentation la plus compacte de tous les scénarios du langage défaillant d'un système cohérent. Il permet de reconstruire l'ensemble du langage défaillant.
- SCM représente l'ensemble des séquences de coupe minimales qui est un sous ensemble de $Kern\,(\mathcal{L}_F)$ et qui permet la reconstruction au minimum de l'ensemble de toutes les séquences de coupes $\mathcal{L}_{SC}$. Étant donnée la définition du noyau du langage défaillant, l'ensemble des séquences de coupe minimales est l'ensemble des séquences du noyau du langage défaillant qui permet de générer l'ensemble des séquences défaillantes dont le seul état marqué traversé (représentant la défaillance du système) est celui atteint en fin de séquence [Chaux, 2013].

La figure (1.7) présente un exemple d'automate à états finis possédant trois états (*Etat* 1, *Etat* 2 et *Etat* 3) et un alphabet comprenant 4 événements (e_1, e_2, e_3, e_4). Si l'on considère que

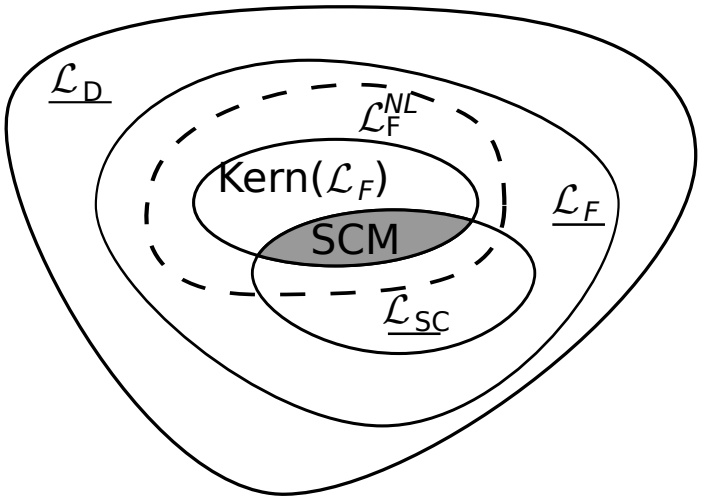


FIGURE 1.6 – Relations d'inclusion entre les différents langages et ensembles de séquences

l'*Etat* 3 de notre figure est un état absorbant représentant la défaillance du système, le noyau du langage défaillant $Kern(\mathcal{L}_F)$ ne sera constitué que d'une seule séquence e_1e_2 et l'ensemble des séquences de coupe minimales SCM contiendra également cette seule séquence.

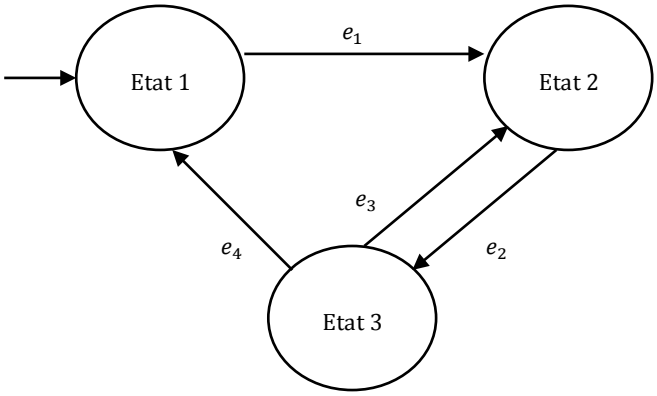


FIGURE 1.7 – Exemple d'automate à états finis

En partant d'une description du comportement d'un système par un modèle de type automate à états finis, cette approche permet donc de déterminer le noyau du langage défaillant et l'ensemble des séquences de coupes minimales, relatif à un ou plusieurs états de défaillance du système qui sont considérés comme des états terminaux ou absorbants. A noter que cette approche peut également s'appliquer aux BDMP dans la mesure où [Chaux, 2013] propose également une méthode pour obtenir systématiquement un automate à états finis « équivalent » à un BDMP. Elle est donc très intéressante pour en phase d'identification des séquences d'événements, mais ne permet pas, en l'état actuel, de réaliser une évaluation quantitative probabiliste de ces séquences. Pour réaliser une telle évaluation, les modèles états-transitions doivent être enrichis avec un aspect temporisé et stochastique.

1.3.2.2 Chaînes de Markov (CdM)

Les chaînes de Markov sont des modèles états-transitions qui servent à la modélisation comportementale d'un système dont la principale caractéristique est que la connaissance de l'état présent est suffisante pour déterminer l'évolution future du système (cette caractéristique est appelée également « sans-mémoire »). Le mathématicien russe Andreï Markov a publié les premiers résultats sur les chaînes de Markov à espace d'états fini en 1906. Une généralisation à un espace d'états infini dénombrable a été publiée par Kolmogorov en 1936.

Un processus markovien s'appelle **chaîne de Markov à temps discret** si l'intervalle d'observation T est un ensemble discret et il s'appelle **chaîne de Markov à temps continu** si l'intervalle d'observation T est un ensemble continu. Dans les études de sûreté de fonctionnement, on utilise en général des chaînes de Markov à temps continu car elles modélisent naturellement l'évolution du système dans le temps. Dans une chaîne de Markov à temps continu, les transitions stochastiques entre les états sont caractérisées par des taux de défaillance/réparation. La figure (1.8) donne un exemple de chaîne de Markov à temps continu à trois états et des taux de défaillance associés aux transitions λ_{12}, λ_{23}, λ_{32} et λ_{31}.

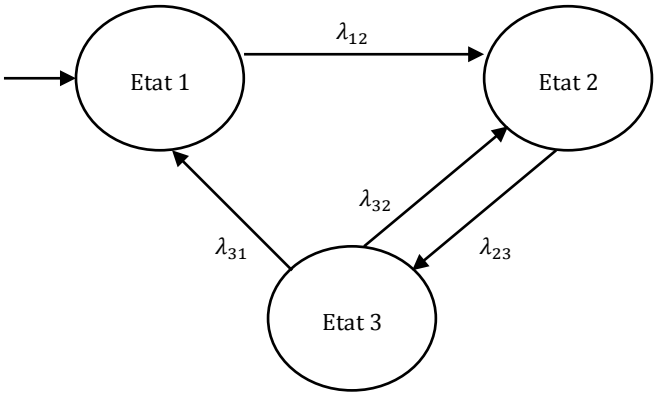


FIGURE 1.8 – Exemple de chaîne de Markov à temps continu

Les probabilités que le système se trouve dans un de ses états sont données par le vecteur de probabilités suivant :

$$\mathbb{P}(t) = [\mathbb{P}_1(t)\,\mathbb{P}_2(t)\,...\mathbb{P}_n(t)],\ t \in T \tag{1.6}$$

Ce vecteur des probabilités d'état est un vecteur stochastique :

$$\sum_{i=1}^{n} \mathbb{P}_i = 1 \tag{1.7}$$

Une chaîne de Markov est décrite complètement par son générateur infinitésimal (matrice des taux de transition) :

$$M = [\lambda_{ij}] \tag{1.8}$$

où λ_{ij} représente le taux de transition de l'état x_i vers l'état x_j. Le générateur infinitésimal d'une chaîne de Markov à temps continu possède les propriétés suivantes :

$$\lambda_{ij} > 0\ \forall i, j = 1, 2, ..., n \tag{1.9}$$

$$\sum_{j=1}^{n} \lambda_{ij} = 0 \tag{1.10}$$

Le vecteur des probabilités d'état d'une chaîne de Markov à temps continu est donné par la relation suivante :

$$\dot{\mathbb{P}}(t) = \mathbb{P}(t) \cdot M \tag{1.11}$$

où $\dot{\mathbb{P}}(t)$ représente la dérivée de $\mathbb{P}(t)$ par rapport au temps.

L'équation (1.11) porte le nom d'équation fondamentale de la chaîne de Markov à temps continu et permet de déterminer le vecteur des probabilités d'état, quel que soit l'instant t en connaissant le vecteur initial des probabilités d'état $\mathbb{P}(0)$:

$$\mathbb{P}(t) = \mathbb{P}(0) \cdot e^{Mt} \tag{1.12}$$

Une chaîne de Markov s'appelle **chaîne ergodique** si dans son comportement asymptotique le système tend vers une distribution limite unique, indépendante des conditions initiales :

$$\pi = \mathbb{P}(\infty) = \lim_{t \to \infty} \mathbb{P}(t) \tag{1.13}$$

Le vecteur π représente la distribution de probabilités d'états en régime permanent et est appelé *distribution stationnaire des probabilités.*

Pour une chaîne de Markov à temps continu ergodique, on peut définir *une chaîne de Markov immergée à temps discret.* La probabilité de transition entre deux états x_i et x_j de la chaîne de Markov immergée est donnée par le ratio entre le taux de défaillance/réparation associé à l'événement (ou transition) e_{ij} et la somme de tous les taux de défaillances/réparations associés aux événements qui ont l'état x_i comme état de départ.

$$\mathbb{P}(e_{ij}) = p_{ij} = {}^{\lambda_{ij}}/_{\sum_{j \neq i}(\lambda_{ij})} \tag{1.14}$$

La relation 1.14 permet de définir *la matrice des probabilités de transitions* suivante :

$$M_{im} = [p_{ij}] \tag{1.15}$$

qui est une matrice stochastique :

$$0 \leq p_{ij} \leq 1 \ \forall i, j = 1, 2, ..., n \tag{1.16}$$

$$\sum_{j=1}^{n} p_{ij} = 1 \tag{1.17}$$

L'équation de Chapman-Kolmogorov pour la chaîne de Markov immergée devient :

$$\pi_{im} = \pi_{im} \cdot M_{im} \tag{1.18}$$

La relation 1.18 permet de déterminer la distribution stationnaire π_{im}, en tenant compte que :

$$\pi_{im} \cdot \overline{1} = 1 \tag{1.19}$$

où $\overline{1}$ est un vecteur de 1 uniquement.

On note que la distribution stationnaire des probabilités d'états correspondant à une chaîne de Markov immergée est unique et cela est dû au fait que la chaîne de Markov est ergodique et est représentée par un graphe fortement connexe. De plus, la distribution stationnaire des

probabilités d'états de la chaîne de Markov immergée permet ensuite d'obtenir la distribution stationnaire des probabilités d'états de la chaîne de Markov à temps continu :

$$\pi_i = \frac{\pi_{im_i} \cdot \eta_i}{\sum_{j=1}^{n} \pi_{im_j} \cdot \eta_j} \tag{1.20}$$

où $\eta_i = {}^{1}/_{-\lambda_{ii}}$ représente le temps de séjour moyen dans l'état x_i de la chaîne de Markov à temps continu.

En analysant différents systèmes réels, on remarque que les transitions entre états ne sont pas toutes nécessairement stochastiques. Dans certains cas, elles peuvent avoir une durée constante de temps et sont donc considérées comme déterministes car n'étant plus caractérisées par une probabilité d'occurrence (*e.g.* les transitions représentant la détection de défaillance des différents composants du système). Pour modéliser ce type de systèmes où les temps de passage d'un état à l'autre peuvent suivre n'importe quelle loi sur $\mathbb{R}_+$, la théorie des **processus semi-markoviens** (qui sont une généralisation naturelle des processus markoviens) a été introduite par P. Levy et W. L. Smith dans les années 1954 [Iosifescu *et al.*, 2007].

Dans le cadre des processus semi-markoviens [Howard, 1971a,V. S. Barbu, 2008], la probabilité conditionnelle, qui représente la probabilité de franchir une transition de l'état x_i vers l'état x_j dans l'intervalle de temps $[0, t]$, est notée $Q_{ij}(t)$ et la matrice $Q(t) = [Q_{ij}(t)]$ est appelée noyau.

La matrice des probabilités de transition de la chaîne de Markov *immergée*, notée avec $M_{im} = [p_{ij}]$, est donnée par :

$$p_{ij} = Q_{ij}(\infty) \tag{1.21}$$

La *durée moyenne conditionnelle de passage de l'état x_i vers l'état x_j* est donné par la relation suivante :

$$\eta_{ij} = \int_0^{\infty} t \, \mathrm{dT_{ij}}(t) \tag{1.22}$$

où T_{ij} est *la fonction de répartition du temps de passage du processus de l'état x_i vers l'état x_j* :

$$T_{ij}(t) = \frac{Q_{ij}(t)}{p_{ij}} \tag{1.23}$$

La durée moyenne de séjour dans l'état x_i d'un processus semi-markovien est donnée par :

$$\eta_i = \int_0^{\infty} t \, \mathrm{d}\left(\sum\nolimits_{\mathrm{j}} \mathrm{Q_{ij}}(\mathrm{t})\right) = \sum\nolimits_j p_{ij} \int_0^{\infty} t \, \mathrm{dT_{ij}}(t) \tag{1.24}$$

Concernant l'évaluation des séquences d'événements, la probabilité d'occurrence d'une séquence ne peut être obtenue que si l'état atteint après l'occurrence de la séquence est un état absorbant et si la séquence d'événements considérée est la seule séquence qui amène le système depuis l'état initial dans cet état absorbant. Dans ce cas particulier, la probabilité d'occurrence de la séquence correspond à la probabilité de l'état absorbant obtenue par l'équation de Chapman-Kolmogorov.

1.3.2.3 Automates Stochastiques Hybrides (ASH)

Les Automates Stochastique Hybrides (ASH) ont été définis de manière formelle dans les travaux de Castaneda *et al.* [Perez-Castaneda, 2009, Perez-Castaneda *et al.*, 2011] avec un objectif de modélisation et d'évaluation probabiliste de la sûreté de fonctionnement des systèmes complexes. En particulier, lorsque la fonction de structure se révèle être un langage d'événements (séquences plutôt que coupes), il devient nécessaire d'utiliser un formalisme de représentation

du comportement du type états-transitions. C'est le caractère premier du « dynamisme » de la fonction de structure d'un système [Aubry *et al.*, 2012b], cet aspect étant souvent caché dans le concept de *fiabilité dynamique,* qui regroupe un ensemble de propriétés, telles que :

- l'existence d'interactions dynamiques entre les paramètres physiques du processus (représentées généralement par des variables continues dont l'évolution est décrite par des équations algébro-différentielles) et le comportement nominal ou dysfonctionnel des composants (représenté généralement par l'occurrence d'événements) ;
- le caractère déterministe ou stochastique des événements et des variables physiques ;
- la multiplicité des modes de vieillissement des composants : vieillissement en fonction du temps, mais aussi en fonction des variables continues précédentes ou de l'occurrence des événements (comme une accumulation d'événements de sollicitation) ;
- des modèles non binaires du comportement des composants (des lois de probabilité différentes peuvent être associées à la défaillance ou à la réparation d'un même composant selon l'état dans lequel se trouve le système, par exemple, le vieillissement d'un composant peut dépendre de son mode de sollicitation, de sa stratégie de réparation ou du niveau de danger de l'état présent) ;
- les changements d'états discrets sont associés aux instants et à l'ordre d'occurrence des événements liés aussi bien aux défaillances des composants qu'aux franchissements de seuils des variables continues.

Un ASH est un automate à états finis temporisé, chaque état discret (mode de fonctionnement ou dysfonctionnement) ayant été caractérisé par des équations différentielles représentant l'évolution des variables continues modélisant les processus physiques spécifiques à chaque mode. Les transitions entre les différents modes de fonctionnement se font sur l'occurrence d'événements déterministes ou stochastiques. Les premiers représentent le franchissement de seuils des variables continues, les seconds représentent les défaillances des composants. Les ASH sont adaptés pour la modélisation des systèmes dynamiques hybrides présentant un couplage fort entre des variables stochastiques représentant le caractère dysfonctionnel d'un système et des variables déterministes représentant le processus physique.

Un exemple d'ASH est représenté dans la figure (1.9). A chaque état discret $\mathcal{X}^i$ est associée une équation différentielle $\dot{x} = f_i(x,t)$ décrivant l'évolution des variables continues x dans l'état discret $\mathcal{X}^i$. Un arc $\mathcal{A}_i$ représentant une transition est défini par deux états discrets représentant l'état de départ et l'état cible de la transition, par un événement e_i, une condition de garde G_i (représentant le franchissement d'un seuil par la variable continue lorsqu'il s'agit d'une transition déterministe) et une fonction de réinitialisation R_i (représentant la fonction de réinitialisation des variables continues x dans le nouvel état cible de la transition). Lorsque les transitions sont franchies sur occurrence des événements stochastiques, ces événements sont caractérisés par un taux d'occurrence (taux de défaillance ou de réparation). S'il est possible de quitter un état donné, sur occurrence du même événement, pour atteindre deux états différents (par exemple les états $\mathcal{X}^3$ ou $\mathcal{X}^4$ atteint depuis l'état $\mathcal{X}^2$ sur occurrence de e_2), une distribution de probabilités discrètes est associée à ces transitions en conflit. Cette distribution peut, par exemple, modéliser la détection ou non de la défaillance d'un composant : l'événement e_2 représente cette défaillance et si cette défaillance est non-détectée (ce qui peut se produire avec une probabilité p_2^3) le système arrive dans l'état dangereux $\mathcal{X}^3$. Si la défaillance est détectée (ce qui peut se produire avec une probabilité $1 - p_2^3$), le système arrive dans l'état $\mathcal{X}^4$ où la réparation est effectuée et elle ramène le système dans l'état $\mathcal{X}^1$.

Les ASH ont été implémentés dans l'environnement de simulation Scilab/Scicos qui est une alternative (libre) aux logiciels commerciaux pour la modélisation et la simulation des systèmes dynamiques tels que Matlab/Simulink et MATRIX/SystemBuild.

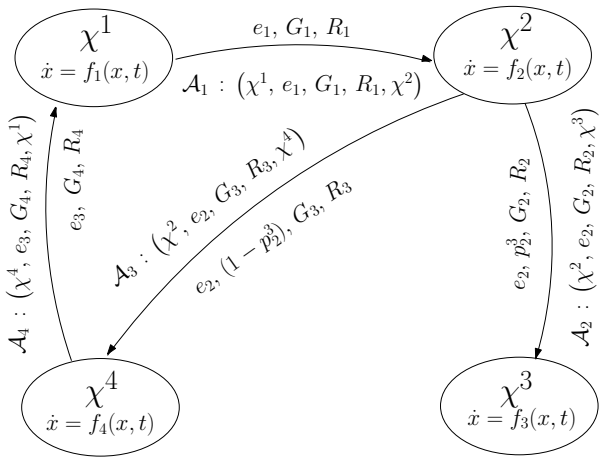


FIGURE 1.9 – Exemple d'Automate Stochastique Hybride

Le modèle ASH a été défini afin d'accéder à l'évaluation des grandeurs de la sûreté de fonctionnement par le biais de la simulation de Monte-Carlo. En plus de l'obtention des grandeurs classiques de la sûreté de fonctionnement (fiabilité, disponibilité, MTTF, MTBF, MTTR, *etc.*) [Perez-Castaneda, 2009, Perez-Castaneda *et al.*, 2011], le modèle ASH a permis d'identifier les séquences d'événements et de calculer leur probabilité d'occurrence [Aub 2012b, Bab 2012] à partir des résultats de la simulation de Monte-Carlo.

L'identification des séquences d'événements et le calcul de leur probabilité d'occurrence peuvent se réaliser en utilisant le modèle ASH. Cependant, les aspects hybrides limitant considérablement les capacités de calcul analytique, l'exploitation des ASH passe très souvent par la simulation de Monte-Carlo. Les séquences d'événements ne sont donc généralement pas obtenues de manière formelle.

1.3.2.4 Réseaux de Petri Stochastiques

Les réseaux de Petri (RdP) sont également des modèles états-transitions mais ont un pouvoir d'expression plus riche que les automates à états finis [Murata, 1989, Cassandras et Lafortune, 1999,Best *et al.*, 2001]. 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 a priori l'ensemble de ses états. Ils peuvent être utilisés à la fois pour des analyses qualitatives (vérification de propriétés) ou pour des analyses quantitatives, comme l'évaluation de performances fonctionnelles ou de SdF.

Les *réseaux de Petri stochastiques* [Chiola *et al.*, 1993a,Chiola *et al.*, 1993b,Haas, 2010,Aubry et Brinzei, 2016] sont une classe de réseaux de Petri dans lesquels les temps de franchissement des transitions sont générés par des variables aléatoires de distributions quelconques à support dans $[0, \infty[$. La définition d'un réseau de Petri stochastique suppose un ensemble fini de places, un ensemble fini de transitions, deux matrices d'incidence (avant et après), le marquage initial et les taux associés aux transitions. Les taux associés aux transitions sont indépendants du marquage et il n'est pas possible d'avoir de transitions multiples au même instant. 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. Le marquage d'un RdP stochastique est un

vecteur aléatoire dont les composantes sont les variables aléatoires qui représentent les marquages des places. Grâce au concept de marquage, le modèle RdP supporte la notion de mémoire ce qui constitue un avantage par rapport au modèle markovien qui est sans mémoire par hypothèse.

Les évolutions de marquage d'un RdP stochastique sont déterminées de la manière suivante : si, pour un marquage donné, plusieurs transitions peuvent être validées, un tirage aléatoire de la durée associée (inverse du taux de franchissement) est réalisée pour chacune d'entre elles ; au bout d'un temps égal à la plus courte de ces durées, la transition correspondante est franchie et aboutit à un nouveau marquage. La même procédure est appliquée sur ce nouveau marquage et ainsi de suite.

L'utilisation de réseaux de Petri stochastiques permet de modéliser des tâches avec des temps d'exécution non déterministes, de modéliser et de spécifier le comportement du système étudié, de prendre en compte des pannes aléatoires et de valider le modèle. D'un point de vue théorique, ils permettent également l'évaluation quantitative des séquences d'événements dans les études de SdF. En revanche, dans la pratique, l'identification de ces séquences reste une tâche ardue dans la mesure où elle suppose de construire le graphe de marquage complet. Très souvent, cette hypothèse ne peut être satisfaite en raison de la complexité des phénomènes à modéliser et impose donc, comme pour les ASH, le recours à la simulation de Monte-Carlo.

1.3.2.5 Langages Probabilistes

La théorie des langages probabilistes est une extension de la théorie des langages rationnels (ou réguliers), développée dans les travaux de Garg, Kumar et Marcus [Garg, 1992, Kumar *et al.*, 1996, Garg *et al.*, 1999, Kumar et Garg, 2001], pour modéliser le comportement des systèmes à événements discrets (SED) stochastiques.

Dans le cadre de la théorie des langages, les langages rationnels sont des langages pouvant être décrits par des expressions régulières et des automates finis [Habrard, 2004]. Pour les SED stochastiques, les langages probabilistes étendent les définitions relatives aux langages rationnels des SED déterministes en associant une mesure de probabilité à chaque séquence appartenant à l'ensemble de toutes les séquences Σ^*. Cette mesure représente la probabilité d'occurrence de la séquence.

Afin de simplifier, du point de vue mathématique, la définition formelle d'un langage probabiliste, un événement particulier, appelé « événement de terminaison » noté e_Δ, a été introduit pour représenter le fait que l'état du système obtenu après l'occurrence d'une séquence est un état terminal pour l'étude. Ainsi, le comportement du système est donné par l'ensemble Ω de toutes les séquences de longueur finie suivies ou non par l'événement de terminaison :

$$\Omega = \Sigma^* \left(e_\Delta + \epsilon\right) = \Sigma^* e_\Delta \cup \Sigma^* \tag{1.25}$$

où ϵ est le mot vide, utilisé pour définir de manière formelle les langages probabilistes.

Pour déterminer la probabilité d'occurrence d'une séquence d'événements s, il faut considérer non pas seulement l'occurrence indépendante de cette séquence s, mais aussi l'occurrence de toutes les séquences qui démarrent par s (toutes les séquences ayant le préfixe s). L'ensemble de toutes les séquences ayant une séquence donnée s comme préfixe est donné par :

$$\langle s \rangle = \{st | st \in \Omega\} \tag{1.26}$$

Définition 2. *[Garg* et al.*, 1999] Soit l'espace mesurable* (Ω, F) *où* $\Omega \subseteq \Sigma^* \left(e_\Delta + \epsilon\right)$ *et* F *est une* σ*-algèbre*[1] *générée par* $\{\langle s \rangle | s \in \Omega\}$*. Alors un langage probabiliste (ou p-langage)* $\mathbb{L}$ *est une mesure de probabilité sur l'espace mesurable* (Ω, F)*.*

1. En mathématiques, un σ*-algèbre* ou *tribu* sur un ensemble X est un ensemble non vide de parties de X, stable

Ainsi, la probabilité de terminaison d'une séquence s (probabilité que l'évolution d'un système se termine après l'occurrence de s) est donnée par la relation suivante :

$$\mathbb{P}\left(se_{\Delta}\right) = \mathbb{P}\left(\langle s \rangle\right) - \Sigma_{e \in \Sigma} \mathbb{P}\left(se\right), \forall s \in \Sigma^{*} \tag{1.27}$$

où $\mathbb{P}\left(\langle s \rangle\right)$ représente la probabilité d'occurrence de toutes les séquences ayant s comme préfixe et $\mathbb{P}\left(se\right)$ représente la probabilité de toutes les séquences ayant se comme préfixe (c'est-à-dire que le système continue à évoluer au-delà de se).

De manière similaire aux représentations des langages rationnels par des automates finis dans le cas déterministe, il est possible d'associer un automate probabiliste (noté p-automate) à un langage probabiliste.

Définition 3. *Un automate probabiliste (ou p-automate) sur un alphabet (ensemble d'événements) Σ est défini par le quintuple suivant :*

$$A_p = (X, \Sigma, f, \mathbb{P}, x_0) \tag{1.28}$$

où :

- X est un ensemble fini d'états ;
- Σ est un ensemble fini d'événements appelé alphabet ;
- $f : X \times \Sigma \rightarrow X$ est la fonction de transition qui associe à chaque état de départ et à chaque événement un état d'arrivée ;
- $\mathbb{P} : X \times \Sigma \times X \rightarrow [0, 1]$ est la fonction de probabilité de transition affectant à chaque transition une probabilité d'occurrence qui vérifie la relation suivante :

$$\Sigma_{x_j \in X} \Sigma_{e \in \Sigma} \mathbb{P}\left(x_i, e, x_j\right) \leq 1, \forall x_i \in X \tag{1.29}$$

- x_0 représente l'état initial.

L'évolution d'un p-automate est la suivante : si le système est dans l'état x_i, la transition e vers l'état x_j est effectuée avec la probabilité $\mathbb{P}\left(x_i, e, x_j\right)$.

Une description plus détaillée de la théorie des langages probabilistes sera proposée au chapitre suivant.

1.3.3 Synthèse

Une synthèse des différentes approches présentées dans ce chapitre est proposée dans le tableau 1.1, selon deux critères : leur capacité à permettre l'identification et la quantification de séquences d'événements et leur capacité à modéliser des systèmes réparables et reconfigurables. Si l'on considère ces deux critères, quatre formalismes se dégagent : les chaînes de Markov, les automates stochastiques hybrides , les réseaux de Petri stochastiques et les langages probabilistes.

Les automates stochastiques hybrides répondent parfaitement à nos objectifs d'identification et de quantification probabiliste de séquences d'événements. Cependant, l'évaluation est réalisée, dans la plupart des cas, à l'aide de simulations de Monte-Carlo, compte tenu de l'aspect hybride qui limite, de manière importante, les capacités de calcul analytique.

par passage au complémentaire et par union dénombrable. Les *σ-algèbres* permettent de définir rigoureusement la notion d'ensemble mesurable.

TABLE 1.1 – Synthèse des approches de modélisation et d'évaluation des séquences d'événements

	Séquences		**Systèmes**	
	Identification	**Evaluation**	**Réparables**	**Reconfigurables**
AdD	-	-	-	-
AdDD	$\star^{(1)}$	$\star^{(1)}$	-	-
AdE	$\star\star$	$\star\star$	-	$\star^{(5)}$
AdED	$\star\star$	$\star\star$	$\star^{(5)}$	$\star^{(5)}$
BDMP	$\star^{(2)}$	$\star^{(3)}$	$\star\star$	$\star^{(6)}$
GBDMP	$\star^{(2)}$	-	$\star\star$	$\star\star$
AEF+Lang	$\star\star$	-	$\star\star$	$\star\star$
CdM	-	$\star^{(3)}$	$\star\star$	$\star\star$
ASH	$\star\star$	$\star^{(4)}$	$\star\star$	$\star\star$
RdPS	$\star$	$\star$	$\star\star$	$\star\star$
p-langages	$\star\star$	$\star\star$	$\star\star$	$\star\star$

$\star$: couverture partielle, $\star\star$: couverture satisfaisante
(1) éléments de séquences associés aux portes dynamiques, (2) obtention de la séquence par exploration, (3) par transformation de modèle, (4) par simulation, (5) représentation non explicite, (6) stratégies de reconfiguration limitées

Les réseaux de Petri stochastiques sont un outil très efficace pour la modélisation probabiliste de systèmes dynamiques réparables et reconfigurables. En revanche, l'extraction de séquences d'événements nécessite la construction du graphe de marquages qui reste une tâche relativement complexe en présence d'aspects temporisés et stochastiques du RdP. De plus, le travail sur le graphe de marquage impose implicitement de disposer d'une représentation non modulaire de l'ensemble du comportement du système qui rend l'identification et la quantification de séquences difficiles pour les systèmes multi-modes, multi-états ou multi-phases.

Les chaînes de Markov, malgré des capacités limitées pour l'identification et la quantification de séquences, restent néanmoins l'outil fondamental pour la modélisation et l'analyse de comportements stochastiques. En effet, la théorie classique des chaînes de Markov permet d'évaluer la distribution des probabilités d'états. Cette probabilité d'état correspond en fait, d'une part, à la probabilité d'atteindre cet état et, d'autre part, à la probabilité d'y rester. La probabilité d'atteindre un état représente la somme des probabilités des séquence d'événements conduisant à cet état. La probabilité d'un état sera donc égale à cette somme de probabilité de séquences uniquement dans le cas où celui-ci est absorbant, c'est à dire sans transitions de sortie. Il s'ensuit que l'évaluation d'une séquence d'événements à l'aide d'une chaîne de Markov nécessite la présence d'un état absorbant en fin de séquence ce qui impose une réécriture de la chaîne de Markov par exemple sous la forme d'un arbre. La prise en compte explicite des séquences apparaît dans l'équation de Chapman-Kolmogorov qui permet de calculer la probabilité de transition d'un état x vesr un état x' par un ensemble de séquences de longueur donnée k. La quantification individuelle des toutes ces séquences reste non explicite dans cette approche qui impose, par ailleurs, de fixer une longueur de séquences qui n'est pas forcément connue a priori.

Face à ces limites, les langages probabilistes, reposant sur la théorie des langages rationnels couplée aux fondements des chaînes de Markov, offrent donc de solides perspectives. Ainsi, l'identification et l'évaluation des séquences d'événements devrait bénéficier des acquis relatifs aux chaînes de Markov enrichis par les apports des expressions régulières pour la description et la manipulation des séquences. La théorie des langages probabilistes nous semble donc un outil pertinent dans le cadre de notre étude.

1.4 Conclusion

La présentation de notre contexte d'étude en début de chapitre, et notamment des propriétés des systèmes dynamiques, réparables, reconfigurables, multi-états et multi-phases a permis de mettre en évidence l'intérêt de procéder à une identification et une quantification de séquences d'événements dans les études de sûreté de fonctionnement en lieu et place des coupes qui peuvent conduire à une surévaluation des paramètres de SdF puisque seules certaines séquences conduisent à la défaillance du système. De plus, l'analyse des scénarios de défaillance doit se baser sur une description comportementale du système, de ses composants et de ses multiples modes de fonctionnement ou de défaillances.

Parmi les modèles de la littérature qui permettent l'analyse qualitative et/ou quantitative de séquences d'événements, le choix s'est porté sur la théorie de langages probabilistes dans la mesure où ils permettent une identification formelle de l'ensemble de séquences d'événements qu'un système peut suivre dans son évolution mais aussi l'évaluation de leurs probabilités d'occurrence. De plus, si cette théorie a fait l'objet d'applications dans le cadre de la synthèse de la commande [Wang et Ray, 2004], son potentiel dans le domaine de la sûreté de fonctionnement n'a pas, à notre connaissance, été exploré.

Un premier cadre formel pour l'évaluation qualitative et quantitative de séquences, basé sur la théorie de langages probabilistes, sera donc proposé au chapitre 2. Un exemple d'illustration nous permettra de montrer l'efficacité des modalités d'évaluation proposées mais en soulignera également les limites pour des systèmes complexes de grande taille. C'est pourquoi le chapitre 3 étendra ce premier cadre formel par une approche compositionnelle permettant l'évaluation de propriétés globales (au niveau du système) à partir de modèles et de calculs locaux. Une application des propositions développées aux chapitres 2 et 3 sera présentée au chapitre 4 sur un cas test industriel constitué d'un système de régulation de niveau d'eau dans un générateur de vapeur d'un réacteur à eau pressurisée.

Chapitre 2

Identification et évaluation quantitative de séquences d'événements

2.1 Introduction

Afin d'identifier et quantifier les séquences d'événements critiques, il est nécessaire de se doter d'un cadre formel de modélisation et d'évaluation permettant de représenter de manière non ambiguë les différents scénarios dysfonctionnels ainsi que le comportement fonctionnel (stratégies de commande et de reconfiguration) des systèmes étudiés. Dans le chapitre 1, nous avons montré que la théorie des langages probabilistes constituait une piste crédible en ce sens.

L'objectif principal de ce chapitre est donc de proposer un cadre formel de modélisation et d'évaluation des séquences d'événements à l'aide des langages probabilistes. Dans le contexte originel des travaux de [Garg, 1992, Kumar *et al.*, 1996, Garg *et al.*, 1999, Kumar et Garg, 2001], les données d'entrée pour l'étude d'un système sont exprimées sous la forme d'un ou plusieurs langages probabilistes. En sûreté de fonctionnement, la connaissance des séquences d'événements critiques n'est souvent pas explicite mais doit se déduire de la connaissance des comportements fonctionnels et dysfonctionnels des systèmes. En conséquence, après avoir présenté les fondements de la théorie des langages probabilistes, nous proposerons une approche basée, dans un premier temps, sur la construction d'un automate probabiliste qui sera, par la suite, exploité à l'aide de la théorie des langages pour procéder à l'évaluation des probabilités d'occurrence des séquences d'événements ou plus généralement de la criticité des séquences à l'aide d'une fonction de coût.

La deuxième partie de ce chapitre est consacrée à l'application de cette démarche sur un cas d'étude de taille et de complexité limitée mais représentatif de la classe de système considérée.

2.2 Langages probabilistes

L'objectif des travaux menés par Garg *et al.* a été d'étendre la théorie des langages rationnels, classiquement utilisés pour l'étude des SED déterministes, afin qu'elle puisse être applicable pour l'étude des SED stochastiques. Cette section propose quelques éléments de définition complémentaires à ceux donnés au chapitre précédent.

2.2.1 Relations entre p-langage et p-automate

Le comportement d'un SED stochastique est supposé être complètement décrit par son *p-langage* dont la définition a été donnée au chapitre précédent (*Définition 2*). Sous réserve que

les probabilités de terminaison correspondant à des séquences distinctes soient mutuellement exclusives, la probabilité cumulée que le système arrive dans un état terminal peut être obtenue en ajoutant les probabilités individuelles de toutes les séquences possibles :

$$\mathbb{P}\left(Syst\ e_{\Delta}\right) = \Sigma_{s\in\Sigma^*}\mathbb{P}\left(se_{\Delta}\right). \tag{2.1}$$

Exemple. Soit un processus Bernoulli où chaque expérimentation peut avoir deux résultats possibles, a et b, avec les probabilités p et $1-p$. L'ensemble des événements est $\Sigma = \{a, b\}$ et le p-langage associé est donné par :

$$\mathbb{L}\left(s\right) = p^{\#(a,s)}\left(1-p\right)^{\#(b,s)},\ \forall s \in \Sigma^*$$

où $\#\left(a,s\right)$ et $\#\left(b,s\right)$ représentent respectivement le nombre d'occurrences de a et de b dans la séquence s. Dans ce contexte, la probabilité cumulée est $\mathbb{P}\left(Bern\ e_{\Delta}\right) = 0$, ce qui signifie que le système n'atteint jamais un état final.

En considérant la définition d'un automate probabiliste (*Définition 3*), donnée au chapitre précédent, nous pouvons déduire qu'un *p-automate* A_p est appelé *déterministe* si pour chaque état $x_i \in X$ et pour chaque événement $e \in \Sigma$, il existe au maximum un état $x_j \in X$, tel que $\mathbb{P}\left(x_i, e, x_j\right) > 0$.

Chaque *p-automate* définit un *p-langage*. La fonction de probabilité de transition peut être étendue à des chemins $\pi \subset X\left(\Sigma X\right)^*$ dans le p-automate A_p (un chemin étant obtenu par la concaténation des transitions où l'état d'arrivée et l'état de départ de deux transitions consécutives coïncident). Étant donné un chemin $\pi = x_0e_{01}x_1...e_{n-1n}x_n$, on peut définir :

- sa longueur : $\mid \pi \mid = n$;
- les sous-chemins de longueur : $k \leq \mid \pi \mid \ : \pi^k = x_0e_{01}x_1...e_{k-1k}x_k$;
- la séquence associée à un chemin : $tr\left(\pi\right) = e_{01}e_{12}...e_{k-1k}$.

La probabilité des chemins est définie de manière inductive par :

$$\begin{gathered}\forall x \in X : \mathbb{P}\left(x\right) = 1 \\ \forall \pi \in X\left(\Sigma X\right)^*, e \in \Sigma \\ x_j \in X : \mathbb{P}\left(\pi e x_j\right) = \mathbb{P}\left(\pi\right)\mathbb{P}\left(x_{|\pi|}ex_j\right)\end{gathered} \tag{2.2}$$

Remarque. La relation 2.2 exprime le fait que *la probabilité d'occurrence de la séquence associée au chemin est égale au produit des probabilités des transitions individuelles constituant le chemin.*

Définition 4. *[Garg* et al.*, 1999] Étant donné un p-automate $A_p = (X, \Sigma, f, \mathbb{P}, x_0)$, le p-langage généré par A_p est défini par :*

$$\mathbb{L}_{A_p} = \sum\nolimits_{\pi:tr(\pi)=s,\pi^0=x_0} \mathbb{P}\left(\pi\right) \tag{2.3}$$

La relation 2.3 signifie que le p-langage $\mathbb{L}_{A_p}$ est donné par la somme des probabilités de tous les chemins π dont la trace est égale à la séquence s à partir de l'état initial x_0.

Inversement, *chaque p-langage peut-être représenté par un p-automate.*

Définition 5. *[Garg* et al.*, 1999] Étant donné un p-langage $\mathbb{L}$, il est généré par le p-automate $A_p = (X, \Sigma, f, \mathbb{P}, x_0)$ tel que :*

$$\forall s, t \in \Sigma^*, e \in \Sigma : \mathbb{P}\left(s, e, t\right) = \begin{cases} \frac{\mathbb{P}(\langle t \rangle)}{\mathbb{P}(\langle s \rangle)} & si\ t = se \\ 0 & autrement \end{cases} \tag{2.4}$$

Par conséquent, les p-langages sont réguliers.

2.2.2 Les fonctions de performance

Pour étudier les performances d'un SED stochastique, des fonctions pouvant exprimer ces performances sont définies [Garg *et al.*, 1999] sur l'ensemble des séquences d'événements, en associant un coût à chaque séquence du système.

Définition 6. *[Garg* et al.*, 1999] Étant donné un système avec des états terminaux, un p-langage* $\mathbb{L} \in \mathcal{L}$*,* $\sum_s \mathbb{P}(se_\Delta) = 1$ *et une fonction de performance* $F : \Sigma^* \to \mathbb{R}$*, qui représente le coût associé à chaque séquence, la valeur moyenne de* F*, notée* $E[F, \mathbb{P}(se_\Delta)] \in \mathbb{R}$ *est donnée par la relation suivante :*

$$E[F, \mathbb{P}(se_\Delta)] = \sum\nolimits_s \mathbb{P}(se_\Delta) F(s). \tag{2.5}$$

Il s'ensuit que :

$$E[F, \mathbb{P}(se_\Delta)] = \sum\nolimits_s \left[\left[\mathbb{P}(\langle s \rangle) - \sum\nolimits_{e \in \Sigma} \mathbb{P}(se) \right] F(s) \right] \tag{2.6}$$

Deux types de fonctions de performance ont été définis : la fonction de performance additive et la fonction de performance multiplicative, afin d'obtenir la valeur de ces fonctions de performances pour un système à partir des valeurs de ces fonctions correspondant aux composants du système.

2.2.2.1 Fonction de performance additive

Une fonction de performance additive $F : \Sigma^* \to \mathbb{R}$ est une fonction qui satisfait la propriété suivante :

$$F(st) = F(s) + F(t), s, t \in \Sigma^* \tag{2.7}$$

Cette fonction pourrait être utilisée par exemple pour calculer le temps moyen d'exécution d'une séquence par le système en connaissant la probabilité d'occurrence et le coût de ses sous-séquences (et par extension de chaque transition de la séquence) [Garg *et al.*, 1999].

Le lemme suivant met en avant quelques propriétés de la moyenne de la fonction de performance additive.

Lemme 1. *[Garg* et al.*, 1999] Soit une fonction de performance additive* $F : \Sigma^* \to \mathbb{R}$ *et les p-langages* $L, L_1, L_2 \in \mathcal{L}$*. Puis :*

i) $E[F, \mathbb{P}(L_1) \circ \mathbb{P}(L_2)] = E[F, \mathbb{P}(L_1)] + E[F, \mathbb{P}(L_2)]$

ii) $E[F, \mathbb{P}(L)^{(n)}] = nE[F, \mathbb{P}(L)]$

Ce lemme est utilisé dans le théorème suivant pour obtenir la performance moyenne d'un système en se basant sur la performance moyenne de ses sous-systèmes.

Théorème 1. *[Garg* et al.*, 1999] Soit une fonction de performance additive* $F : \Sigma^* \to \mathbb{R}$ *et les p-langages* $L, L_1, L_2 \in \mathcal{L}$*. Alors :*

i) $E[F, \mathbb{P}(L_1 +_p L_2)] = pE[F, \mathbb{P}(L_1)] + p'E[F, \mathbb{P}(L_2)]$

ii) $E[F, \mathbb{P}(L_1 \cdot_p L_2)] = E[F, \mathbb{P}(L_1)] + pE[F, \mathbb{P}(L_2)]$

iii) $E[F, \mathbb{P}(L^{*_p})] = \frac{1}{p'} E[F, \mathbb{P}(L)]$*, quand* $p < 1$

2.2.2.2 Fonction de performance multiplicative

Une fonction de performance multiplicative $F : \Sigma^* \to \mathbb{R}$ est une fonction qui satisfait la propriété suivante :

$$F(st) = F(s)\,F(t)\,, s, t \in \Sigma^* \tag{2.8}$$

Dans la même référence [Garg *et al.*, 1999], Garg *et al.* affirment qu'un exemple de cette fonction est la fonction exprimant la fiabilité, qui donne la probabilité que le système ne défaille pas après l'exécution d'une séquence. Effectivement, cette fonction permet de calculer la probabilité d'une séquence d'événements comme le produit des probabilités de ses sous-séquences (et par extension de chaque événement de la séquence) à condition que les sous-séquences soient indépendantes l'une de l'autre et que, quelle que soit l'une de ses sous séquences, celle-ci n'est en compétition avec aucune autre sous-séquence.

Le lemme suivant met en avant quelques propriétés de la moyenne de la fonction de performance multiplicative.

Lemme 2. *[Garg* et al.*, 1999] Soit une fonction de performance multiplicative $F : \Sigma^* \to \mathbb{R}$ et les p-langages $L, L_1, L_2 \in \mathcal{L}$. Puis :*

i) $E[F, \mathbb{P}(L_1) \circ \mathbb{P}(L_2)] = E[F, \mathbb{P}(L_1)]E[F, \mathbb{P}(L_2)]$

ii) $E[F, \mathbb{P}(L)^{(n)}] = (E[F, \mathbb{P}(L)])^n$

De la même manière que pour la fonction de performance additive, ce lemme peut être utilisé dans le théorème suivant :

Théorème 2. *[Garg* et al.*, 1999] Soit une fonction de performance multiplicative $F : \Sigma^* \to \mathbb{R}$ et les p-langages $L, L_1, L_2 \in \mathcal{L}$. Alors :*

i) $E[F, \mathbb{P}(L_1 +_p L_2)] = pE[F, \mathbb{P}(L_1)] + p'E[F, \mathbb{P}(L_2)]$

ii) $E[F, \mathbb{P}(L_1 \cdot_p L_2)] = p'E[F, \mathbb{P}(L_1)] + pE[F, \mathbb{P}(L_1)]E[F, \mathbb{P}(L_2)]$

iii) $E[F, \mathbb{P}(L^{*_p})] = (p'E[F, \mathbb{P}(L)]/1 - pE[F, \mathbb{P}(L)])\,, pE[F, \mathbb{P}(L)] < 1$

2.2.3 Application des langages probabilistes

La théorie des langages probabilistes et les fonctions de performance peuvent être utilisées pour :

- concevoir la commande des SED stochastiques par supervision ;
- décrire les mesures de performance des SED stochastiques, comme la durée moyenne d'exécution ou la fiabilité ;
- optimiser le fonctionnement des SED stochastiques.

Si pour la commande par supervision et l'optimisation, cette théorie a effectivement été appliquée [Kumar et Garg, 2001, Wang et Ray, 2004, Pantelic et Lawford, 2012], à notre connaissance aucune application de cette théorie n'a été réalisée pour l'évaluation des performances ou de la fiabilité des systèmes. Une première explication pourrait être le fait que les *p-langages* et les *p-automates* utilisent des probabilités discrètes pour décrire le fonctionnement des systèmes modélisés, alors que dans la majorité des études concernant l'évaluation des performances ou de la fiabilité, des lois de distribution des probabilités continues sont en général privilégiées. De plus, si la fonction multiplicative peut effectivement exprimer la fiabilité d'une séquence d'événements, ceci nécessite que les événements de la séquence soient indépendants dans le temps et qu'à aucun instant une de ses sous-séquences ou de ses événements ne soit en compétition avec une

autre sous-séquence ou un autre événement. Cette limitation nous semble assez restrictive, car en général dans l'étude de la fiabilité ou plus généralement de tout indicateur de sûreté de fonctionnement, les événements qui peuvent apparaître (défaillances et/ou réparations) sont souvent en compétition avec d'autres événements, qu'ils soient de même nature ou non. Un troisième point concerne l'évaluation de la fiabilité d'un système qui implique non pas seulement le calcul de la probabilité d'une séquence au bout de laquelle le système ne défaille pas, mais il sera nécessaire d'évaluer la probabilité de l'ensemble de séquences fonctionnelles pour lesquelles le système ne défaille pas. Or dans la plupart des cas des systèmes réels constitués de plusieurs composants, l'ensemble de toutes les séquences fonctionnelles est soit très grand, soit infini, ce qui rend très difficile, voire impossible l'évaluation de sa fiabilité.

Dans le cadre de la théorie de langages probabilistes, plusieurs opérateurs ont été définis sur les p-langages : le choix, la concaténation, *etc.* Des généralités sur ces opérateurs et leur application dans la modélisation de systèmes complexes seront présentées au chapitre 3.

2.3 Cadre général de la proposition

Dans le contexte des Garg *et al.* autour des langages probabilistes, le point d'entrée pour l'étude d'un système est donné sous la forme d'un ou plusieurs langages probabilistes. Ceux-ci décrivent l'ensemble des séquences admissibles caractérisant le comportement d'un système ainsi que, pour chaque séquence s, la probabilité d'occurrence $\mathbb{P}(s)$ du sous-ensemble de séquences ayant la séquence s comme préfixe $\langle s \rangle = \{st | st \in \Omega\}$. De plus, l'identification d'un état terminal et absorbant (état dangereux, par exemple) est nécessaire pour pouvoir calculer les probabilités d'atteindre cet état par une séquence donnée s, $\mathbb{P}(se_\Delta)$ (équation 1.27) ou pour toutes les séquences s qui amènent le système dans cet état, $\mathbb{P}(Syst\ e_\Delta)$ (équation 2.1). A partir de ces p-langages, il sera alors possible, selon les besoins de l'étude, de déterminer un automate probabiliste A_p qui génère le (qui correspond au) langage probabiliste donné.

Cette approche, qui est pertinente pour les études de synthèse de la commande par supervision, nous semble difficilement applicable pour des études de sûreté de fonctionnement. En effet, en sûreté de fonctionnement, le problème est plutôt inverse puisqu'il s'agit, à partir d'un modèle comportemental fonctionnel et dysfonctionnel, supposé connu, d'exhiber l'ensemble des séquences d'événements critiques et de calculer leur probabilité d'occurrence. De plus, la nécessaire présence d'états terminaux et absorbants pose quelques limites pour les analyses de fiabilité (puisque le calcul de la fiabilité du système nécessite de calculer les probabilités de toutes les séquences, , dans la majorité des cas une infinité, amenant à l'état absorbant) et pour les études de disponibilité (basée sur des modèles ne comprenant pas d'états absorbants). De plus, l'utilisation des probabilités discrètes permet uniquement le calcul de la fiabilité asymptotique pour obtenir un résultat largement connu (fiabilité du système égale à zéro $R_{Syst} = 0$).

Afin de pallier ces inconvénients, nous proposons une approche qui débute par la modélisation du comportement fonctionnel et dysfonctionnel d'un système sous la forme d'un automate fini (*étape 0*) à partir duquel il est possible d'identifier les séquences d'événements conduisant à des états d'intérêt (*étape 1*). Sur cette base, le cadre théorique des langages probabilistes est utilisé pour calculer les probabilités d'occurrence des séquences en régime asymptotique (*étape 2.1*) et en régime transitoire (*étape 2.2*). Ces trois étapes sont présentées schématiquement sur la figure (2.1) et sont décrites dans les sections suivantes.

L'intérêt de cette approche, au-delà du cadre formel de raisonnement qu'elle propose est, d'une part, de contourner l'hypothèse concernant les chaînes de Markov immergées pour lesquelles les calculs des probabilités des séquences d'événements ne sont possibles que si l'état terminal d'une

séquence est un état absorbant et, d'autre part, de prendre en compte le caractère dynamique des probabilités associés aux événements (la valeur de la probabilité d'un événement peut être différente selon l'état à partir duquel il se produit).

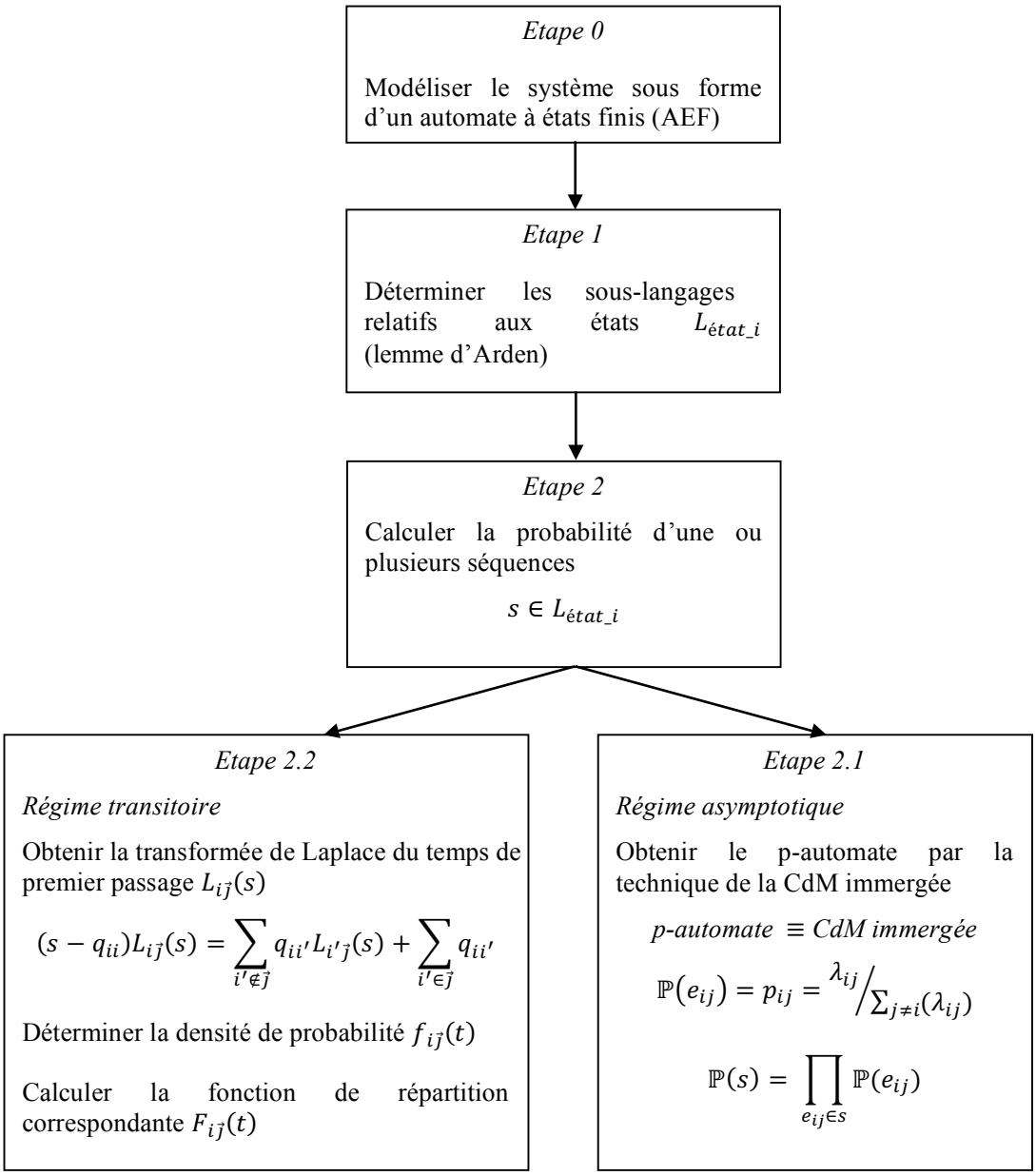


FIGURE 2.1 – Approche proposée pour l'évaluation probabiliste des séquences d'événements en régime asymptotique et transitoire

Deux remarques importantes peuvent être faites concernant cette approche qui concerne l'évaluation probabiliste des séquences.

Remarque 1. L'approche permettra d'évaluer la probabilité des séquences à l'aide d'une chaîne de Markov immergée. Ceci nécessite de considérer que le système a atteint *un régime de fonctionnement asymptotique*. En revanche, elle n'est pas restrictive à des systèmes pouvant être

modélisés par des chaînes de Markov à temps continu, c'est à dire avec une modélisation des phénomènes aléatoires limitée à des lois exponentielles. Elle peut être appliquée à tout processus stochastique (processus semi-markoviens, semi-régénératifs) pour lequel une chaîne de Markov immergée pourra être définie, autorisant ainsi l'usage d'autres types de lois de distribution.

Remarque 2. Notons que cette démarche permet de déterminer les sous-langages relatifs à un état quelconque du système et que la présence d'états absorbants dans nos modèles n'est donc plus requise. Autrement dit, le calcul des probabilités d'occurrence d'une séquence pourra être réalisé pour des séquences amenant à un état non-absorbant, ce qui ouvre la voie à des évaluations quantitatives de séquences d'événements, y compris dans des études de disponibilité.

2.4 Etape 0 : modélisation du système

Notre approche consiste d'abord à modéliser le système comme un automate à états finis et non comme un langage probabiliste donnant a priori les probabilités d'occurrence d'un ensemble de séquences. En fonction des spécificités du système à modéliser, l'automate à états finis peut correspondre à une chaîne de Markov à temps continu, mais aussi aux autres types de processus stochastiques comme les processus semi-markoviens ou semi-régénératifs.

Pour illustrer les étapes de cette approche, nous utilisons comme support un exemple de système simple constitué d'un seul composant ayant trois états : *Attente*, *Marche* et *Panne*. Ce système est modélisé sous la forme d'une chaîne de Markov à temps continu représenté sur la figure (2.2).

Initialement, le système se trouve dans l'état d'attente 1. A l'occurrence d'un taux de sollicitation α ($\alpha = 0.4$), le système arrive dans l'état 2 où son composant est en état de marche. Depuis cet état, le système peut être remis à nouveau en attente après avoir réalisé sa mission (avec un taux $\beta = 0.5$) ou, si un événement décrit par un taux de défaillance $\lambda = 4.2 \cdot 10^{-5}$ se produit, le système peut se déplacer dans l'état de panne 3. Suite à sa réparation avec un taux $\mu = 2.3 \cdot 10^{-2}$, le système revient à son état initial (état 1).

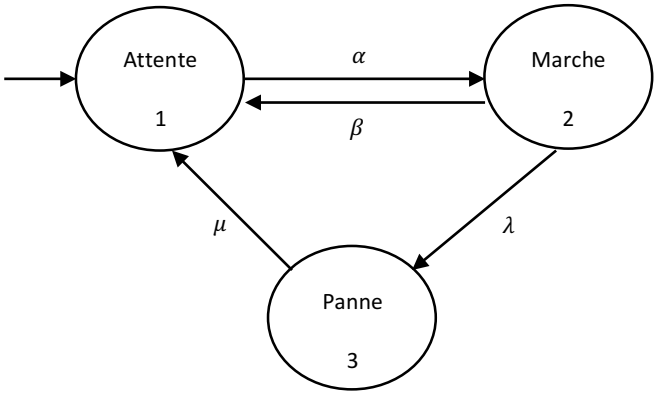


FIGURE 2.2 – Chaîne de Markov à Temps Continu de l'exemple support

2.5 Etape 1 : détermination des sous-langages

Cette étape permet de déterminer formellement l'ensemble des séquences suivies par un système. L'automate précédent est transformé en un automate non probabiliste pour lequel une fonction de transition $f(x_i, e_{ij}) = x_j$ est définie en associant un événement e_{ij} à chaque transitions entre les états x_i et x_j. L'ensemble des séquences admissibles pour le système est représenté par le langage L_{Sys} défini comme l'union des sous-langages associés à chacun des états du système. Un sous-langage L_i associé à un état x_i est défini comme l'ensemble de toutes les séquences d'événements, qui conduisent le système à partir de son état initial dans l'état considéré x_i :

$$L_{Sys} = \bigcup_{x_i} L_i \tag{2.9}$$

Pour déterminer les sous-langages L_i, nous proposons d'utiliser la théorie des langages rationnels et plus précisément **le lemme d'Arden** [Carton, 2008].

Lemme 3. *Soit deux langages A et B et soit l'équation :*

$$L_i = L_i A + B \tag{2.10}$$

où L_i représente le langage inconnu et A, B sont des ensembles des séquences d'événements, supposés connus :

1. si $\varepsilon \notin A$ (où ε est la séquence vide) la solution unique de l'équation (2.10) est $L_i = BA^*$;
2. si $\varepsilon \in A$ les solutions ont la forme $L_i = (B + C)\, A^*$ où $C \subseteq \Sigma^*$.

Ce lemme est utilisé surtout dans le cas où $\varepsilon \notin A$ et le langage $L_i = BA^*$ représente dans ce cas la solution unique de l'équation (2.10). Pour chaque état du système, x_i, l'équation (2.10) du sous-langage associé est écrite, en considérant les séquences à partir de tous les autres états $x_j \neq x_i$ et qui arrivent dans l'état x_i par une seule transition. L'ensemble de n équations (où n est le nombre d'états du système) permet d'obtenir l'expression analytique pour chaque sous-langage L_i. Ainsi, toutes les séquences d'événements qui décrivent l'évolution d'un système sont déterminées formellement sans faire appel à l'exploration du modèle (*e.g.* comme dans le cas des BDMP). En considérant chaque état de l'automate de la figure (2.2) comme étant terminal (mais non absorbant) et en appliquant l'équation (2.10) du lemme d'Arden, nous obtenons :

$$\begin{cases} L_1 = L_2 e_{21} + L_3 e_{31} \\ L_2 = L_1 e_{12} \\ L_3 = L_2 e_{23} \end{cases}$$

Ces équations sont obtenues en considérant que le sous-langage L_i associé à chaque état x_i est donné par :

$$\sum\nolimits_{x_k \in X_s, e_k \in \Sigma} L_k.e_k$$

où X_s représentent l'ensemble des états pour lesquels la fonction de transition entre x_k et x_i est définie pour au moins un événement e_k telle que $f(x_k, e_k) = x_i$ et L_k est le sous-langage associé à x_k. Les solutions de ces équations sont les expressions régulières des sous-langages associés aux états du système :

$$L_1 = (e_{12}e_{21} + e_{12}e_{23}e_{31})^*,$$
$$L_2 = L_1 e_{12} = (e_{12}e_{21} + e_{12}e_{23}e_{31})^* e_{12},$$
$$L_3 = L_1 e_{12} e_{23} = (e_{12}e_{21} + e_{12}e_{23}e_{31})^* e_{12}e_{23}.$$

A partir de ces expressions analytiques, il est possible d'extraire une ou plusieurs séquences, selon les objectifs de l'étude, et de procéder au calcul de leur probabilité d'occurrence.

2.6 Etape 2 : calcul des probabilités des séquences

A partir des séquences identifiées précédemment, le calcul des probabilités peut être effectué selon deux modes : en **régime asymptotique** où le système a atteint un comportement stationnaire et où les probabilités d'occurrence des événements ne dépend pas du temps et en **régime transitoire** où la probabilité d'occurrence de chaque événement dépend du temps. Les modalités d'obtention des probabilités d'occurrence des séquences d'événements diffèrent selon le régime dans lequel le système se trouve après exécution de la séquence ; elles sont décrites dans les deux sections suivantes.

2.6.1 Etape 2.1 : calcul en régime asymptotique

2.6.1.1 Evaluation de la probabilité d'occurrence d'une séquence d'événements

La démarche de calcul de probabilités des séquences en régime asymptotique suppose d'abord de transformer l'automate à états finis qui a été construit dans la première étape de l'approche (*étape 0*) en un p-automate. Nous allons considérer la même définition formelle d'un p-automate que celle présentée dans la section *1.3.2.5* du chapitre 1 (*définition 3*). Cette définition est suffisante et le fait d'avoir une somme des probabilités de transition à partir d'un état qui soit inférieure ou égale à 1 (équation 1.29) n'est pas restrictif, car nous pouvons toujours considérer le complément non nul de cette somme comme une probabilité de transition depuis l'état vers lui-même.

Ainsi, conformément à cette définition, l'automate à états finis représentant le système doit être accompagné des probabilités discrètes d'occurrence de chaque transition pour pouvoir le qualifier comme un p-automate. Puisque, dans les études de sûreté de fonctionnement, les lois de probabilités décrivant les phénomènes aléatoires sont plutôt des lois continues dans le temps, nous proposons d'utiliser la chaîne de Markov à temps discret immergée (CdM immergée) dans un processus stochastique continu afin de déterminer les probabilités discrètes requises par la théorie de p-automates. Cette chaîne de Markov immergée est obtenue en considérant, dans le processus stochastique continu, les instants de saut à la fin du temps de séjour dans l'état courant et elle permet au système d'atteindre la distribution stationnaire des probabilités d'état.

Cette technique (CdM immergée) n'est pas limitée aux chaînes de Markov à temps continu mais peut s'appliquer à d'autres types de processus stochastiques autorisant une résolution analytique (semi-markoviens, semi-régénératifs). Ainsi en appliquant cette technique, la probabilité d'occurrence de chaque transition sera donnée par l'équation (1.14) rappelée ci-dessous :

$$\mathbb{P}\left(e_{ij}\right) = p_{ij} = {}^{\lambda_{ij}}/_{\sum\limits_{j \neq i}\left(\lambda_{ij}\right)}$$

Remarque. Un intérêt de cette technique est que la probabilité d'occurrence d'un événement (transition) caractérisée par un taux λ_{ij} pourra être différente selon l'état de départ x_i (car elle dépendra aussi des taux d'occurrence des autres événements avec lesquels l'événement considéré est en compétition). En sûreté de fonctionnement, cet aspect est intéressant car on peut ainsi modéliser des probabilités d'occurrence différentes du même événement en fonction du contexte dans lequel l'événement se produit.

En utilisant la probabilité de chaque transition du p-automate, nous proposons d'utiliser l'équation suivante afin d'obtenir la probabilité d'occurrence en régime asymptotique pour la séquence $s = e_{12}e_{23}...e_{(n-1)n}$:

$$\mathbb{P}\left(se_{\Delta}\right) = \prod_{e_{ij} \in s} \mathbb{P}\left(e_{ij}\right), \forall s \in \Sigma^{*} \tag{2.11}$$

La signification de l'équation (2.11) est la suivante : la probabilité de la séquence est égale au produit des probabilités des tous ses événements e_{ij}.

Il est également possible de calculer symboliquement la somme des probabilités des séquences appartenant à un sous-langage, notée par $\mathbb{Q}(L_i)$:

$$\mathbb{Q}(L_i) = \sum\nolimits_{s \in L_i} \mathbb{P}(se_\Delta). \tag{2.12}$$

Il est à noter que la somme des probabilités des séquences appartenant à un sous-langage associé à un état est différente de la probabilité de l'état auquel ce sous-langage est associé. Cet indicateur est néanmoins très utile en sûreté de fonctionnement, en particulier pour borner la probabilité d'atteindre un état non désiré ou dangereux.

Le p-automate correspondant à l'automate à états finis qui représente le système à un seul composant de la figure (2.2) est donné dans la figure (2.3).

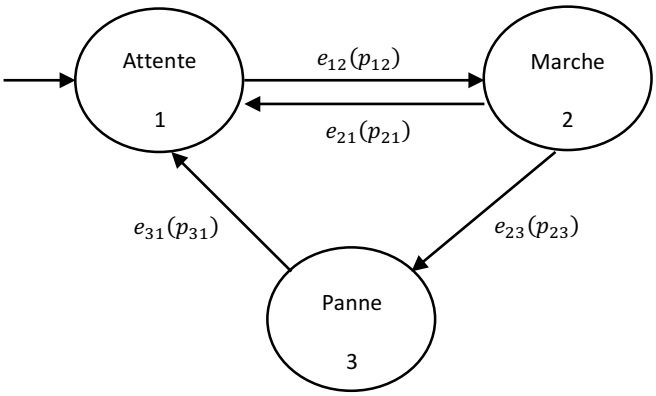


FIGURE 2.3 – Chaîne de Markov à Temps Discret représentant le p-automate de l'exemple support

En appliquant l'équation (1.14), nous avons déterminé les probabilités d'occurrence des événements e_{12}, e_{21}, e_{23} et e_{31} qui peuvent être observés dans l'automate de la figure (2.3). Les résultats obtenus sont :

$$\mathbb{P}(e_{12}) = \mathbb{P}(e_{31}) = 1,$$

$$\mathbb{P}(e_{21}) = p_{21} = \frac{\beta}{\beta + \lambda},$$

$$\mathbb{P}(e_{23}) = p_{23} = \frac{\lambda}{\beta + \lambda}.$$

Si on extrait, par exemple, deux séquences $s_1 = e_{12}e_{21}e_{12}$ et $s_2 = e_{12}e_{23}e_{31}e_{12}$ appartenant au sous-langage associé à l'état 2 (L_2) et si on applique l'équation (2.11), nous obtenons leurs probabilités d'occurrence en régime asymptotique :

$$\mathbb{P}(s_1) = p_{12}p_{21}p_{12},$$

$$\mathbb{P}(s_2) = p_{12}p_{23}p_{31}p_{12}.$$

En considérant les expressions régulières des sous-langages L_1, L_2 et L_3 obtenues à l'*Etape 1*, nous avons calculé les probabilités des différentes itérations qui composent les expressions régulières de ces sous-langages. Nous avons aussi déterminé la somme de probabilités de séquences

appartenant à chaque sous-langage en utilisant l'équation (2.12) :

$$\mathbb{Q}(L_1) = \sum_{\substack{s_i \in L_1 \\ s_i e_\Delta \to x_1}} \mathbb{P}(s_i) = 1,$$

$$\mathbb{Q}(L_2) = \sum_{\substack{s_i \in L_2 \\ s_i e_\Delta \to x_2}} \mathbb{P}(s_i) = \mathbb{Q}(L_1)\, p_{12} = 1,$$

$$\mathbb{Q}(L_3) = \sum_{\substack{s_i \in L_3 \\ s_i e_\Delta \to x_3}} \mathbb{P}(s_i) = \mathbb{Q}(L_1)\, p_{12} p_{23} = p_{23}.$$

Le résultat obtenu pour la somme des probabilités des séquences appartenant au sous-langage L_1, $\mathbb{Q}(L_1)$, est celui attendu : en effet, pour le p-automate de la figure (2.3), la somme des probabilités de toutes les séquences qui ont comme état de départ l'état 1 et comme état final l'état 1 doit être évidemment égale à 1. Quant aux résultats obtenus pour les sous-langages L_2 et L_3, ils n'ont pas d'interprétation particulière.

Les probabilités de séquences d'événements en régime asymptotique peuvent servir à l'évaluation de leur criticité selon différents critères de coût ou de longueur.

2.6.1.2 Évaluation de la criticité des séquences

Le sujet du risque joue aujourd'hui un rôle important dans la conception, le développement, l'exploitation et la gestion des composants, systèmes et structures où il existe une source potentielle de dommage, de perte ou de danger (menace) pour une cible (*e.g.* les personnes ou l'environnement) [Zio, 2013]. L'approche probabiliste de l'analyse de risque est apparue comme une méthode efficace d'analyse de la sûreté des systèmes, non limitée à la considération des scénarios de défaillance (séquences d'événements critiques) mais étendue à l'analyse de tous les scénarios possibles et leurs conséquences. L'évaluation probabiliste de ces scénarios est devenue un aspect clé qui doit être quantifié afin de gérer rationnellement et quantitativement l'incertitude [Breeding *et al.*, 1992, Aven, 2003, Bedford et Cooke, 2001, Kaplan et Garrick, 1981]. Il est en effet important de déterminer quelles sont les séquences les plus critiques qui posent des problèmes du point de vue de la sûreté du système ou qui affectent ses performances afin de pouvoir rechercher des moyens (stratégies de commande, système de sécurité, architecture du système, *etc*) afin d'éviter l'occurrence de la séquence (ou des séquences).

En analyse des risques, la criticité d'un événement (ou d'une séquence d'événements) est souvent définie comme le produit entre la probabilité d'occurrence de l'événement (ou de la séquence) et de la gravité des effets induits par l'occurrence de cet événement (ou de la séquence). Cette évaluation de la criticité peut alors être assimilée à un indicateur quantitatif que l'on associera à chaque événement ou séquence d'événements.

A partir des résultats obtenus sur l'évaluation des probabilités d'occurrence des séquences d'événements en régime asymptotique, nous proposons de déterminer la criticité des séquences d'événements, soit sur la base d'un coût (représentant la mobilisation de ressources financières, des effets sur l'environnement humain ou matériel, *etc.*), soit sur la base de la longueur des séquences.

Criticité déterminée par le coût global d'une séquence d'événements

Dans un premier temps, le coût global d'une séquence d'événements est déterminé en fonction du coût de séjour dans les états x_i avant l'apparition des événements e_{ij}, de la probabilité d'occurrence des événements e_{ij} (p_{ij}) et du coût des événements (transitions) (c_{ij}). L'expression suivante donne le coût global pour une séquence s_i :

$$C_g\left(s_i\right) = \sum\nolimits_{e_{ij} \in s_i} p_{ij}\left(\eta_{ij}\gamma_i + c_{ij}\right) \tag{2.13}$$

où η_{ij}, donné par l'équation (1.22), représente le temps moyen de séjour dans l'état x_i avant l'occurrence de l'événement e_{ij}.

Le premier terme de l'équation (2.13) représente le coût associé au séjour dans l'état x_i avant l'occurrence de l'événement e_{ij}. Il est calculé comme le produit entre le temps moyen de séjour η_{ij} et un coût par unité de temps associé à l'état x_i, γ_i. Le deuxième terme, c_{ij}, est le coût associé à la transition de l'état x_i vers l'état x_j. Le coût unitaire associé aux états (γ_i) et le coût associé aux transitions (c_{ij}) peuvent avoir des valeurs positives ou négatives selon la nature des états/événements : positive si le système se retrouve dans un état où il accomplit ses fonctions et négative si des ressources sont dépensées pour des réparations (dans des états défaillants). Par conséquent, le coût de la séquence s_i est la somme des coûts de tous les événements e_{ij} qui composent la séquence.

Dans un second temps, l'équation (2.13) est appliquée pour calculer la criticité des séquences identifiées dans la section 2.6.1.1. Il faut noter que le classement des séquences peut être différent selon que l'on considère uniquement leur probabilité d'occurrence ou bien leur criticité.

Criticité déterminée par la longueur d'une séquence d'événements

Une autre manière d'analyser la criticité d'une séquence d'événements est de considérer sa longueur. En effet, le coût probabiliste d'une séquence va dépendre du nombre d'événements qui la composent. La criticité de la séquence (s_i) est alors donnée par le produit de la probabilité d'occurrence de la séquence $\mathbb{P}\left(s_i\right)$ en régime asymptotique donnée par l'équation (2.11) et de sa longueur $l\left(s_i\right)$:

$$C_l\left(s_i\right) = \mathbb{P}\left(s_i\right) l\left(s_i\right) \tag{2.14}$$

Il faut souligner que la criticité d'une séquence basée sur ce calcul ne dépend pas uniquement de la longueur de la séquence mais également de sa probabilité d'occurrence. Ainsi, une séquence avec un petit nombre d'événements mais une grande probabilité d'occurrence peut avoir une criticité plus importante qu'une séquence avec un grand nombre d'événements mais une faible probabilité d'occurrence. Autrement dit, une séquence contenant un petit nombre d'événements peut être plus critique qu'une séquence en comprenant un grand nombre en fonction des probabilités des événements qui composent ces séquences.

Dans la première partie de la section 2.6, nous avons établi les modalités de calcul des probabilités d'occurrence des séquences d'événements en régime asymptotique puis de leur criticité. La section suivante aborde le calcul des probabilités d'occurrence de séquences d'événements en régime transitoire.

2.6.2 Etape 2.2 : calcul en régime transitoire

Un système évoluant en régime transitoire avant d'atteindre son régime asymptotique, notre objectif est de proposer une méthode de calcul des probabilités des séquences en tenant compte

des instants auxquels les événements de la séquence se produisent. En particulier, l'ordre d'occurrence des événements dans une séquence a une signification temporelle directe, puisque si on considère une séquence contenant deux événements $s_i = e_{mn}e_{no}$, on peut facilement en déduire que le deuxième événement e_{no} se produira sur un intervalle de temps consécutif à l'intervalle de temps où s'est produit le premier événement e_{mn}. Cette relation directe entre occurrence des événements et écoulement du temps, que le calcul des probabilités en régime transitoire prend en compte, ajoute un argument supplémentaire en faveur de la quantification des séquences d'événements en lieu et place des coupes.

Dans la bibliographie, différentes approches et outils permettent l'évaluation quantitative des séquences en régime transitoire. Dans le cas de systèmes pouvant être représentés comme des processus markoviens ou semi-markoviens, la détermination des probabilités des séquences en régime transitoire reste accessible du point de vue analytique mais dans les cas plus généraux, la simulation de Monte Carlo ou des techniques basées sur l'exploration de modèles restent souvent les seules solutions [Aubry et Brînzei, 2015]. En particulier, l'outil appelé FigSeq, développé par EDF, permet le calcul de probabilités de séquences (en régime transitoire) qui conduisent le système vers des états définis comme "cible" par l'utilisateur. Il repose sur une description du système sous la forme d'un BDMP ou d'un automate à états finis et applique différents algorithmes permettant l'identification de séquences par exploration du modèle et leur quantification. Même si les approches de la littérature s'avèrent être efficaces dans certains cas, elles n'offrent pas de cadre formel pour la détermination exhaustive et la quantification des séquences en régime transitoire.

L'approche que nous proposons pour calculer les probabilités d'occurrence des séquences en régime transitoire est basée sur l'algorithme de Harisson. Dans l'article publié en 2002, P.G. Harrison et W.J. Knottenbelt [Harrison et Knottenbelt, 2002] développent une technique pour déterminer la distribution de probabilité du temps de premier passage pour une chaîne de Markov à temps continu. Cette technique est basée sur le calcul de la transformée de Laplace de la densité de probabilité du temps de premier passage d'un état initial vers un ensemble d'états cibles.

2.6.2.1 Temps du premier passage dans les chaînes de Markov à temps continu

Soit une chaîne de Markov à temps continu (CMTC) finie et irréductible avec n états $1, 2, ..., n$ et la matrice M son générateur. Si $X(t)$ dénote l'état de la CMTC au moment t ($t \geq 0$), alors le temps de premier passage d'un état source i vers un ensemble non-vide des états cibles $\overrightarrow{j}$ est :

$$T_{i\overrightarrow{j}}(t) = inf\{u > 0 : X(t+u) \in \overrightarrow{j} \mid X(t) = i\} \ \ (\forall t \geq 0)$$

La fonction inf (infimum) représente la borne inférieure (le plus grand des minorants) de l'ensemble des temps de premier passage.

Pour une CMTC homogène du point de vue du temps, $T_{i\overrightarrow{j}}(t)$ est indépendant de t, ce qui signifie que :

$$T_{i\overrightarrow{j}} = inf\{u > 0 : X(u) \in \overrightarrow{j} \mid X(0) = i\}$$

$T_{i\overrightarrow{j}}$ est une variable aléatoire avec une fonction de densité de probabilité associée $f_{i\overrightarrow{j}}(t)$ telle que :

$$Pr\left(a < T_{i\overrightarrow{j}} < b\right) = \int_a^b f_{i\overrightarrow{j}}(t)\,dt \ \ (0 \leq a < b)$$

L'objectif est de déterminer $f_{i\overrightarrow{j}}(t)$. En fait, cela implique la convolution des temps de séjour dans les états sur tous les chemins/séquences possibles (y compris les boucles) depuis l'état

i vers l'un des états de l'ensemble $\overrightarrow{j}$. Si on transfère ce problème dans le domaine de Laplace nous pouvons exploiter la propriété de la transformée de base qui stipule que la transformée de la convolution de deux fonctions est le produit entre les transformées de ces deux fonctions [J.Abate *et al.*, 2000]. A partir de la transformée de Laplace, on peut déterminer la densité de probabilité, $f_{i\overrightarrow{j}}(t)$, pour n'importe quel instant t en utilisant un des algorithmes pour l'inversion de la transformée numérique. On obtient alors $f_{i\overrightarrow{j}}(t)$ à un instant t à partir de $L_{i\overrightarrow{j}}(s)$ calculée pour plusieurs valeurs de s.

En général, la valeur de $L_{i\overrightarrow{j}}(s)$ peut être calculée en résolvant un ensemble d'équations linéaires obtenues de la manière suivante :

$$\begin{aligned} L_{i\overrightarrow{j}}(s) &= \int_0^\infty e^{-st} f_{i\overrightarrow{j}}(t)\,dt \\ &= E[e^{-sT_{i\overrightarrow{j}}}] \\ &= \sum_{i'\notin\overrightarrow{j}} -\frac{\lambda_{ii'}}{\lambda_{ii}} E[e^{-s\left(S_{ii'}+T_{i'\overrightarrow{j}}\right)}] + \sum_{i'\in\overrightarrow{j}} -\frac{\lambda_{ii'}}{\lambda_{ii}} E[e^{-sS_{ii'}}] \\ &= \sum_{i'\notin\overrightarrow{j}} \frac{\lambda_{ii'}}{(s-\lambda_{ii})} L_{i'\overrightarrow{j}}(s) + \sum_{i'\in\overrightarrow{j}} \frac{\lambda_{ii'}}{(s-\lambda_{ii})} \end{aligned}$$

ce qui donne :

$$(s-\lambda_{ii})\,L_{i\overrightarrow{j}}(s) = \sum_{i'\notin\overrightarrow{j}} \lambda_{ii'} L_{i'\overrightarrow{j}}(s) + \sum_{i'\in\overrightarrow{j}} \lambda_{ii'} \tag{2.15}$$

où $S_{ii'} \sim Exp(-\lambda_{ii})$ est le temps de séjour dans l'état i $(1 \le i \le n)$ avant l'occurrence de l'événement $e_{ii'}$. Le premier passage de l'état i à l'ensemble des états cibles $\overrightarrow{j}$ peut se faire via des états intermédiaires i' qui appartiennent ou bien qui n'appartiennent pas à $\overrightarrow{j}$. Ces deux cas correspondent respectivement au deuxième et au premier terme de l'équation (2.15).

Si on exprime ce système de n équations linéaires sous la forme d'une équation matricielle, il devient dans le cas où le seul état cible est l'état 1 ($\overrightarrow{j} = \{1\}$) :

$$\begin{pmatrix} s-\lambda_{11} & -\lambda_{12} & \cdots & -\lambda_{1n} \\ 0 & s-\lambda_{22} & \cdots & -\lambda_{2n} \\ 0 & -\lambda_{32} & \cdots & -\lambda_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & -\lambda_{n2} & \cdots & s-\lambda_{nn} \end{pmatrix} \begin{pmatrix} L_{1\overrightarrow{j}}(s) \\ L_{2\overrightarrow{j}}(s) \\ L_{3\overrightarrow{j}}(s) \\ \vdots \\ L_{n\overrightarrow{j}}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_{21} \\ \lambda_{31} \\ \vdots \\ \lambda_{n1} \end{pmatrix} \tag{2.16}$$

Dans ce cas, le but est de déterminer la densité de probabilité du premier passage d'un certain état i du système vers l'état 1. La démarche consiste à résoudre l'équation (2.16) pour déterminer $L_{i1}(s)$ et ensuite, à l'aide d'un algorithme d'inversion numérique, obtenir $f_{i1}(t)$. Ayant $f_{i1}(t)$, la fonction de répartition $F_{i1}(t)$ peut être calculée, sa valeur correspondant à la probabilité d'occurrence des séquences qui se produisent pendant le temps de premier passage depuis l'état i vers l'état 1.

Dans les autres cas, l'équation (2.16) se réécrit en fonction du choix des états cibles qui appartiennent à l'ensemble $\overrightarrow{j}$, le deuxième membre de l'égalité étant obtenu à l'aide de l'équation (2.15). Une démarche similaire de résolution peut alors être appliquée.

De manière générale, la valeur de la fonction de répartition du temps de premier passage $F_{i\overrightarrow{j}}(t)$ dépend de l'intervalle de temps $[0,t]$ sur lequel sont effectués les calculs. Dans le cas

particulier où les états source et cible sont identiques $\overrightarrow{j} = \{i\}$, pour un processus stochastique ergodique, la valeur de $F_{i\overrightarrow{j}}(t)$ converge vers 1 si l'intervalle de temps est suffisamment grand ($t \to \infty$). En effet, le calcul de $F_{i\overrightarrow{j}}(t)$ prend en considération tous les chemins de l'état i vers l'état i dans le graphe fortement connexe associé au processus ergodique. Dans le cas où l'ensemble d'états cible $\overrightarrow{j}$ est différent de l'état source i, $F_{i\overrightarrow{j}}(t)$ ne converge pas vers 1 puisqu'il existe des états k avec $k \notin \overrightarrow{j}$ atteignables depuis l'état i, sauf si $\overrightarrow{j}$ est un état absorbant ou un groupe d'états absorbants (le graphe n'est plus fortement connexe dans ce cas).

Afin d'illustrer l'application de la technique de Harrison [Harrison et Knottenbelt, 2002], nous reprenons l'exemple support du système à un seul composant représenté par l'automate probabiliste de la figure (2.3). Nous déterminons tout d'abord la matrice M associée à la chaîne de Markov à temps continu :

$$M = \begin{pmatrix} -\lambda_{12} & \lambda_{12} & 0 \\ \lambda_{21} & -(\lambda_{21}+\lambda_{23}) & \lambda_{23} \\ \lambda_{31} & 0 & -\lambda_{31} \end{pmatrix}$$

où λ_{ij} correspond au taux de transition entre les états i et j. Par exemple λ_{12} correspond au taux de transition entre les états 1 et 2 avec une valeur numérique égale à $\lambda_{12} = \alpha = 0.4$.

1er cas : état cible $\overrightarrow{j} = \{1\}$

On part de l'hypothèse qu'à l'instant initial le système se trouve dans l'état 1 où son unique composant est en attente et on va considérer que l'état 1 est le seul état cible. L'objectif est alors de déterminer la fonction de répartition du temps de premier passage depuis l'état 1 vers lui même. En considérant la méthode de Harrison, on réécrit l'équation (2.16) pour notre système :

$$\begin{pmatrix} s+\lambda_{12} & -\lambda_{12} & 0 \\ 0 & s+\lambda_{21}+\lambda_{23} & -\lambda_{23} \\ 0 & 0 & s+\lambda_{31} \end{pmatrix} \begin{pmatrix} L_{11}(s) \\ L_{21}(s) \\ L_{31}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_{21} \\ \lambda_{31} \end{pmatrix}$$

Les solutions de ce système à 3 inconnues sont les transformées de Laplace du temps de premier passage depuis chacun des 3 états vers l'état 1 :

$$L_{11}(s) = \frac{1}{s+\lambda_{12}} \cdot \lambda_{12} \cdot \frac{\lambda_{21} + \frac{\lambda_{23}\lambda_{31}}{s+\lambda_{31}}}{s+\lambda_{21}+\lambda_{23}}$$

$$L_{21}(s) = \frac{\lambda_{21} + \frac{\lambda_{23}\lambda_{31}}{s+\lambda_{31}}}{s+\lambda_{21}+\lambda_{23}}$$

$$L_{31}(s) = \frac{\lambda_{31}}{s+\lambda_{31}}$$

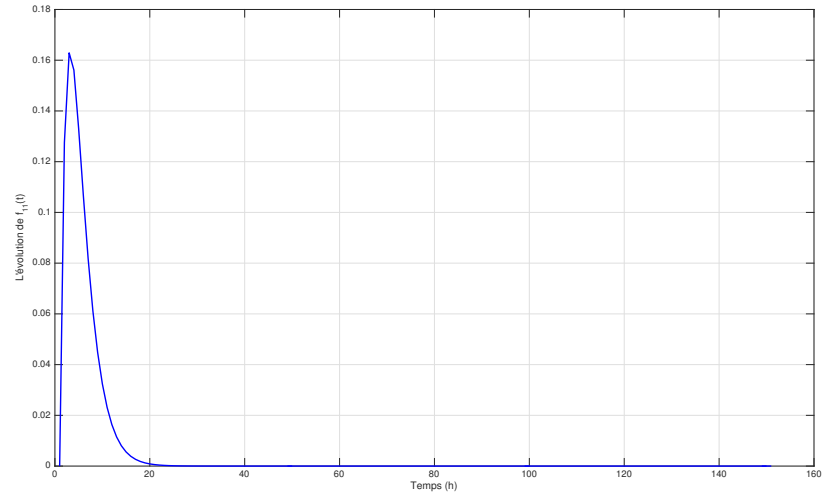
En prenant la transformée de Laplace $L_{11}(s)$ et en l'inversant numériquement (par exemple à l'aide de la fonction *ilaplace* fournie par Matlab), nous obtenons la densité de probabilité $f_{11}(t)$. Après avoir déterminé cette densité de probabilité, la fonction de répartition correspondante $F_{11}(t)$ peut être déterminée (par exemple à l'aide d'une autre fonction Matlab nommée *trapz*). Cette fonction de répartition représente la probabilité que la séquence d'événements se produise avant la date t du premier passage de l'état 1 à lui même. Pour gérer tous les calculs, un script Matlab a été développé puis lancé sur de multiples intervalles de temps différents. Un premier

calcul fournit l'évolution de $f_{11}(t)$ donnée dans la figure (2.4a). La surface en-dessous de la courbe $f_{11}(t)$ fournit la valeur numérique de la fonction de répartition correspondante $F_{11}(t)$:

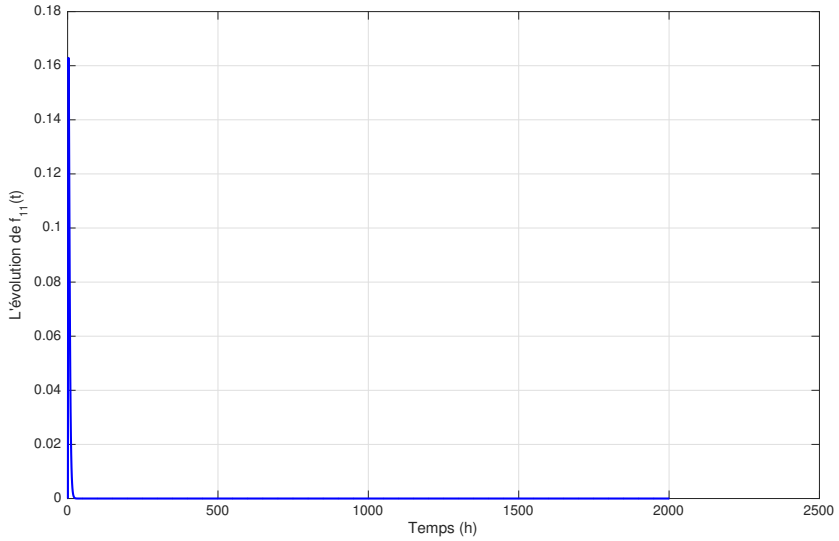
$$F_{11}(t) = 0.9325, \ \textit{pour } t = 150 \textit{ heures}$$

Au fur et à mesure que l'on augmente l'intervalle de temps sur lequel le calcul numérique est effectué, la valeur de $F_{11}(t)$ s'accroît jusqu'à ce qu'elle se stabilise à 1. La figure (2.4b) présente l'évolution de $f_{11}(t)$, ce qui donne pour $F_{11}(t)$:

$$F_{11}(t) = 0.9935, \ \textit{pour } t = 2000 \textit{ heures}$$



(a) Calcul effectué sur 150 heures



(b) Calcul effectué sur 2000 heures

FIGURE 2.4 – Densité de probabilité du temps de premier passage de l'état 1 vers lui même

Ces résultats contribuent à valider l'approche proposée puisque lorsque les intervalles de temps augmentent, le système finit par atteindre un régime asymptotique où la somme des probabilités de séquences appartenant au sous-langage L_1 (séquences de l'état initial (1) vers lui même), est $\mathbb{Q}(L_1) = 1$. En particulier, le résultat obtenu pour le calcul sur 2000 heures signifie que le système a atteint son régime asymptotique.

2ème cas : état cible $\overrightarrow{j} = \{3\}$

Considérons maintenant un deuxième état cible. Il s'agit de l'état 3 où le système est en panne à cause de l'occurrence de l'événement de défaillance caractérisé par le taux λ. Avec $\overrightarrow{j} = \{3\}$, l'équation (2.16) devient :

$$\begin{pmatrix} s+\lambda_{12} & -\lambda_{12} & 0 \\ -\lambda_{21} & s+\lambda_{21}+\lambda_{23} & 0 \\ -\lambda_{31} & 0 & s+\lambda_{31} \end{pmatrix} \begin{pmatrix} L_{13}(s) \\ L_{23}(s) \\ L_{33}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_{23} \\ 0 \end{pmatrix}$$

De la même façon que pour l'état cible précédent, le système de 3 équations est résolu avec pour solutions les transformées de Laplace du temps de premier passage depuis chacun des trois états du système vers l'état 3. Les résultats obtenus sont les suivants :

$$L_{13}(s) = \frac{\lambda_{12}}{(s+\lambda_{12})} \cdot \frac{\lambda_{23}}{s+\lambda_{21}+\lambda_{23}-\frac{\lambda_{21}\lambda_{12}}{s+\lambda_{12}}}$$

$$L_{23}(s) = \frac{\lambda_{23}}{s+\lambda_{21}+\lambda_{23}-\frac{\lambda_{21}\lambda_{12}}{s+\lambda_{12}}}$$

$$L_{33}(s) = \frac{\lambda_{31}}{(s+\lambda_{31})} \cdot \frac{\lambda_{12}}{(s+\lambda_{12})} \cdot \frac{\lambda_{23}}{s+\lambda_{21}+\lambda_{23}-\frac{\lambda_{21}\lambda_{12}}{s+\lambda_{12}}}$$

Dans la mesure où l'état initial lors de la mise en marche du système est l'état 1, nous allons chercher maintenant à déterminer la fonction de répartition du temps de premier passage depuis l'état initial vers l'état cible 3 (panne). Ainsi, à partir de la solution obtenue pour $L_{13}(s)$ et en utilisant la fonction Matlab *ilaplace*, la densité de probabilité $f_{13}(t)$ est déterminée. Ensuite, à l'aide de la fonction Matlab *trapz*, la fonction de répartition $F_{13}(t)$ est calculée. Enfin, un script Matlab est utilisé pour effectuer les calculs sur différents intervalles de temps. L'évolution de $f_{13}(t)$ est présentée sur la figure (2.5). Pour la fonction de répartition $F_{13}(t)$ la valeur obtenue est :

$$F_{13}(t) = 8.2133 \cdot 10^{-5},\ \textit{pour } t = 2000\ \textit{heures}$$

Comme nous l'avions signalé pour les calculs relatifs à l'état cible 1, un intervalle de temps de 2000 heures correspond à un système ayant atteint son régime asymptotique. Le fait que le résultat obtenu, dans ce cas, en régime transitoire converge vers celui obtenu en régime asymptotique pour la somme de probabilités de séquences, qui amènent le système de son état initial vers l'état de panne 3, contribue à valider la technique proposée pour le calcul de probabilités de séquences en régime transitoire.

2.6.2.2 Temps du premier passage dans les processus semi-markoviens

La technique de Harrison *et al.*, concernant l'évaluation des séquences d'événements en régime transitoire, n'est pas limitée aux chaînes de Markov à temps continu, elle a été étendue aux

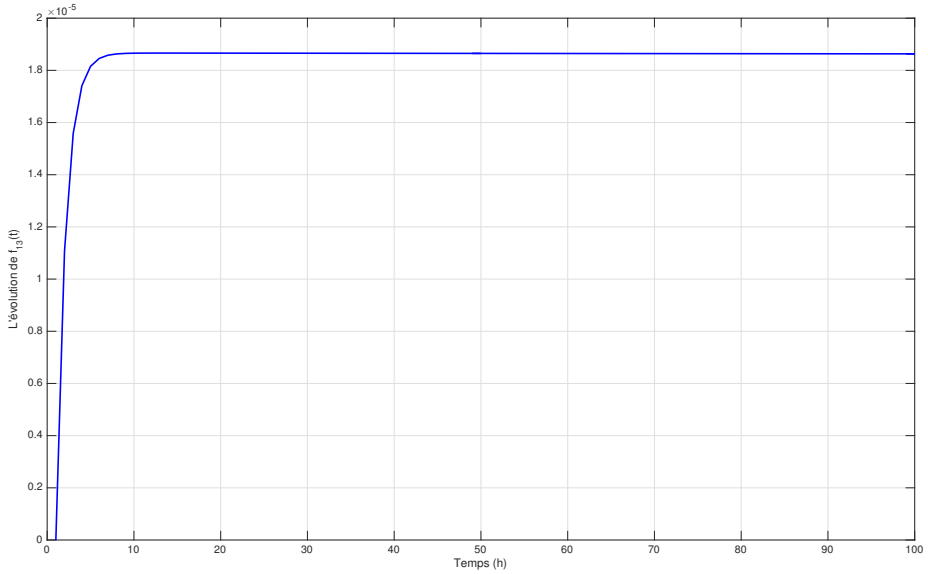


FIGURE 2.5 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 3 (sur 2000 heures)

processus semi-markoviens [Harrison et Knottenbelt, 2002]. Ces modèles contiennent des transitions caractérisées par des lois exponentielles mais également des transitions déterministes. De ce fait, l'évaluation quantitative des séquences va se réaliser différemment des chaînes de Markov à temps continu où toutes les transitions sont exponentielles.

Considérons un processus de renouvellement $\{(X_n, T_n) \mid n \geq 0\}$ où T_n est le temps de la $n^{\text{ème}}$ transition ($T_0 = 0$) et $X_n \in S$ est l'état du système au moment T_n^+. Soit $Q_{ij}(n,t)$, le noyau de ce processus :

$$Q_{ij}(n,t) = P(X_{n+1} = j, T_{n+1} - T_n \leq t \mid X_n = i)$$

avec $i, j \in S$. Le processus semi-markovien défini par le noyau Q est alors $Y(t) = X_n$ où n est l'entier positif pour lequel $t \in [T_n, T_{n+1})$. On considère le processus semi-markovien homogène du point de vue du temps où $Q_{ij}(n,t)$ ne dépend pas de n ; on le note $Q_{ij}(t)$. Le temps de premier passage depuis un état source i et un ensemble d'états cible $\overrightarrow{j}$ a alors une densité de probabilité qui est la transformée de Laplace $L_{i\overrightarrow{j}}(s)$ donnée par l'équation suivante, analogue à l'équation (2.15) [Harrison et Knottenbelt, 2002] :

$$\begin{aligned} L_{i\overrightarrow{j}}(s) &= \int_0^\infty e^{-st} f_{i\overrightarrow{j}}(t)\, dt \\ &= E[e^{-sT_{i\overrightarrow{j}}}] \\ &= \sum_{i' \notin \overrightarrow{j}} p_{ii'} E[e^{-s\left(S_{ii'} + T_{i'\overrightarrow{j}}\right)}] + \sum_{i' \in \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}] \end{aligned}$$

$$= \sum_{i' \notin \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}] E[e^{-sT_{i'\overrightarrow{j}}}] + \sum_{i' \in \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}]$$
$$= \sum_{i' \notin \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}] L_{i'\overrightarrow{j}} + \sum_{i' \in \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}]$$

ce qui donne :

$$L_{i\overrightarrow{j}}(s) - \sum_{i' \notin \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}] L_{i'\overrightarrow{j}} = \sum_{i' \in \overrightarrow{j}} p_{ii'} E[e^{-sS_{ii'}}] \tag{2.17}$$

où, comme pour les chaînes de Markov à temps continu, $S_{ii'}$ est le temps de séjour dans l'état i avant occurrence de l'événement $e_{ii'}$ et $p_{ii'}$ est la probabilité d'occurrence de l'événement $e_{ii'}$.

En fonction de la nature des transitions, exponentielles ou déterministes, le temps de séjour dans les états du système est déterminé différemment. On rappelle que, pour un processus semi-markovien, la probabilité des transitions, exponentielles ou déterministes, est donnée par :

$$p_{ii'} = Q_{ii'}(\infty) = \lim_{x \to +\infty} Q_{ii'}(t)$$

Dans le cas des transitions exponentielles, le temps de séjour dans un état i avant l'occurrence de l'événement $e_{ii'}$ est donné par :

$$S_{ii'} \sim F_{ii'}(s_i) = \frac{Q_{ii'}(s_i)}{p_{ii'}} \tag{2.18}$$

où $F_{ii'}(s_i)$ est la fonction de répartition du temps de séjour dans l'état i avant l'occurrence de l'événement $e_{ii'}$ et $Q_{ii'}$ est l'élément du noyau Q qui correspond à cet événement. Si on remplace le temps de séjour $S_{ii'}$ dans l'équation (2.17), qui nous donne la transformée de Laplace du temps de premier passage de l'état i vers l'ensemble d'états cible $\overrightarrow{j}$, celle-ci devient :

$$L_{i\overrightarrow{j}}(s) = \sum_{i' \notin \overrightarrow{j}} p_{ii'} \int_0^\infty e^{-ss_i} \frac{1}{p_{ii'}} Q'_{ii'}(s_i)\, \mathrm{d}s_i \; L_{i'\overrightarrow{j}} + \sum_{i' \in \overrightarrow{j}} p_{ii'} \int_0^\infty e^{-ss_i} \frac{1}{p_{ii'}} Q'_{ii'}(s_i)\, \mathrm{d}s_i$$
$$= \sum_{i' \notin \overrightarrow{j}} \int_0^\infty e^{-ss_i} dQ_{ii'}(s_i) \; L_{i'\overrightarrow{j}} + \sum_{i' \in \overrightarrow{j}} \int_0^\infty e^{-ss_i} dQ_{ii'}(s_i)$$
$$= \sum_{i' \notin \overrightarrow{j}} r^*_{ii'}(s)\, L_{i'\overrightarrow{j}} + \sum_{i' \in \overrightarrow{j}} r^*_{ii'}(s)$$

qui peut se réécrire :

$$L_{i\overrightarrow{j}}(s) - \sum_{i' \notin \overrightarrow{j}} r^*_{ii'}(s)\, L_{i'\overrightarrow{j}} = \sum_{i' \in \overrightarrow{j}} r^*_{ii'}(s) \tag{2.19}$$

Autrement dit, $r^*_{ii'}(s)$ représente la transformée Laplace-Stieltjes de $Q_{ii'}(t)$ donnée par :

$$r^*_{ii'}(s) = \int_0^\infty e^{-st} dQ_{ii'}(t) \tag{2.20}$$

Dans le cas des transitions déterministes, le temps de séjour dans un état i avant l'occurrence de l'événement $e_{ii'}$ est donné par :

$$S_{ii'} = \begin{cases} 0, & t < d_{e_{ii'}} \\ d_{e_{ii'}}, & t \geq d_{e_{ii'}} \end{cases}$$

où $d_{e_{ii'}}$ est la durée nécessaire à la détection de $e_{ii'}$ (*e.g.* la détection d'une défaillance).

Ainsi, pour déterminer la transformée de Laplace du temps de premier passage d'un état i vers un ensemble d'états $\overrightarrow{j}$, il suffit de remplacer l'expression correspondant au temps de séjour $S_{ii'}$ dans l'équation (2.17).

2.6.3 Synthèse

Dans la section 2.6, nous avons proposé un cadre formel pour l'évaluation des probabilités d'occurrence de séquences d'événements en régime asymptotique, à l'aide d'une chaîne de Markov immergée, puis en régime transitoire pour les processus markoviens puis semi-markoviens. La section suivante présente une application de cette démarche sur un cas d'étude.

2.7 Cas d'étude

Le cas d'étude retenu est un système de contrôle de la température d'un four représenté dans la figure (2.6). Le système de contrôle contient deux boucles de régulation.

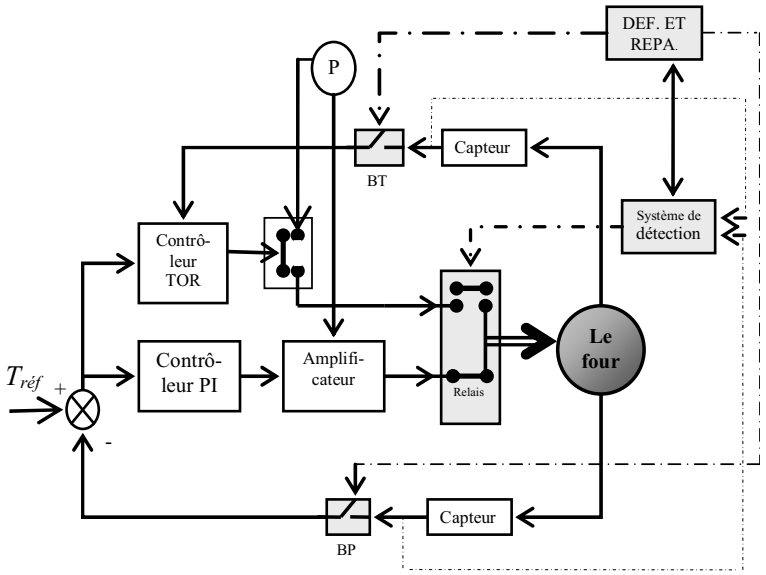


FIGURE 2.6 – Diagramme structurel du système de contrôle de la température d'un four

La première inclut un contrôleur proportionnel et intégral (PI) dont le rôle est de contrôler la température du four en fonction de la température de référence en fournissant à chaque instant la puissance juste nécessaire. Elle correspond au fonctionnement normal du système et est active au démarrage du système. En cas de défaillance du régulateur PI, une boucle de secours de type tout ou rien (TOR) permet de maintenir la température du four aux alentours de la température de référence $+/-\Delta t$ en chauffant à pleine puissance ou en ne chauffant pas.

Les deux boucles ne doivent évidemment pas fonctionner en même temps. Pour cela, un relais bascule ses deux contacts permettant ainsi d'activer soit le régulateur PI soit le régulateur TOR. L'ordre de basculement est donné par un système de détection dont le rôle est d'identifier les défaillances et les réparations et de réagir en commutant d'un régulateur à l'autre. Lorsque le système de détection repère que la température du four est hors contrôle (franchissement d'un seuil correspondant à une température du four atteignant une valeur dangereuse ($t \geq t_{max}$, $t_{max} = 240$°C), il donne l'ordre au relais de basculer sur la boucle TOR (boucle du contrôleur PI ouverte et boucle du contrôleur TOR fermée) ce qui a pour effet de confier au régulateur TOR le contrôle de la température ($t_{infTOR} \leq t \leq t_{supTOR}$). Une fois que le contrôleur PI est réparé, le système de détection bascule le relais sur la boucle de celui-ci et ouvre la boucle du TOR

ce qui a pour effet de rendre au régulateur PI le contrôle de la température. Dans un premier temps, l'organe de détection (diagnostic) est supposé parfait mais il est possible de lui affecter une probabilité de non-détection par exemple.

On suppose que les contrôleurs peuvent subir des défaillances en fonctionnement (après un temps aléatoire de taux λ_{PI} ou λ_{TOR}) ainsi qu'un refus au démarrage avec une probabilité discrète (p_{refPI} ou p_{refTOR}).

Enfin, dès qu'un défaillance des contrôleurs PI ou TOR est détectée, un processus de réparation du contrôleur défaillant est enclenché (on considère une réparation de durée aléatoire de taux μ_{PI} ou μ_{TOR}).

On note que t_{ref} est la température de référence (t_{ref} = 190℃), P est l'alimentation de puissance, BP est l'ouverture de la boucle du contrôleur PI et BT celle de la boucle du contrôleur TOR et $DEF.\ ET\ REPA.$ représente les défaillances et les réparations.

Ce fonctionnement représente la version simplifiée du cas d'étude considéré dans [Perez-Castaneda, 2009] et défini afin d'illustrer les problèmes posés en fiabilité dynamique. Le modèle présenté ici a été obtenu en éliminant la partie continue du système décrivant l'évolution de la température (décrite par des équations différentielles) dans les différents états du système.

Pour identifier et évaluer quantitativement les séquences d'événements caractérisant le comportement du système de contrôle de la température du four, le cadre formel proposé précédemment est appliqué. La présentation des modèles et des résultats reprend le plan en trois étapes suivi en section 2.6 et schématisé sur la figure (2.1).

2.7.1 Modélisation du système

La première étape de l'approche consiste à modéliser le système sous la forme d'un automate à états finis. Nous allons considérer trois types d'événements : les événements caractérisés par des des lois de probabilité continues (défaillances et réparations des contrôleurs de taux λ_{PI}, λ_{TOR}, μ_{PI}, μ_{TOR}), les événements caractérisés par des durées déterministes (détection des pannes après une période d_{smax} et d_{smin}, commutations arrêt/démarrage du four par le contrôleur TOR avec une durée d_{infTOR}, d_{supTOR}) et les événements caractérisés par des probabilités discrètes (refus au démarrage de probabilités p_{refPI} et p_{refTOR}).

La présence de ces trois types de transitions (probabilistes, déterministes et discrètes) justifie l'utilisation de processus semi-markoviens (stochastiques à temps continu) pour la modélisation du système. La figure (2.7) présente l'automate à états finis qui correspond au modèle semi-markovien du cas d'étude.

A l'instant initial, le système se trouve dans l'état 1 de l'automate où la température du four est contrôlée par le PI alors que le contrôleur TOR est en attente. La défaillance du contrôleur PI amène le système dans l'état 2 où le contrôleur PI est défaillant mais encore actif et où le contrôleur TOR n'a pas encore été sollicitée. Après détection d'un franchissement de seuil de la température maximum (de durée d_{smax}, temps nécessaire pour que le système de détection identifie la défaillance du contrôleur PI), le système se retrouve dans l'état 3 où le contrôleur TOR est actif et le contrôleur PI en réparation. Dans les états 3 et 4, la température du four est régulée par le contrôleur TOR (arrêt du four dans l'état 3 jusqu'à ce que la température atteigne le seuil bas t_{min} = 150℃ après une durée d_{infTOR}, démarrage du four dans l'état 4 jusqu'a atteindre le seuil de température haut après une durée d_{supTOR}). Pendant que le système se trouve dans ces états 3 ou 4, le contrôleur PI peut être réparé et le système peut revenir à l'état initial 1 où la température est à nouveau régulée par le contrôleur PI. Si à la suite de sa réparation, le contrôleur PI refuse de démarrer, avec une probabilité discrète p_{refPI}, le système est maintenu dans les états 3 ou 4 (boucles sur les états 3 et 4). Le fonctionnement est similaire

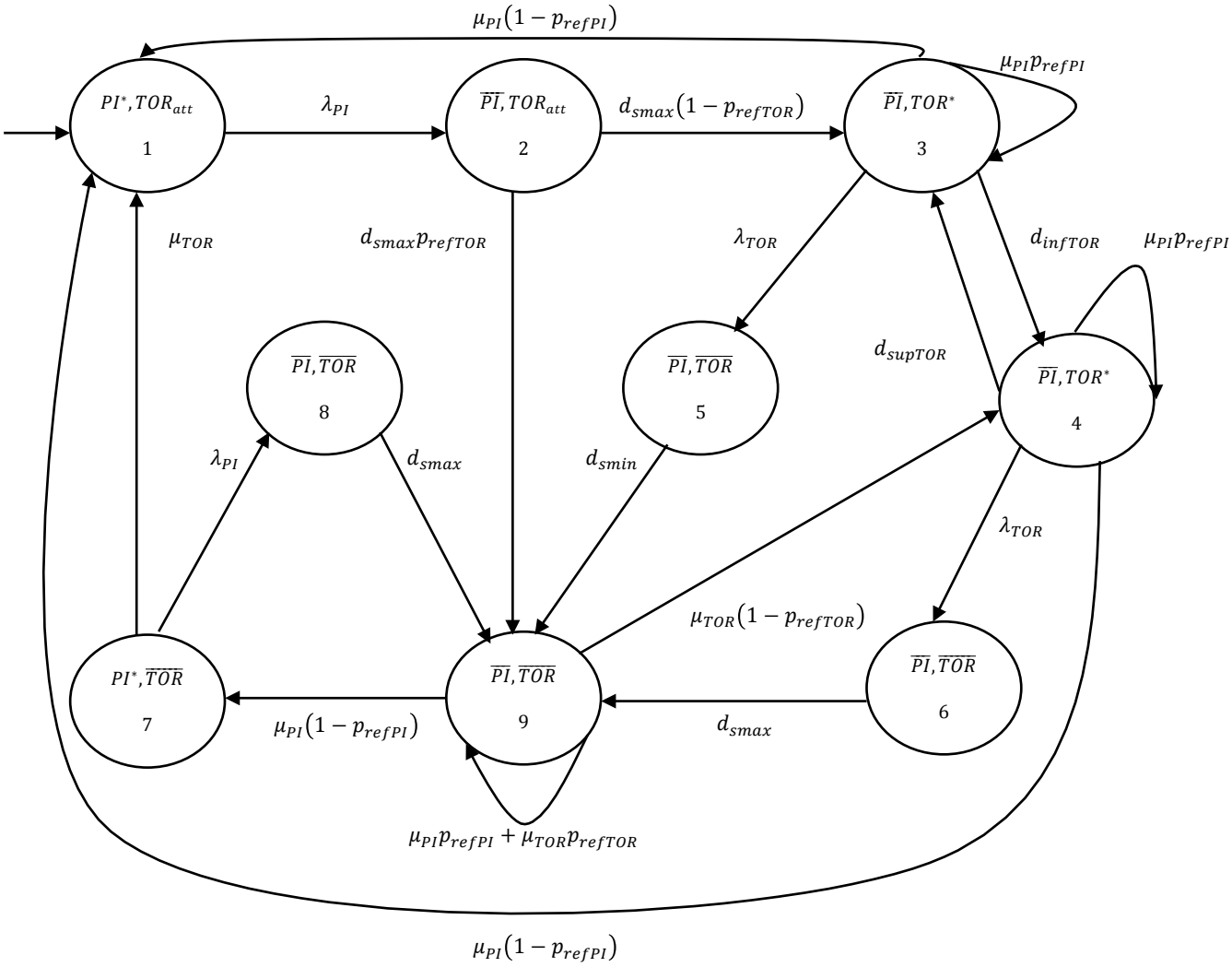


FIGURE 2.7 – Le modèle semi-markovien du cas d'étude

lorsque le contrôleur TOR subit une défaillance en fonctionnement (états 5 et 6 puis état 9 après détection). L'état 7 est un état où le régulateur PI est en fonctionnement mais où le régulateur TOR est en panne, ce qui aura pour effet, en cas de défaillance de PI, d'interdire le recours au contrôleur TOR (état 8). L'état 9 représente l'état dangereux où la température du four n'est plus contrôlée car les deux contrôleurs sont défaillants.

Les valeurs qui caractérisent les transitions du système sont constantes (en exceptant λ_{PI}) : $\lambda_{TOR} = 2{\cdot}10^{-5}h^{-1}$, $\mu_{PI} = 8{\cdot}10^{-2}h^{-1}$, $\mu_{TOR} = 10^{-1}h^{-1}$, $p_{refPI} = 0.03$, $p_{refTOR} = 0.05$. Puisque nous prenons en considération le vieillissement du contrôleur PI, son taux de défaillance λ_{PI} est caractérisé par une loi de distribution de *Weibull*. La valeur du paramètre de forme considéré est $\beta = 7$ (ceci indique que le taux de défaillance augmente avec le temps) et la valeur du paramètre d'échelle de la distribution est $\eta = 5 \cdot 10^4$. En tenant compte de ces valeurs pour les paramètres de la distribution de Weibull, la valeur de $MTTF$ du contrôleur PI est $MTTF_{PI} = 28570\ h$.

2.7.2 Détermination des sous-langages

La deuxième étape de notre approche concerne la détermination des sous-langages associés aux états du système. Le sous-langage L_i relatif à l'état x_i représente l'ensemble des séquences amenant le système depuis l'état initial à l'état x_i. En considérant chaque état de l'automate à états finis associé au système (figure 2.7) comme état terminal (non absorbant), on peut écrire les équations suivantes relatives à chacun des neuf états :

$$L_1 = L_3 e_{31} + L_4 e_{41} + L_7 e_{71} \tag{2.21}$$

$$L_2 = L_1 e_{12} \tag{2.22}$$

$$L_3 = L_2 e_{23} + L_3 e_{33} + L_4 e_{43} \tag{2.23}$$

$$L_4 = L_3 e_{34} + L_4 e_{44} + L_9 e_{94} \tag{2.24}$$

$$L_5 = L_3 e_{35} \tag{2.25}$$

$$L_6 = L_4 e_{46} \tag{2.26}$$

$$L_7 = L_9 e_{97} \tag{2.27}$$

$$L_8 = L_7 e_{78} \tag{2.28}$$

$$L_9 = L_2 e_{29} + L_5 e_{59} + L_6 e_{69} + L_8 e_{89} + L_9 e_{99} \tag{2.29}$$

A titre d'exemple, l'équation (2.21) est obtenue en considérant les trois transitions aboutissant à l'état 1 : depuis l'état 3 (L_3) sur occurrence de l'événement e_{31} représentant la réparation du régulateur PI et son non-refus au démarrage, depuis l'état 4 (L_4) sur occurrence de l'événement e_{41} ayant la même signification que l'événement e_{31} et depuis l'état 7 (L_7) sur occurrence de l'événement e_{71} représentant la réparation du régulateur TOR. De la même manière, l'équation (2.29) est obtenue en considérant l'ensemble des transitions aboutissant à l'état 9 (état dangereux où la température n'est plus contrôlée) : depuis l'état 2 (L_2) sur occurrence de l'événement e_{29} représentant la détection de la défaillance du régulateur PI et le refus au démarrage du contrôleur TOR, depuis l'état 5 (L_5) sur occurrence de l'événement e_{59} qui représente la détection de la défaillance du contrôleur TOR, depuis l'état 6 (L_6) sur occurrence de l'événement e_{69} ayant la même signification que l'événement e_{59}, depuis l'état 8 (L_8) sur occurrence de l'événement e_{89} représentant la détection de la défaillance du régulateur PI et enfin depuis l'état 9 (L_9) sur occurrence de l'événement e_{99}. Un raisonnement analogue est appliqué pour obtenir les équations (2.22 à 2.28).

En appliquant le lemme d'Arden (2.10) [Carton, 2008], les solutions à cet ensemble d'équations (2.21 à 2.29) sont données par les expressions régulières suivantes :

- pour l'état 1 :

$$\begin{aligned} L_1 = \{&[e_{12}e_{23}e_{33}^*e_{31} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{31} + e_{41}\right)] + \\ &[e_{12}e_{29} + e_{12}e_{23}e_{33}^*e_{35}e_{59} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)] \\ &[e_{97}e_{78}e_{89} + e_{99} + e_{94}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)]^* \\ &[e_{94}\left(e_{44} + e_{43}e_{33}^*e_{69}e_{34}\right)^*\left(e_{43}e_{33}^*e_{31} + e_{41}\right) + e_{97}e_{71}]\}^* \end{aligned} \tag{2.30}$$

- pour l'état 2 :

$$\begin{aligned} L_2 = \{&[e_{12}e_{23}e_{33}^*e_{31} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{31} + e_{41}\right)] + \\ &[e_{12}e_{29} + e_{12}e_{23}e_{33}^*e_{35}e_{59} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)] \\ &[e_{97}e_{78}e_{89} + e_{99} + e_{94}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)]^* \\ &[e_{94}\left(e_{44} + e_{43}e_{33}^*e_{69}e_{34}\right)^*\left(e_{43}e_{33}^*e_{31} + e_{41}\right) + e_{97}e_{71}]\}^*e_{12} \end{aligned} \tag{2.31}$$

- pour l'état 3 :

$$L_3 = \left(L_1e_{12}e_{23} + L_4e_{43}\right)e_{33}^* \tag{2.32}$$

- pour l'état 4 :

$$\begin{aligned} L_4 = L_1\{&e_{12}e_{23}e_{33}^*e_{34} + \\ &[e_{12}e_{29} + e_{12}e_{23}e_{33}^*e_{35}e_{59} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)] \\ &[e_{97}e_{78}e_{89} + e_{99} + e_{94}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)]^*e_{94}\} \\ &\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^* \end{aligned} \tag{2.33}$$

- pour l'état 5 :

$$L_5 = \left(L_1e_{12}e_{23} + L_4e_{43}\right)e_{33}^*e_{35} \tag{2.34}$$

- pour l'état 6 :

$$L_6 = L_4e_{46} \tag{2.35}$$

- pour l'état 7 :

$$\begin{aligned} L_7 = L_1&[e_{12}e_{29} + e_{12}e_{23}e_{33}^*e_{35}e_{59} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)] \\ &[e_{97}e_{78}e_{89} + e_{99} + e_{94}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)]^*e_{97} \end{aligned} \tag{2.36}$$

- pour l'état 8 :

$$\begin{aligned} L_8 = L_1&[e_{12}e_{29} + e_{12}e_{23}e_{33}^*e_{35}e_{59} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)] \\ &[e_{97}e_{78}e_{89} + e_{99} + e_{94}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)]^*e_{97}e_{78} \end{aligned} \tag{2.37}$$

- pour l'état 9 :

$$\begin{aligned} L_9 = L_1&[e_{12}e_{29} + e_{12}e_{23}e_{33}^*e_{35}e_{59} + e_{12}e_{23}e_{33}^*e_{34}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)] \\ &[e_{97}e_{78}e_{89} + e_{99} + e_{94}\left(e_{44} + e_{43}e_{33}^*e_{34}\right)^*\left(e_{43}e_{33}^*e_{35}e_{59} + e_{46}e_{69}\right)]^* \end{aligned} \tag{2.38}$$

Nous pouvons observer que les expressions régulières obtenues pour les sous-langages sont assez complexes par rapport au nombre d'états (9) du système considéré qui reste pourtant de taille réduite. Si leur identification demeure néanmoins tout à fait possible pour ce cas d'étude, le passage à l'échelle sur des systèmes de taille industrielle laisse présager un problème d'explosion combinatoire qui rendra difficile, voire impossible, l'identification et l'exploitation de ces sous-langages. Ce constat justifie, en partie, la proposition d'une approche compositionnelle et modulaire qui fera l'objet du chapitre suivant.

2.7.3 Calcul des probabilités des séquences

2.7.3.1 Calcul en régime asymptotique

A partir des expressions des sous-langages obtenues à l'étape précédente, les étapes suivantes consistent à :

- déterminer les probabilités d'occurrence de tous les événements en utilisant la technique de chaîne de Markov immergée ;
- extraire des séquences ayant un intérêt particulier pour l'analyse de la sûreté de fonctionnement du système étudié et déterminer leur probabilité en régime asymptotique ;
- calculer la somme des probabilités des séquences appartenant à un sous-langage, c'est-à-dire la probabilité d'atteindre un des états depuis l'état initial ;
- valider de manière analytique les résultats obtenues ;
- évaluer la criticité des séquences en fonction de leur coût global et de leur longueur.

Détermination des probabilités d'événements

Le système étant représenté sous la forme d'un automate à états finis, l'objectif est de le transformer en un automate probabiliste. Pour cela, les probabilités d'occurrence de toutes les transitions sont calculées en utilisant la technique de chaîne de Markov immergée ; la probabilité de chaque événement e_{ij} est déterminé en utilisant l'équation (1.14).

Les résultats numériques obtenus sont donnés dans le tableau 2.1. On remarque que la somme des probabilités de tous les événements ayant un état x_i comme état de départ est égale à 1, ce qui vérifie l'équation (1.17) ; pour l'état 4, par exemple, $\mathbb{P}(e_{41}) + \mathbb{P}(e_{43}) + \mathbb{P}(e_{44}) + \mathbb{P}(e_{46}) = 1$.

TABLE 2.1 – Probabilités d'occurrence des événements

Evénement e_{ij}	**Probabilité d'occurrence** $\mathbb{P}(e_{ij})$	**Evénement** e_{ij}	**Probabilité d'occurrence** $\mathbb{P}(e_{ij})$
e_{12}	1	e_{46}	$2.4994 \cdot 10^{-4}$
e_{23}	0.9500	e_{59}	1
e_{29}	0.0500	e_{69}	1
e_{31}	0.9698	e_{71}	0.9997
e_{33}	0.0300	e_{78}	$3.4988 \cdot 10^{-4}$
e_{34}	$1.7530 \cdot 10^{-7}$	e_{89}	1
e_{35}	$2.4994 \cdot 10^{-4}$	e_{94}	0.5278
e_{41}	0.9698	e_{97}	0.4311
e_{43}	$3.2162 \cdot 10^{-11}$	e_{99}	0.0411
e_{44}	0.0300		

En associant une probabilité d'occurrence à chaque transition (événement) de l'automate à états finis, construit dans la première étape de l'approche, on obtient l'automate probabiliste présenté dans la figure (2.8).

Détermination des probabilités d'occurrence de quelques séquences

Les probabilités d'occurrence d'événements $\mathbb{P}(e_{ij})$ calculés précédemment sont utilisées pour obtenir les probabilités d'occurrence de quelques séquences d'intérêt. Nous allons illustrer ces calculs pour deux sous-langages : L_1 et L_9. Le sous-langage L_1 représente l'ensemble des séquences

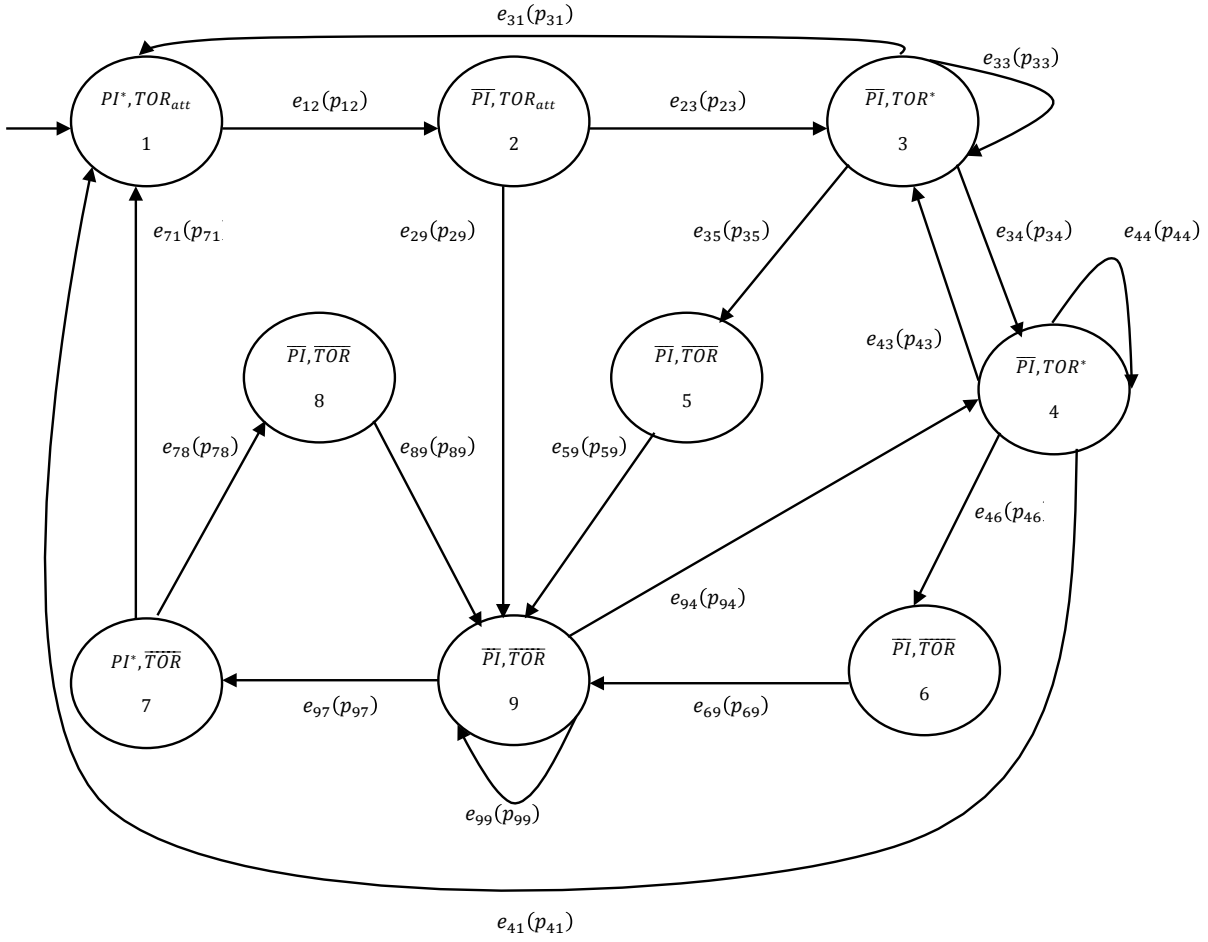


FIGURE 2.8 – Chaîne de Markov en Temps Discret représentant le p-automate du cas d'étude

qui amènent le système dans son état de bon fonctionnement à partir de lui même. Le sous-langage L_9 représente l'ensemble des séquences qui amènent le système depuis son état initial vers son état dangereux.

En fonction des objectifs de l'étude, il est possible d'extraire quelques séquences d'intérêt appartenant au sous-langage L_1 dont l'expression régulière est donnée par l'équation (2.30). La probabilité des séquences sélectionnées est obtenue en utilisant l'équation (2.11). Le tableau 2.2 présente les probabilités pour quelques séquences d'événements. Les résultats obtenus permettent de propose une première classification des séquences basée sur leur probabilité d'occurrence.

La même démarche peut être mise en œuvre pour le sous-langage L_9 dont l'expression régulière est donnée par l'équation (2.38). Le tableau 2.3 présente les probabilités d'occurrence de quelques séquences appartenant à L_9, c'est à dire conduisant à un état de panne et permet d'établir une première classification des séquences d'événements les plus critiques (au sens de leur probabilité d'occurrence).

Enfin, de manière identique, cette démarche peut être appliquée pour les séquences appartenant aux sous-langages L_2 à L_8.

TABLE 2.2 – Probabilités d'occurrence de quelques séquences conduisant à l'état 1

Séquence (s_i)	**Probabilité de la séquence** $(\mathbb{P}(s_i))$
$s_1 = e_{12}e_{23}e_{31}$	0.9213
$s_2 = e_{12}e_{23}e_{34}e_{41}$	$1.6150 \cdot 10^{-7}$
$s_3 = e_{12}e_{29}e_{97}e_{71}$	0.0215
$s_4 = e_{12}e_{29}e_{94}e_{41}$	0.0256
$s_5 = e_{12}e_{23}e_{34}e_{43}e_{31}$	$5.1943 \cdot 10^{-18}$
$s_6 = e_{12}e_{29}e_{99}e_{94}e_{43}e_{31}$	$3.3837 \cdot 10^{-14}$
$s_7 = e_{12}e_{23}e_{33}e_{35}e_{59}e_{97}e_{71}$	$3.0691 \cdot 10^{-6}$
$s_8 = e_{12}e_{29}e_{97}e_{78}e_{89}e_{97}e_{71}$	$3.2502 \cdot 10^{-6}$
$s_9 = e_{12}e_{23}e_{33}e_{35}e_{59}e_{94}e_{41}$	$3.6449 \cdot 10^{-6}$
$s_{10} = e_{12}e_{23}e_{34}e_{44}e_{46}e_{69}e_{97}e_{71}$	$5.3801 \cdot 10^{-13}$
$s_{11} = e_{12}e_{29}e_{94}e_{44}e_{43}e_{35}e_{59}e_{97}e_{71}$	$2.7419 \cdot 10^{-18}$
$s_{12} = e_{12}e_{23}e_{34}e_{46}e_{69}e_{97}e_{78}e_{89}e_{97}e_{71}$	$2.7057 \cdot 10^{-15}$

TABLE 2.3 – Probabilités d'occurrence de quelques séquences conduisant à l'état 9

Séquence (s_i)	**Probabilité de la séquence** $(\mathbb{P}(s_i))$
$s_1 = e_{12}e_{29}$	0.0500
$s_2 = e_{12}e_{29}e_{99}$	0.0021
$s_3 = e_{12}e_{23}e_{35}e_{59}$	$2.3744 \cdot 10^{-4}$
$s_4 = e_{12}e_{23}e_{34}e_{46}e_{69}$	$4.1624 \cdot 10^{-11}$
$s_5 = e_{12}e_{29}e_{97}e_{78}e_{89}$	$7.5418 \cdot 10^{-6}$
$s_6 = e_{12}e_{23}e_{33}e_{35}e_{59}$	$7.1214 \cdot 10^{-6}$
$s_7 = e_{12}e_{23}e_{33}e_{34}e_{46}e_{69}$	$1.2484 \cdot 10^{-12}$
$s_8 = e_{12}e_{23}e_{33}e_{31}e_{12}e_{29}$	0.0014
$s_9 = e_{12}e_{29}e_{99}e_{97}e_{78}e_{89}e_{99}$	$1.2747 \cdot 10^{-8}$
$s_{10} = e_{12}e_{23}e_{35}e_{59}e_{94}e_{46}e_{69}$	$3.1321 \cdot 10^{-8}$

Calcul de la somme des probabilités des séquences appartenant à un sous-langage

Les expressions régulières des sous-langages L_1-L_9, données par les équations (2.30 - 2.38), présentent au moins un opérateur d'itérations (fermeture de Kleene notée *) qui, appliqué à un événement e_{ii}^* (ou à un sous-langage L_i), est défini symboliquement par :

$$e_{ii}^* = \epsilon + e_{ii} + e_{ii}e_{ii} + ... + \underbrace{e_{ii}..e_{ii}}_{\text{n fois}} \tag{2.39}$$

Connaissant la probabilité d'occurrence de l'événement e_{ii}, la probabilité d'une itération est donnée par :

$$\mathbb{P}(e_{ii}^*) = \mathbb{P}(\epsilon) + \mathbb{P}(e_{ii}) + \mathbb{P}(e_{ii})^2 + ... + \mathbb{P}(e_{ii})^n = \frac{1}{1 - \mathbb{P}(e_{ii})} \tag{2.40}$$

A titre d'exemple, l'itération $(e_{44} + e_{43}e_{33}^*e_{34})^*$, présente dans l'expression de chaque sous-langage, a pour développement :

$$(e_{44} + e_{43}e_{33}^*e_{34})^* = \epsilon + (e_{44} + e_{43}e_{33}^*e_{34}) + (e_{44} + e_{43}e_{33}^*e_{34})(e_{44} + e_{43}e_{33}^*e_{34}) + ... + \underbrace{(e_{44} + e_{43}e_{33}^*e_{34})..(e_{44} + e_{43}e_{33}^*e_{34})}_{\text{n fois}}. \tag{2.41}$$

Cette itération contient à son tour l'itération e_{33}^* qui peut se développer en utilisant la relation 2.39. La probabilité de l'itération $(e_{44} + e_{43}e_{33}^*e_{34})^*$ peut alors être obtenue en appliquant la relation 2.40 :

$$\mathbb{P}\left((e_{44} + e_{43}e_{33}^*e_{34})^*\right) = \frac{1}{1 - [1 - (\mathbb{P}(e_{46}) + \mathbb{P}(e_{41}) + \mathbb{P}(e_{43})) + \frac{\mathbb{P}(e_{34})\mathbb{P}(e_{43})}{\mathbb{P}(e_{31})+\mathbb{P}(e_{35})+\mathbb{P}(e_{34})}]} \tag{2.42}$$

A partir des calculs de probabilités pour les itérations et des probabilités d'occurrence d'événements $\mathbb{P}(e_{ij})$ (données dans le tableau 2.1), la somme des probabilités des séquences appartenant à chaque sous-langage L_1 à L_9 peut être obtenue en appliquant l'équation (2.12). Les résultats numériques sont présentés dans la tableau 2.4.

TABLE 2.4 – Somme des probabilités des séquences appartenant aux sous-langages L_1-L_9

Sous-langage (L_i)	**Somme des probabilités des séquences appartenant ($\mathbb{Q}(L_i)$)**
L_1	1
L_2	1
L_3	0.9794
L_4	$6.2897 \cdot 10^{-4}$
L_5	$2.4479 \cdot 10^{-4}$
L_6	$1.5720 \cdot 10^{-7}$
L_7	0.0226
L_8	$7.9038 \cdot 10^{-6}$
L_9	0.0524

Le résultat obtenu pour la somme des probabilités des séquences appartenant au sous-langage L_1 est $\mathbb{Q}(L_1) = 1$. Ce résultat était attendu puisque la probabilité de toutes les séquences qui reviennent à l'état 1 du p-automate de la figure (2.8) en partant de lui-même doit être égale à 1 lorsque le système ne peut être bloqué dans un autre état intermédiaire (2, 3, ..., 9) et qu'il n'y a pas de groupe d'états atteignable depuis l'état 1 et à partir duquel le système ne peut plus atteindre l'état 1. Dit autrement, le système finira toujours par revenir dans un état de bon fonctionnement. Concernant le sous-langage L_2, la somme des probabilités des séquences lui appartenant peut s'expliquer en tenant compte de son expression régulière $L_2 = L_1e_{12}$, et du fait que la somme des probabilités des séquences appartenant au sous-langage L_1 est égale à 1 et que $\mathbb{P}(e_{12}) = 1$. Par ailleurs, les résultats obtenus pour les autres sous-langages n'ont pas de signification particulière.

Validation des résultats

Afin de vérifier la validité des probabilités des séquences en régime asymptotique, obtenues en appliquant notre approche sur le cas d'étude, nous proposons l'approche suivante :

- déterminer la distribution stationnaire des probabilités d'états pour la chaîne de Markov immergée : en utilisant les équations (1.18) et (1.19).
- prouver que les probabilités des séquences d'événements appartenant aux sous-langages associés aux états du système vérifient la relation suivante :

$$\pi(x_1) \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}j}} \mathbb{P}(se_\Delta) = \pi(x_1) \cdot \mathbb{Q}(L_j) = \pi(x_j) \tag{2.43}$$

La relation (2.43) donne la probabilité de l'état atteint x_j, après l'occurrence de toutes les séquences appartenant à un sous-langage du p-automate, sous-langage qui amène le système

depuis l'état initial x_1 dans l'état final x_j. Nous pouvons exprimer cette probabilité par le produit entre la probabilité de l'état initial x_1 et la somme des probabilités d'occurrence de tous les chemins (toutes les séquences) qui amènent le système dans l'état final considéré, x_j.

Nous disposons de toutes les données nécessaires pour vérifier numériquement ces résultats pour notre cas d'étude.

La première étape de l'approche de validation débute par la détermination de la matrice des probabilités de transition de la chaîne de Markov immergée du cas d'étude (M_{im}) de la figure (2.8) :

$$M_{im} = \begin{bmatrix} 0 & p_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{23} & 0 & 0 & 0 & 0 & 0 & p_{29} \\ p_{31} & 0 & p_{33} & p_{34} & p_{35} & 0 & 0 & 0 & 0 \\ p_{41} & 0 & p_{43} & p_{44} & 0 & p_{46} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{59} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{69} \\ p_{71} & 0 & 0 & 0 & 0 & 0 & 0 & p_{78} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_{89} \\ 0 & 0 & 0 & p_{94} & 0 & 0 & p_{97} & 0 & p_{99} \end{bmatrix} \tag{2.44}$$

où les probabilités des événements p_{ij} sont donnés dans le tableau 2.1.

A partir de cette matrice et en appliquant les relations 1.18 et 1.19, nous obtenons :

$$[\pi_{im1}\ \pi_{im2}\ \pi_{im3}\ ...\ \pi_{im9}\ 1] = [\pi_{im1}\ \pi_{im2}\ \pi_{im3}\ ...\ \pi_{im9}] \cdot M_{im} \tag{2.45}$$

où $[\pi_{im1}\ \pi_{im2}\ \pi_{im3}\ ...\ \pi_{im9}]$ représente la distribution stationnaire des probabilités d'états en régime asymptotique du cas d'étude considéré.

La résolution de l'équation ci-dessus fournit les valeurs suivantes pour les probabilités d'états :

$$[\pi_{im1}\ \pi_{im2}\ \pi_{im3}\ \pi_{im4}\ \pi_{im5}\ \pi_{im6}\ \pi_{im7}\ \pi_{im8}\ \pi_{im9}] =$$
$$[0.3243\ 0.3243\ 0.3177\ 0.0092\ 0.0001\ 0.0000\ 0.0073\ 0.0000\ 0.0170] \tag{2.46}$$

A partir de ces éléments, nous pouvons alors vérifier que les probabilités des sous-langages respectent la relation (2.43) qui s'établit comme suit pour chacun des sous-langages L_i, $i = 1..9$ (décrits par les expressions régulières données dans les équations 2.30 - 2.38) :

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}1}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_1) = \pi_{im1} \tag{2.47}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}2}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_2) = \pi_{im2} \tag{2.48}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}3}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_3) = \pi_{im3} \tag{2.49}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}4}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_4) = \pi_{im4} \tag{2.50}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}5}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_5) = \pi_{im5} \tag{2.51}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}6}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_6) = \pi_{im6} \tag{2.52}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}7}} \mathbb{P}(se_\Delta) = \pi_{im1} \cdot \mathbb{Q}(L_7) = \pi_{im7} \tag{2.53}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}8}} \mathbb{P}\left(se_\Delta\right) = \pi_{im1} \cdot \mathbb{Q}\left(L_8\right) = \pi_{im8} \tag{2.54}$$

$$\pi_{im1} \cdot \Sigma_{\substack{s\in\Sigma^* \\ 1\xrightarrow{s}9}} \mathbb{P}\left(se_\Delta\right) = \pi_{im1} \cdot \mathbb{Q}\left(L_9\right) = \pi_{im9} \tag{2.55}$$

Remarque. Les neuf relations ci-dessus permettent de valider l'approche proposée pour l'évaluation probabiliste des séquences d'événements en régime asymptotique. Les calculs numériques ont été réalisés sous *Matlab* qui utilise une représentation de type « simple précision » pour les valeurs réelles.

Evaluation de la criticité des séquences

Suite à l'application de l'approche que nous avons proposée sur le cas d'étude, nous avons déterminé de manière formelle l'ensemble des séquences d'événements existantes et nous avons calculé leur probabilités d'occurrence en régime asymptotique. Ces résultats sont utilisés pour évaluer leur criticité sur la base d'indicateurs relatifs à leur coût global ou à leur longueur.

1. Criticité évaluée en fonction du coût global

En application des principes présentés en section 2.6.1.2, le coût global de quelques séquences d'événements, extraites des sous-langages L_1 et L_9, est évalué (une démarche analogue peut être développée pour les autres langages). Ce coût global est constitué d'un coût associé aux états et d'un coût associé aux transitions. Pour le premier, nous avons fixé un coût par unité de temps γ_i associé à chaque état x_i qui reflète les ressources dépensées/acquises lorsque le système est dans cet état (tableau 2.5). Ce coût unitaire est multiplié par la durée moyenne conditionnelle de passage de l'état x_i vers l'état x_j, évaluée à l'aide de l'équation (1.22), pour obtenir le coût du séjour dans l'état x_i. Un coût associé à chaque événement c_{ij} a également été défini et est présenté dans le tableau (2.6) : il prend des valeurs positives ou négatives selon la nature des événements. Il est à noter que les coûts unitaires des états γ_i et des événements c_{ij} sont purement arbitraires et ne sont fixés que pour illustrer l'approche proposée. L'équation (2.13) peut ensuite être appliquée afin d'obtenir le coût global d'une séquence, $C_g\left(s_i\right)$.

TABLE 2.5 – Coût associé à chaque état du système

Etat (x_i)	1	2	3	4	5	6	7	8	9
Coût/h associé à l'état (γ_i)	11	4	9	9	2	2	8	2	-11

Les résultats obtenus pour la criticité de quelques séquences appartenant au sous-langage L_9 (qui correspond à un état dangereux) sont présentés dans le tableaux (2.7). L'analyse de ces résultats démontre l'intérêt de compléter le calcul des probabilités d'occurrence par un calcul de criticité puisqu'ils conduisent à des conclusions divergentes. En effet, si l'on ne considère que la probabilité d'occurrence, la séquence s_1 apparaît comme la plus critique car sa probabilité d'occurrence est supérieur à celle de toutes les autres séquences. En revanche, lorsque l'on considère le coût global des séquences, la séquence s_8 apparaît comme la plus critique. Dans le même ordre d'idée, le coût global des séquences s_3 à s_7 est très proche de la valeur du coût global de la séquence s_2 alors que la probabilité d'occurrence de cette dernière est supérieure aux probabilités des séquences s_3 à s_7. Dans la mesure où cette analyse doit conduire à la définition de moyens de prévention permettant de réduire le risque d'occurrence des séquences les plus critiques, on peut donc en déduire que les priorités à donner, pour conduire les études visant optimiser le niveau de sûreté de fonctionnement, seront différentes selon que l'on considère la probabilité d'occurrence

TABLE 2.6 – Coût associé à chaque événement du système

Evénement e_{ij}	**Coût associé à l'événement** (e_{ij})	**Evénement** e_{ij}	**Coût associé à l'événement** (e_{ij})
e_{12}	-3	e_{46}	-2
e_{23}	-0.5	e_{59}	-0.75
e_{29}	-0.5	e_{69}	-0.5
e_{31}	8	e_{71}	10
e_{33}	8	e_{78}	-3
e_{34}	0.3	e_{89}	-0.5
e_{35}	-2	e_{94}	10
e_{41}	8	e_{97}	8
e_{43}	0.6	e_{99}	18
e_{44}	8		

TABLE 2.7 – Coût global pour quelques séquences qui amènent le système dans l'état 9

Séquence (s_i)	**Probabilité de la séquence** $(\mathbb{P}(s_i))$	**Coût de la séquence** $(C_g(s_i))$
$s_1 = e_{12}e_{29}$	0.0500	$3.1429 \cdot 10^5$
$s_2 = e_{12}e_{29}e_{99}$	0.0021	$3.1429 \cdot 10^5$
$s_3 = e_{12}e_{23}e_{35}e_{59}$	$2.3744 \cdot 10^{-4}$	$3.1877 \cdot 10^5$
$s_4 = e_{12}e_{23}e_{34}e_{46}e_{69}$	$4.1624 \cdot 10^{-11}$	$3.1951 \cdot 10^5$
$s_5 = e_{12}e_{29}e_{97}e_{78}e_{89}$	$7.5418 \cdot 10^{-6}$	$3.1614 \cdot 10^5$
$s_6 = e_{12}e_{23}e_{33}e_{35}e_{59}$	$7.1214 \cdot 10^{-6}$	$3.1877 \cdot 10^5$
$s_7 = e_{12}e_{23}e_{33}e_{34}e_{46}e_{69}$	$1.2484 \cdot 10^{-12}$	$3.1951 \cdot 10^5$
$s_8 = e_{12}e_{23}e_{33}e_{31}e_{12}e_{29}$	0.0014	$6.3197 \cdot 10^5$
$s_9 = e_{12}e_{29}e_{99}e_{97}e_{78}e_{89}e_{99}$	$1.2747 \cdot 10^{-8}$	$3.1614 \cdot 10^5$
$s_{10} = e_{12}e_{23}e_{35}e_{59}e_{94}e_{46}e_{69}$	$3.1321 \cdot 10^{-8}$	$3.2065 \cdot 10^5$

ou le coût global. Compte tenu du fait que le coût global d'une séquence inclut sa probabilité d'occurrence, il est donc préférable de privilégier une analyse basée sur ce coût.

2. Criticité évaluée en fonction de la longueur de la séquence

La deuxième manière d'évaluation de la criticité des séquences d'événements est basée sur le calcul d'un coût dépendant de la longueur $l(s_i)$ des séquences, c'est à dire du nombre d'événements qui la composent. Le coût probabiliste d'une séquence est déterminé par l'équation (2.14), sa longueur et sa probabilité d'occurrence étant supposées connues à ce stade.

Comme lors du paragraphe précédent, nous allons nous intéressé à l'analyse de la criticité des séquences appartenant au sous-langage L_9 correspondant à un état dangereux. Le tableau (2.8) présente les résultats obtenus pour quelques séquences. Les conclusions sont identiques à celles proposées précédemment puisqu'elles mettent en évidence l'intérêt de compléter les calculs de probabilités d'occurrence des séquences par un critère basé sur leur longueur. En effet, si l'on ne considère que les probabilités d'occurrence, la séquence s_2 apparaît comme plus critique que la séquence s_8 mais lorsque l'on considère également les longueurs, le classement et, par conséquent, les priorités à donner aux études de SdF s'inversent. Il est à noter que, inversement, si l'on ne considère que la longueur des séquences, la séquence s_8 apparaît moins critique que les séquences s_2 à s_6, ce qui n'est pas le cas si l'on considère à la fois les probabilités d'occurrence et la longueur.

TABLE 2.8 – Longueur et coût pour quelques séquences qui conduisent le système à l'état 9

Séquence (s_i)	**Probabilité** ($\mathbb{P}(s_i)$)	**Longueur** ($l(s_i)$)	**Coût de la séquence** ($C_l(s_i)$)
$s_1 = e_{12}e_{29}$	0.0500	2	0.1000
$s_2 = e_{12}e_{29}e_{99}$	0.0021	3	0.0063
$s_3 = e_{12}e_{23}e_{35}e_{59}$	$2.3744 \cdot 10^{-4}$	4	$9.4976 \cdot 10^{-4}$
$s_4 = e_{12}e_{23}e_{34}e_{46}e_{69}$	$4.1624 \cdot 10^{-11}$	5	$2.0812 \cdot 10^{-10}$
$s_5 = e_{12}e_{29}e_{97}e_{78}e_{89}$	$7.5418 \cdot 10^{-6}$	5	$3.7709 \cdot 10^{-5}$
$s_6 = e_{12}e_{23}e_{33}e_{35}e_{59}$	$7.1214 \cdot 10^{-6}$	5	$3.5607 \cdot 10^{-5}$
$s_7 = e_{12}e_{23}e_{33}e_{34}e_{46}e_{69}$	$1.2484 \cdot 10^{-12}$	6	$7.4904 \cdot 10^{-12}$
$s_8 = e_{12}e_{23}e_{33}e_{31}e_{12}e_{29}$	0.0014	6	0.0084
$s_9 = e_{12}e_{29}e_{99}e_{97}e_{78}e_{89}e_{99}$	$1.2747 \cdot 10^{-8}$	7	$8.9229 \cdot 10^{-8}$
$s_{10} = e_{12}e_{23}e_{35}e_{59}e_{94}e_{46}e_{69}$	$3.1321 \cdot 10^{-8}$	7	$2.1925 \cdot 10^{-7}$

2.7.3.2 Régime transitoire

Cette section porte sur l'application de la technique de détermination du temps de premier passage (proposé par Harrison *et al.* [Harrison et Knottenbelt, 2002]) dans le contexte du calcul des probabilités des séquences d'événements en régime transitoire. Cette technique sera appliquée sur un modèle markovien (pour l'étude d'une défaillance de cause commune du four) puis sur le modèle semi-markovien de la commande du four présenté en figure (2.7).

2.7.3.2.1 Application sur un modèle markovien

Pour illustrer l'application de cette technique sur un modèle markovien, considérons d'abord un seul mode de défaillance du système de contrôle de la température d'un four. Il s'agit d'une défaillance de cause commune (DCC) représentée dans la figure (2.9) comme une chaîne de Markov à temps continu. Cette DCC est décrite par le modèle de facteur β et nous supposons qu'elle peut être détectée instantanément. La DCC entraîne la défaillance simultanée des deux contrô-

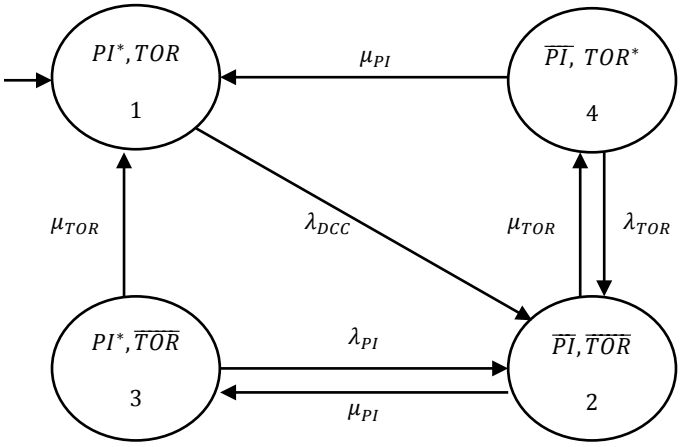


FIGURE 2.9 – Défaillance de cause commune

leurs (PI et TOR) et conduit directement à l'état de panne (2) de l'automate où la température du four n'est plus contrôlée. Selon le modèle du facteur β [Cai *et al.*, 2012], la valeur du β est donnée par le rapport entre le taux de défaillance de cause commune λ_{DCC} et la somme entre

le taux de défaillances indépendantes λ_{ind} et le taux de défaillances de cause commune λ_{DCC} (λ_{tot}) :

$$\beta = \frac{\lambda_{DCC}}{\lambda_{tot}} = \frac{\lambda_{DCC}}{\lambda_{ind} + \lambda_{DCC}} \tag{2.56}$$

Il s'en suit que le taux de défaillances de cause commune peut être exprimé sous la forme :

$$\lambda_{DCC} = \beta \lambda_{tot}, \tag{2.57}$$

et que le taux de défaillances indépendantes est donné par l'équation :

$$\lambda_{ind} = (1 - \beta)\, \lambda_{tot} \tag{2.58}$$

Pour le mode de défaillance de cause commune de notre cas d'étude, le taux d'occurrence de la DCC, λ_{DCC}, est obtenu en utilisant la relation suivante, en considérant que $\beta = 0.05$:

$$\lambda_{DCC} = \max\left(\beta \lambda_{PI}, \beta \lambda_{TOR}\right) \tag{2.59}$$

Les valeurs qui correspondent aux autres taux de transition du système sont constantes : $\lambda_{PI} = 3.5 \cdot 10^{-5}$, $\lambda_{TOR} = 2 \cdot 10^{-5} h^{-1}$, $\mu_{PI} = 8 \cdot 10^{-2} h^{-1}$, $\mu_{TOR} = 10^{-1} h^{-1}$.

Le calcul des probabilités en régime transitoire nécessite de déterminer la matrice M associée à la chaîne de Markov de la figure (2.9). Celle-ci est donnée ci-dessous :

$$M = \begin{pmatrix} -\lambda_{12} & \lambda_{12} & 0 & 0 \\ 0 & -(\lambda_{23} + \lambda_{24}) & \lambda_{23} & \lambda_{24} \\ \lambda_{31} & \lambda_{32} & -(\lambda_{31} + \lambda_{32}) & 0 \\ \lambda_{41} & \lambda_{42} & 0 & -(\lambda_{41} + \lambda_{42}) \end{pmatrix}$$

où λ_{ij} correspond au taux de transition entre les états i et j. A titre d'exemple λ_{32} correspond au taux de transition entre les états 3 et 2 et sa valeur numérique est $\lambda_{32} = \lambda_{PI} = 3.5 \cdot 10^{-5}$.

Transformée de Laplace du temps de premier passage

A partir de cette matrice, l'équation (2.15) peut être réécrite en considérant successivement les états (1), (2), (3) puis (4) comme état cible (avec $\overrightarrow{j} = \{1\}$, puis $\overrightarrow{j} = \{2\}$, puis $\overrightarrow{j} = \{3\}$ et enfin $\overrightarrow{j} = \{4\}$). La résolution de ces quatre réécritures de l'équation (2.15) permet d'obtenir les transformées de Laplace du temps de premier passage depuis chacun des quatre états vers l'état cible.

Nous donnons ci-dessous, à titre d'exemple, les équations obtenues en considérant l'état (1), puis l'état (2) comme état cible. L'intérêt de l'état (2) est qu'il correspond à une DCC entraînant la panne des deux régulateurs de température. La même démarche est appliquée pour les états cibles (3) et (4).

- Etat cible : ($\overrightarrow{j} = \{1\}$)

La réécriture de l'équation (2.15) pour ($\overrightarrow{j} = \{1\}$) donne :

$$\begin{pmatrix} s + \lambda_{12} & -\lambda_{12} & 0 & 0 \\ 0 & s + \lambda_{23} + \lambda_{24} & -\lambda_{23} & -\lambda_{24} \\ 0 & -\lambda_{32} & s + \lambda_{31} + \lambda_{32} & 0 \\ 0 & -\lambda_{42} & 0 & s + \lambda_{41} + \lambda_{42} \end{pmatrix} \begin{pmatrix} L_{11}(s) \\ L_{21}(s) \\ L_{31}(s) \\ L_{41}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \lambda_{31} \\ \lambda_{41} \end{pmatrix}$$

La résolution de ce système à quatre inconnues fournit les transformées de Laplace du temps de premier passage depuis chaque état vers l'état (1) :

$$L_{11}(s) = \frac{\lambda_{12}}{(s+\lambda_{12})} \cdot L_{21}(s)$$

$$L_{21}(s) = \frac{\frac{\lambda_{23}\lambda_{31}}{(s+\lambda_{31}+\lambda_{32})} + \frac{\lambda_{24}\lambda_{41}}{(s+\lambda_{41}+\lambda_{42})}}{(s+\lambda_{23}+\lambda_{24}) - \frac{\lambda_{23}\lambda_{32}}{(s+\lambda_{31}+\lambda_{32})} - \frac{\lambda_{24}\lambda_{42}}{(s+\lambda_{41}+\lambda_{42})}}$$

$$L_{31}(s) = \frac{\lambda_{31} + \lambda_{32}L_{21}(s)}{(s+\lambda_{31}+\lambda_{32})}$$

$$L_{41}(s) = \frac{\lambda_{41} + \lambda_{42}L_{21}(s)}{(s+\lambda_{41}+\lambda_{42})}$$

- Etat cible : ($\overrightarrow{j} = \{2\}$)

La réécriture de l'équation (2.15) pour ($\overrightarrow{j} = \{2\}$) donne :

$$\begin{pmatrix} s+\lambda_{12} & 0 & 0 & 0 \\ 0 & s+\lambda_{23}+\lambda_{24} & -\lambda_{23} & -\lambda_{24} \\ -\lambda_{31} & 0 & s+\lambda_{31}+\lambda_{32} & 0 \\ -\lambda_{41} & 0 & 0 & s+\lambda_{41}+\lambda_{42} \end{pmatrix} \begin{pmatrix} L_{12}(s) \\ L_{22}(s) \\ L_{32}(s) \\ L_{42}(s) \end{pmatrix} = \begin{pmatrix} \lambda_{12} \\ 0 \\ \lambda_{32} \\ \lambda_{42} \end{pmatrix}$$

La résolution de ce système à quatre inconnues fournit les transformées de Laplace du temps de premier passage depuis chaque état vers l'état (2) :

$$L_{12}(s) = \frac{\lambda_{12}}{(s+\lambda_{12})}$$

$$L_{22}(s) = \frac{\lambda_{24}L_{42}(s) + \lambda_{23}L_{32}(s)}{(s+\lambda_{23}+\lambda_{24})}$$

$$L_{32}(s) = \frac{\lambda_{32} + \lambda_{31}L_{12}(s)}{(s+\lambda_{31}+\lambda_{32})}$$

$$L_{42}(s) = \frac{\lambda_{42} + \lambda_{41}L_{12}(s)}{(s+\lambda_{41}+\lambda_{42})}$$

Fonctions de répartition

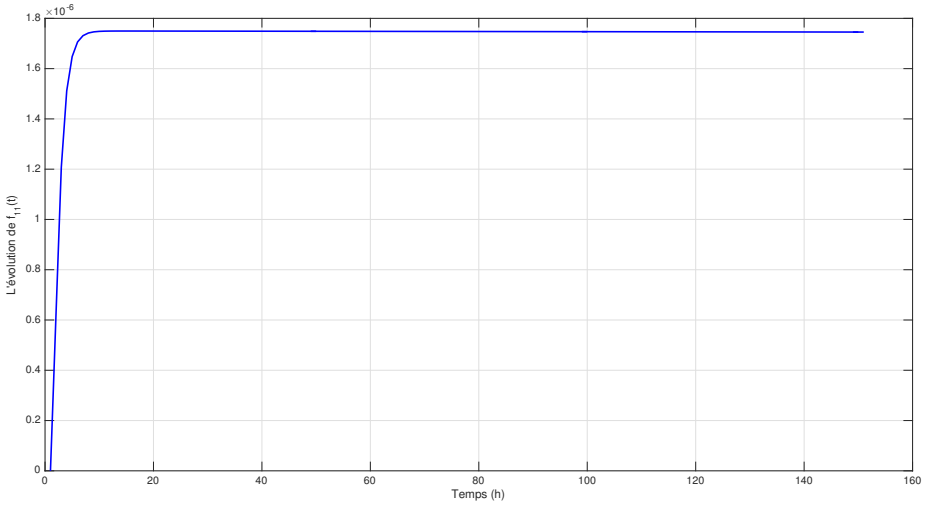
A partir de ces transformées de Laplace, l'objectif est d'obtenir les densités de probabilités $f_{11}(t)$ puis les fonctions de répartition $F_{11}(t)$. Partant de l'hypothèse qu'à l'instant initial le système se trouve dans l'état (1) où les deux régulateurs sont fonctionnels, nous allons focaliser les évaluations de probabilités en régime transitoire sur les temps de premier de passage depuis l'état (1) vers lui-même, puis vers l'état (2), puis l'état (3) et enfin l'état (4).

La démarche suivie est celle présentée en section 2.6.2.1. La densité de probabilité $f_{1i}(t)$ est obtenue en inversant numériquement la transformée de Laplace $L_{1i}(s)$ à l'aide de la fonction Matlab *ilaplace*. La fonction de répartition $F_{1i}(t)$ est alors obtenue à partir de ces densités de probabilités à l'aide de la fonction Matlab *trapz* (surface en dessous de la courbe). Comme indiqué en section 2.6.2.1, l'ensemble de ces calculs est supporté par un script Matlab qui permet

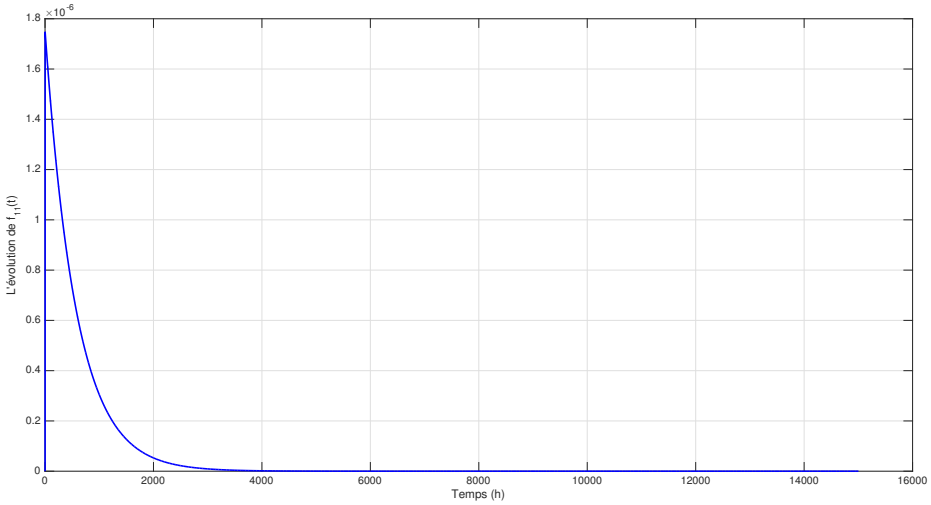
d'effectuer plusieurs calculs sur des intervalles de temps différents. Les résultats obtenus pour deux fonctions de répartition, $F_{11}(t)$, $F_{12}(t)$ sont présentés et discutés ci-dessous.

- $F_{11}(t)$

Le premier calcul a été effectué pour une durée $t = 1500$ heures et la dernière avec une durée $t = 15 \cdot 10^6$ heures. Les évolutions de la densité de probabilité, dans ces deux cas, $f_{11}(t)$ sont présentés sur les figures (2.10a) et (2.10b).



(a) Calcul effectué sur sur $15 \cdot 10^2$ heures



(b) Calcul effectué sur sur $15 \cdot 10^6$ heures

FIGURE 2.10 – Densité de probabilité du temps de premier passage de l'état 1 vers lui même

Pour un calcul sur une durée $t = 1500$ heures, la valeur numérique de la fonction de répartition $F_{11}(t)$ est :

$$F_{11}(t) = 0.0026$$

et, pour une durée $t = 15 \cdot 10^6$ heures, elle devient :

$$F_{11}(t) = 0.9992$$

Lorsque l'on augmente l'intervalle de temps, la valeur de $F_{11}(t)$ augmente jusqu'à se stabiliser à 1. Ce résultat s'explique par le fait que plus le temps de augmente, plus le système se rapproche de son régime asymptotique pour lequel la somme des probabilités des séquences appartenant aux sous-langage L_1 est égale à 1. En ce sens, les résultats obtenus en régime transitoire sont cohérents avec ceux obtenus en régime asymptotique à l'aide de la théorie de langages probabilistes.

- $F_{12}(t)$

La fonction de répartition $F_{12}(t)$ est particulièrement intéressante, sur cet exemple, du point de vue de la sûreté de fonctionnement puisque l'état (2) correspond à un état dangereux où les deux contrôleurs sont défaillants. Le premier calcul a été effectué pour une durée $t = 15 \cdot 10^3$ heures et la dernière avec une durée $t = 15 \cdot 10^6$ heures. Les évolutions de la densité de probabilité, dans ces deux cas, $f_{12}(t)$ sont présentés sur les figures (2.11a) et (2.11b).

Pour un calcul sur une durée $t = 15 \cdot 10^3$ heures, la valeur numérique de la fonction de répartition $F_{12}(t)$ est :

$$F_{12}(t) = 0.0259$$

et, pour une durée $t = 15 \cdot 10^6$ heures, elle devient :

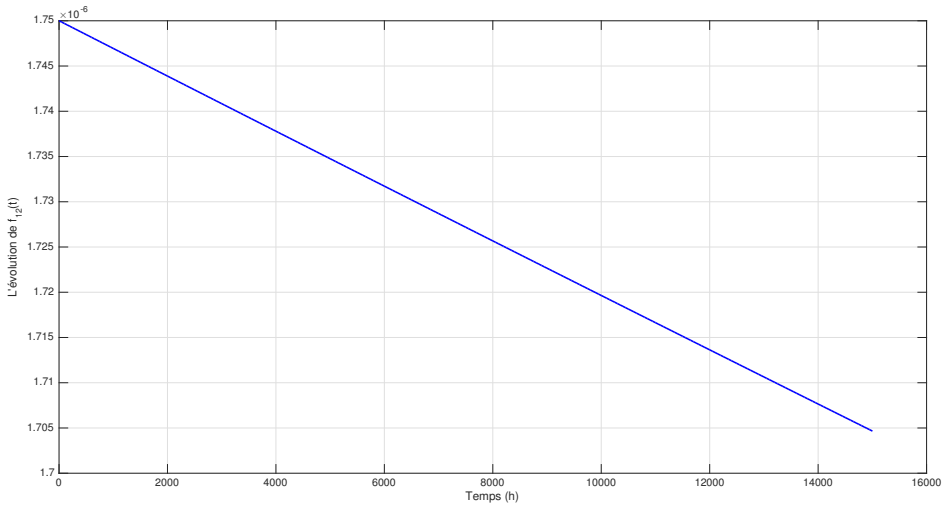
$$F_{12}(t) = 0.5500$$

La tendance observée pour l'évolution de $F_{12}(t)$, en fonction de l'intervalle de temps considéré pour effectuer le calcul, est similaire à celle observée pour $F_{11}(t)$: lorsque que la durée augmente, la valeur de $F_{12}(t)$ suit la même tendance. Le résultat obtenu pour $15 \cdot 10^6$ heures ($F_{12}(t) = 0.5500$) est proche de la valeur obtenue par le calcul en régime asymptotique pour la somme des probabilités des séquences qui amènent le système depuis son état initial à l'état de panne (0.5557). Ceci qui est cohérent avec le fait, que plus la durée augmente, plus le système tend vers son régime asymptotique, et contribue à valider notre approche.

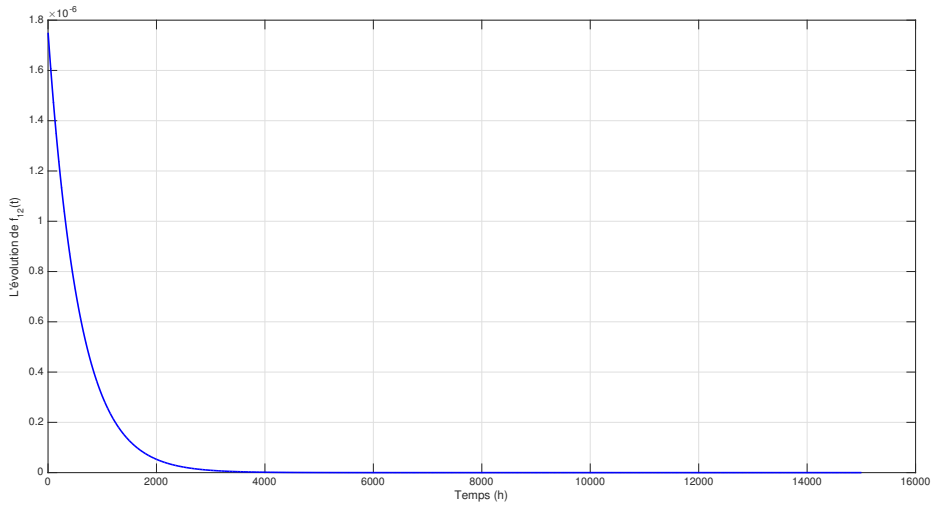
- Généralisation

De la même manière, plusieurs calculs ont été réalisés pour obtenir les densités de probabilités $f_{13}(t)$ et $f_{14}(t)$ et en déduire les fonctions de répartitions $F_{13}(t)$ et $F_{14}(t)$. Nous ne donnons ici que le résultat obtenu pour $F_{13}(t) = 0.0013$ sur la base d'une durée de 1500 heures et pour $F_{14}(t) = 0.1957$ sur la base d'une durée de $15 \cdot 10^3$ heures. En effet, la démarche est strictement identique et les résultats ne font l'objet d'aucune interprétations supplémentaires, si ce n'est qu'ils confirment que lorsque l'intervalle de temps s'accroît et que le système tend vers son régime asymptotique, les valeurs numériques des fonctions de répartition convergent vers les valeurs obtenues en régime asymptotique pour la somme des probabilités des séquences conduisant le système depuis l'état initial vers les états cibles. Ainsi, les calculs effectués en régime asymptotique peuvent être considérés comme une validation pour les probabilités des séquences en régime transitoire obtenues suite à l'application de la méthode de Harrison *et al.* [Harrison et Knottenbelt, 2002].

Une autre approche de validation, basée sur l'utilisation de l'outil industriel KB3/FigSeq, est proposée dans le paragraphe suivant.



(a) Calcul effectué sur sur $15 \cdot 10^3$ heures



(b) Calcul effectué sur sur $15 \cdot 10^6$ heures

FIGURE 2.11 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 2

Validation numérique des résultats par KB3/FigSeq

Comme nous l'avons indiqué au début de la section 2.6.2, l'outil FigSeq, développé par EDF, permet le calcul de probabilités de séquences en régime transitoire. Pour atteindre cet objectif, FigSeq utilise en entrée une description dans un format neutre (dans le langage Figaro0 basés sur des modèles probabilistes états-transitions) qui peut être importée de l'outil KB3, également développé par EDF, et qui permet l'édition de modèles sous la forme de BDMP ou de réseaux de Petri. Différents algorithmes sont appliqués par FigSeq pour analyser les séquences d'événements qui peuvent se produire dans le système. L'utilisateur a également la possibilité d'établir un certain nombre des critères pour l'exploration de séquences et le calcul de leur probabilités d'occurrence.

Le mode de défaillance de cause commune du système de commande de la température a été représenté en KB3 sous la forme d'une réseau de Petri pour lequel la somme des marquages de toutes ses places est toujours égale à 1 (autrement dit sous la forme d'un automate à états finis). Une fois importée dans FigSeq (via sa transformation au format Figaro0), plusieurs simulations ont été effectuées sous FigSeq afin d'évaluer les probabilités des séquences d'événements en régime transitoire. Ces simulations ont été réalisées en se basant sur l'algorithme *"Sequences Normales"*, c'est-à-dire sur un algorithme ayant pour objectif l'analyse des toutes les séquences se produisant sur un intervalle de temps donné (y compris celles conduisant à plusieurs passage par le même état).

Les intervalles de temps ont été choisis à l'identique de ceux utilisés dans les calculs réalisés sous Matlab et nous avons sélectionné les mêmes états cibles. La comparaison des résultats obtenus montre que les valeurs numériques que nous avons obtenues pour la fonction de répartition du temps de premier passage dans les calculs Matlab coïncident avec les probabilités d'occurrence des séquences obtenues avec FigSeq en appliquant l'algorithme *"Sequences Normales"*.

2.7.3.2.2 Application sur un modèle semi-markovien

Pour illustrer la démarche proposée pour l'évaluation de probabilités d'occurrence des séquences d'événements sur un modèle semi-markovien, nous reprenons le modèle du système de commande de la température du four (figure 2.7). Il est représenté sous la forme d'un automate à 9 états et plusieurs transitions. On note ici que la détection de la défaillance du régulateur PI se réalise dans une durée de temps d_{smax} heures et que le système reste dans l'état 2 tant que la détection n'est pas réalisée. Ainsi, afin de s'assurer que le système évolue de telle manière à ne pas s'arrêter à l'état 2, les intervalles de temps choisis pour effectuer les calculs auront comme borne inférieure la valeur de d_{smax} heures.

Tout d'abord, les éléments du noyau, $Q_{ij}(t)$, du modèle semi-markovien sont déterminés en sachant que :

$$Q_{ij}(t) = \int_0^t \prod_{k \neq j} (1 - F_{ik}(u))\, f_{ij}(u)\, du \tag{2.60}$$

Après avoir déterminé chaque élément du noyau, la transformée Laplace-Stieltjes de $Q_{ij}(t)$ est calculée à l'aide de l'équation (2.20).

A titre d'exemple, nous allons considérer un élément du noyau qui correspond à une transition exponentielle et un élément correspondant à une transition déterministe et nous présenterons, pour chacun d'entre eux, les résultats obtenus pour la transformée Laplace-Stieltjes $r_{ij}^*(s)$.

La transition exponentielle considérée est celle qui amène le système de l'état 1 (état initial) à l'état 2. L'élément du noyau lui correspondant est alors :

$$Q_{12}(t) = 1 - e^{-\lambda_{PI} t}$$

Sachant que le temps de séjour dans l'état 1 est déterminé par l'équation (2.18), le résultat obtenu pour la transformée Laplace-Stieltjes est :

$$r_{12}^*(s) = \frac{\lambda_{PI}}{s + \lambda_{PI}}$$

La transition déterministe considérée est celle qui amène le système de l'état 2 à l'état 3. Ainsi, l'élément du noyau qui lui correspond est :

$$Q_{23}(t) = (1 - p_{refTOR}) \begin{cases} 0, & t < d_{smax} \\ 1, & t \geq d_{smax} \end{cases}$$

Le temps de séjour dans l'état 2 avant que la transition e_{23} ne se produise est donné, cette fois, par une valeur constante :

$$S_{23} = \begin{cases} 0, & t < d_{smax} \\ d_{smax}, & t \geq d_{smax} \end{cases}$$

Ainsi, la transformée Laplace-Stieltjes associée à cette transition est :

$$r^*_{23}(s) = p_{23} \begin{cases} 1, & t < d_{smax} \\ e^{-sd_{smax}}, & t \geq d_{smax} \end{cases}$$

Les éléments de noyau et les transformées Laplace-Stieltjes afférentes aux autres transitions exponentielles ou déterministes, présentes dans le modèle, seront déterminés de manière analogue à celle utilisé pour les deux transitions détaillées au-dessus. Les transformées Laplace-Stieltjes connues, il est alors possible de procéder à la détermination des transformées de Laplace des temps de premier passage depuis n'importe quel état vers un état cible puis d'obtenir les densités de probabilités et les fonctions de répartition. Dans les exemples développés ci-dessous, nous avons considéré deux états cibles : $\overrightarrow{j} = \{1\}$ et $\overrightarrow{j} = \{9\}$ (ce dernier correspondant à un état dangereux).

Les transformées de Laplace des temps de premier passage sont obtenues par réécriture de l'équation (2.19).

Le système d'équations obtenu pour $\overrightarrow{j} = \{1\}$ est le suivant :

$$\begin{pmatrix} 1 & -r^*_{12}(s) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -r^*_{23}(s) & 0 & 0 & 0 & 0 & 0 & -r^*_{29}(s) \\ 0 & 0 & 1 & -r^*_{34}(s) & -r^*_{35}(s) & 0 & 0 & 0 & 0 \\ 0 & 0 & -r^*_{43}(s) & 1 & 0 & -r^*_{46}(s) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -r^*_{59}(s) \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -r^*_{69}(s) \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -r^*_{78}(s) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -r^*_{89}(s) \\ 0 & 0 & 0 & -r^*_{94}(s) & 0 & 0 & -r^*_{97}(s) & 0 & 1 \end{pmatrix} \begin{pmatrix} L_{11}(s) \\ L_{21}(s) \\ L_{31}(s) \\ L_{41}(s) \\ L_{51}(s) \\ L_{61}(s) \\ L_{71}(s) \\ L_{81}(s) \\ L_{91}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ r^*_{31}(s) \\ r^*_{41}(s) \\ 0 \\ 0 \\ r^*_{71}(s) \\ 0 \\ 0 \end{pmatrix}$$

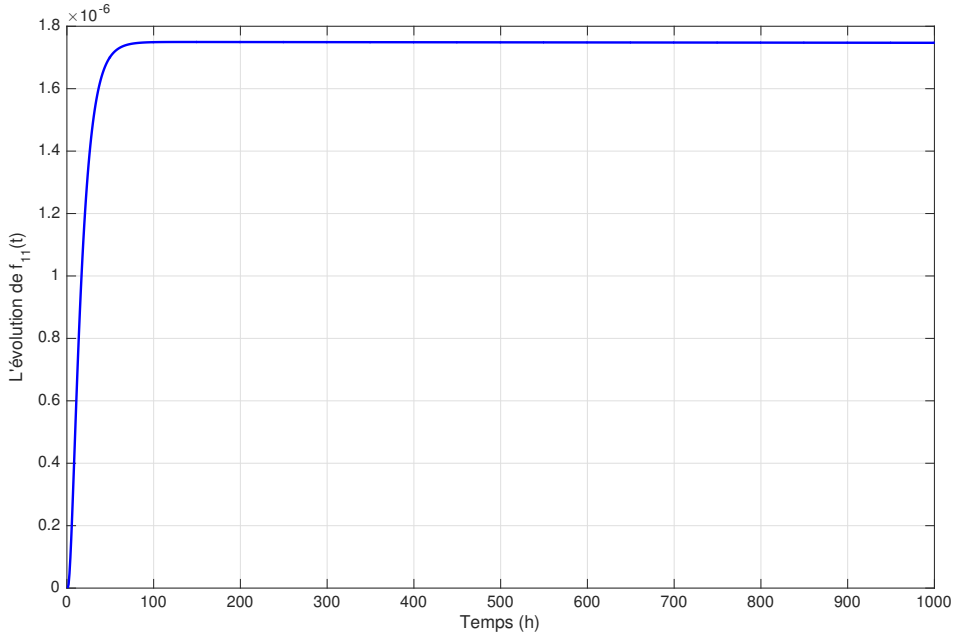
celui obtenu pour $\overrightarrow{j} = \{9\}$ est :

$$\begin{pmatrix} 1 & -r^*_{12}(s) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -r^*_{23}(s) & 0 & 0 & 0 & 0 & 0 & 0 \\ -r^*_{31}(s) & 0 & 1 & -r^*_{34}(s) & -r^*_{35}(s) & 0 & 0 & 0 & 0 \\ -r^*_{41}(s) & 0 & -r^*_{43}(s) & 1 & 0 & -r^*_{46}(s) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -r^*_{71}(s) & 0 & 0 & 0 & 0 & 0 & 1 & -r^*_{78}(s) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -r^*_{94}(s) & 0 & 0 & -r^*_{97}(s) & 0 & 1 \end{pmatrix} \begin{pmatrix} L_{19}(s) \\ L_{29}(s) \\ L_{39}(s) \\ L_{49}(s) \\ L_{59}(s) \\ L_{69}(s) \\ L_{79}(s) \\ L_{89}(s) \\ L_{99}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ r^*_{29}(s) \\ 0 \\ 0 \\ r^*_{59}(s) \\ r^*_{69}(s) \\ 0 \\ r^*_{89}(s) \\ 0 \end{pmatrix}$$

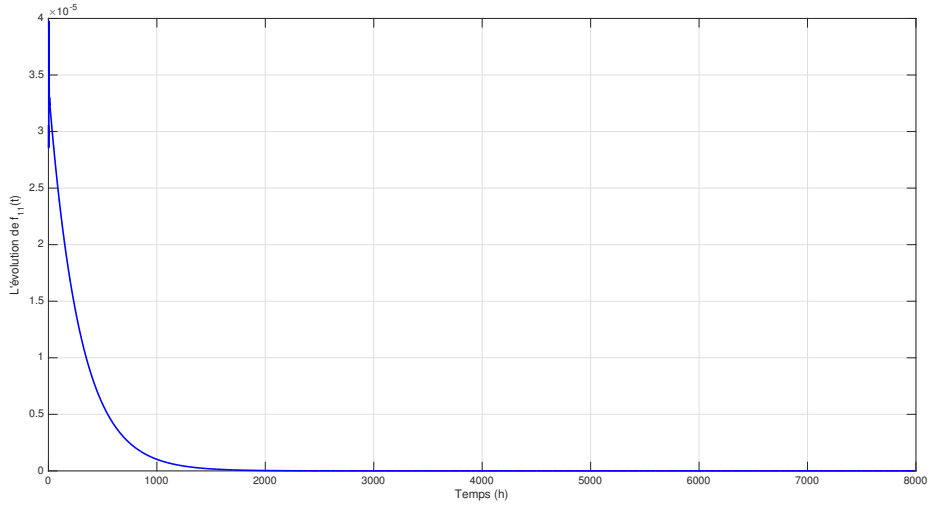
La résolution de ces deux systèmes d'équations permet d'obtenir les transformées de Laplace du temps de premier passage depuis chacun des états vers respectivement les états (1) et (9). Compte tenu de la taille de ces systèmes d'équations, les expressions obtenues pour $L_{i1}(s)$ et $L_{i9}(s)$ sont très complexes et nous ne les présentons pas ici. Néanmoins, elles sont exploitées dans le cadre des calculs réalisés sur Matlab pour obtenir les densités de probabilités puis les fonctions de répartition. Pour les mêmes raisons que celles invoquées lors des calculs réalisés en régime transitoire sur des modèles markoviens, nous ne considérons que les temps de premier passage depuis l'état initial, c'est à dire l'état (1), vers les états cibles (1) et (9).

La démarche consiste à obtenir les densités de probabilité $f_{11}(t)$ et $f_{19}(t)$ en inversant de manière numérique les transformées de Laplace $L_{11}(s)$ et $L_{19}(s)$ à l'aide de la fonction Matlab *euler_ inversion*. A partir de ces densités, la fonction Matlab *trapz* permet d'obtenir les fonctions de répartition $F_{11}(t)$ et $F_{19}(t)$. Ces calculs sont réalisés par un script Matlab sur plusieurs intervalles de temps de plus en plus larges.

Pour le temps de premier passage de l'état initial vers lui même, nous présentons ici les résultats obtenus pour une durée de 1000 heures $[10^3, 2 \cdot 10^3]$ puis de $799 \cdot 10^3$ heures. Les évolutions de $f_{11}(t)$ pour ces deux durées sont présentées dans les figures (2.12a) et (2.12b).



(a) Calcul effectué sur 10^3 heures



(b) Calcul effectué sur $799 \cdot 10^3$ heures

FIGURE 2.12 – Densité de probabilité du temps de premier passage de l'état 1 vers lui-même

Les valeurs numériques correspondantes de la fonction de répartition $F_{11}(t)$ sont pour les calculs effectués sur une durée de 10^3 heures et de $799 \cdot 10^3$:

$$F_{11}(t) = 0.0017,\ pour\ t = 10^3\ heures, \qquad F_{11}(t) = 0.9657,\ pour\ t = 799 \cdot 10^3\ heures$$

Comme attendu, plus l'intervalle de temps s'élargit, plus la valeur de la fonction de répartition $F_{11}(t)$ s'approche de 1 et finit par se stabiliser autour de 1 pour des intervalles très grand. Ce résultat est identique à celui obtenu pour la somme de probabilités des séquences appartenant au sous-langage L_1 en régime asymptotique.

Pour le temps de premier passage depuis l'état initial vers l'état 9 où le système est défaillant (car les deux régulateurs sont en panne), nous présentons ici les résultats obtenus sur un intervalle de 1000 heures $[10^3, 2 \cdot 10^3]$ puis de $8 \cdot 10^7$ heures. Les évolutions de $f_{19}(t)$ sur ces deux intervalles sont présentées dans les figures (2.13a) et (2.13b).

Les valeurs numériques de la fonction de répartition $F_{19}(t)$ pour des durées de temps de 10^3 heures et de $8 \cdot 10^7$ heures sont :

$$F_{19}(t) = 0.0011,\ pour\ t = 10^3\ heures, \qquad F_{19}(t) = 0.0520,\ pour\ t = 8 \cdot 10^7\ heures$$

L'observation de ces résultats montre que lorsque l'on élargit l'intervalle de temps, la valeur de la fonction de répartition $F_{19}(t)$ se stabilise autour de 0.0520. Ce résultat est conforme à celui obtenu pour la somme des probabilités des séquences d'événements appartenant au sous-langage L_9 en régime asymptotique, ce qui constitue un argument supplémentaire en faveur de la validation de la technique proposée pour le calcul de probabilités de séquences en régime transitoire.

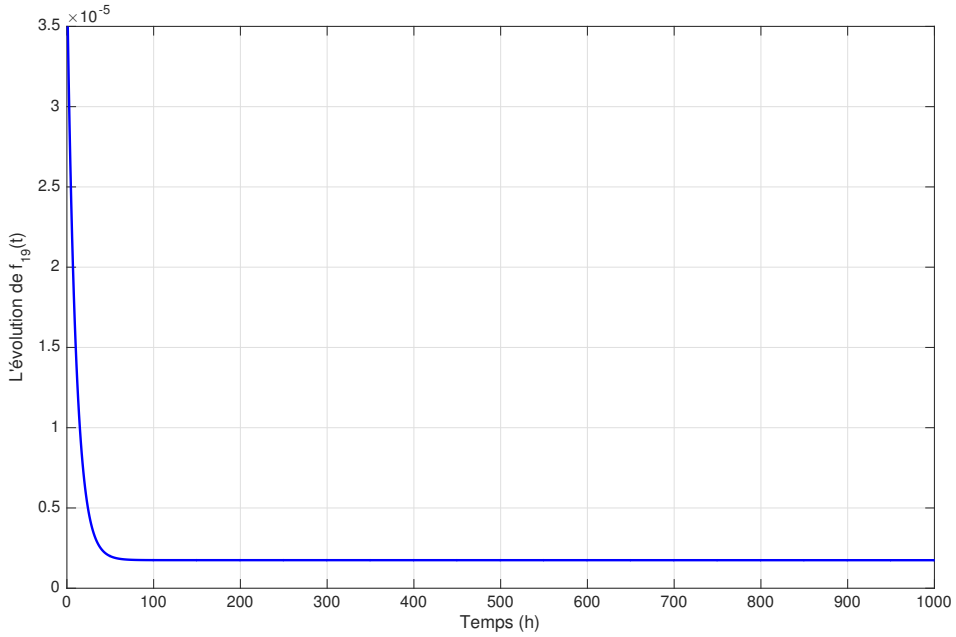
De la même manière que pour les résultats obtenus pour le régime transitoire sur des modèles markoviens, les résultats obtenus sur le modèle semi-markovien ont été validés à l'aide de l'outil KB3/YAMS. YAMS est un outil développé par EDF pour l'analyse de modèles semi-markoviens. Le modèle de la chaîne de Markov à temps continu, édité sur l'outil *KB3*, a été importé sur YAMS via le format *Figaro0* puis simulé en prenant les mêmes intervalles de temps que ceux utilisés pour les calculs Matlab. La comparaison des résultats obtenus est probante et constitue également une validation de l'approche que nous avons proposée.

2.8 Conclusion

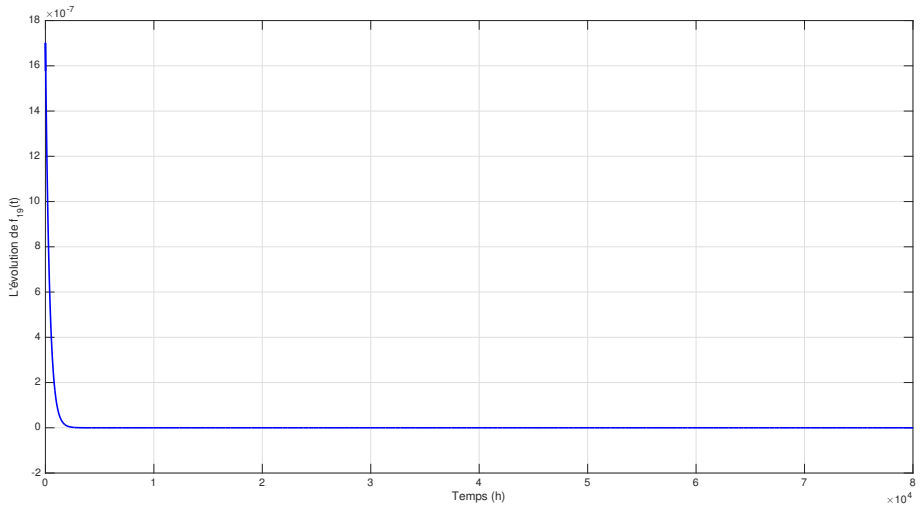
Notre contribution dans le cadre de ce chapitre porte sur la proposition d'un cadre formel, basé sur la théorie des langages probabilistes de Garg *et al.*, pour l'identification et l'évaluation quantitative de séquences d'événements, en régime asymptotique et en régime transitoire. Les applications de cette théorie, notamment dans le cadre de la synthèse de la commande, supposent que les langages (c'est à dire les séquences d'événements) ainsi que leurs probabilités sont a priori connus.

Cette hypothèse, non adaptée à la sûreté de fonctionnement, nous a conduit à proposer une démarche qui, partant d'un modèle comportemental du système, permet :

- d'extraire les séquences d'événements amenant le système dans un état quelconque (absorbant ou non) à l'aide de la théorie des langages rationnels ;
- de quantifier, en régime asymptotique, les probabilités d'occurrence des séquences identifiées, en particulier des séquences critiques ; le calcul est réalisé de manière analytique à partir d'un p-automate obtenu en utilisant la technique de la chaîne de Markov immergée ; toujours en régime asymptotique, le calcul des probabilités d'occurrence est complété par un calcul de criticité basé sur le coût global des séquences ou leur longueur ;



(a) Calcul effectué sur 10^3 heures



(b) Calcul effectué sur $8 \cdot 10^7$ heures

FIGURE 2.13 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 9

- de quantifier, en régime transitoire, les probabilités d'occurrence des séquences sur des modèles markoviens ou semi-markoviens, à l'aide de la méthode de détermination du temps de premier passage proposé par Harrison *et al.* ; l'approche repose sur la détermination de la transformée de Laplace du temps de premier passage pour ensuite, en l'inversant, obtenir la densité de probabilité puis la fonction de répartition qui correspond finalement à la probabilité que la séquence se produise pendant le temps de premier passage d'un état (i) à un état (j).

En régime asymptotique, la validation de l'approche proposée a été réalisée analytiquement et numériquement en démontrant que la probabilité d'un état cible définie comme le produit entre la probabilité de l'état initial et la probabilité de toutes les séquences qui amènent le système depuis l'état initial à l'état cible (obtenue à l'aide de notre approche) est identique à la probabilité d'un état cible obtenue par la théorie classique des chaînes de Markov.

Plus généralement, la deuxième partie de ce chapitre consacrée à l'application de notre approche sur un cas d'étude de taille réduite a permis de, au delà des aspects illustratifs, valider l'approche proposée pour le calcul de probabilités en régime asymptotique mais aussi en régime transitoire.

Si notre approche s'est révélée efficace pour le cas d'étude considéré, elle a montré quelques limites, notamment pour l'identification et le traitement des expressions régulières caractérisant les différents sous-langages de notre modèle. Celles-ci pourraient devenir rédhibitoires pour le traitement de systèmes de grande taille à échelle industrielle. Pour faire face à cette limite, le chapitre suivant est consacré à la définition d'une approche modulaire, basée sur des opérateurs de composition issus de la théorie des langages. L'objectif est d'autoriser l'évaluation de séquences d'événements d'un système complexe à partir de l'évaluation de séquences réalisées sur des modèles locaux de taille réduite sans nécessiter la construction explicite et le traitement d'un modèle global du système.

Chapitre 3

Approche modulaire pour l'évaluation des séquences d'événements

3.1 Introduction

Au cours du chapitre précédent, nous avons proposé un cadre formel pour la détermination et l'évaluation quantitative des séquences d'événements, complétée par une analyse de leur criticité en fonction de leur coût global ou en fonction de leur longueur. L'approche proposée a été appliquée sur un système (considéré comme cas d'étude) de taille réduite dont la représentation par un automate à états finis se limite à une dizaine d'états et un vingtaine de transitions. Sur cet exemple, notre approche s'est révélée efficace et n'a présenté aucune difficulté majeure.

En revanche, l'application sur ce cas d'étude a mis en évidence la complexité inhérente à l'identification et le traitement des expressions régulières associées aux sous-langages caractérisant le comportement du système. Il est clair que le risque d'être confronté à un problème d'explosion combinatoire, lorsque la taille des modèles va augmenter, est important. Cela risque en particulier lorsque l'on va s'intéresser à des systèmes industriels dynamiques réparables, reconfigurables, ayant de multiples modes de défaillance et/ou possédant de multiples stratégies de reconfiguration ou de maintenance. Ces systèmes sont en effet caractérisés par la présence de plusieurs modes de fonctionnement qui possèdent leur propre logique de fonctionnement et de commande mais qui doivent être intégrés au sein d'une stratégie globale de pilotage du système gérant les priorités et les commutations entre modes (en fonction des variables physiques qui décrivent l'état du système, des états de fonctionnement des composants, d'interventions humaines, ...). La taille et la complexité des modèles décrivant ce type de système va donc augmenter considérablement (une dizaine d'automates, une centaine d'états, plus de deux cents transitions pour le cas d'étude industriel du chapitre 4 qui reste pourtant une version simplifiée du système réel). De plus, cette forte variabilité comportementale aura des impacts importants sur les modèles et outils de la sûreté de fonctionnement dans le sens où la structure fiabiliste est amenée à évoluer dans le temps.

Dans ce contexte, la complexité des systèmes ayant de multiples modes de défaillance et/ou possédant de multiples stratégies de reconfiguration ou de maintenance va constituer un frein à la mise en œuvre de l'approche proposée au chapitre précédent dans la mesure où les phénomènes d'explosion du nombre d'états va poser des problèmes d'identification des séquences d'événements dans le cadre de la théorie des langages rationnels et, dans certains cas, d'analyse probabiliste.

Le principal objectif de ce chapitre est donc de compléter le cadre formel présenté au chapitre précédent par la proposition d'une approche modulaire autorisant la modélisation d'un système

complexe sous la forme d'un ensemble d'automates caractérisant localement un mode de fonctionnement et/ou de défaillance. A partir de ces modèles locaux, des opérateurs de composition permettront d'obtenir les propriétés globales du système à partir de ses propriétés locales. En particulier, l'identification et l'évaluation globale des séquences d'événements sera réalisée à partir des identifications et évaluations réalisées sur les modèles locaux. L'intérêt est de ne pas avoir à construire et exploiter un modèle global du système.

La première section de ce chapitre dresse un rapide état de l'art des approches modulaires (compositionnelle) basées sur des modèles états/transitions et de leurs applications probabiliste. La section suivante présente le cadre formel de l'approche modulaire que nous proposons qui repose sur les opérateurs de choix et de concaténation définis dans le cadre de la théorie des langages probabilistes [Garg *et al.*, 1999]. Des exemples d'applications de ces deux opérateurs sont ensuite présentés sur une extension du cas d'étude du four permettant d'illustrer les principes de composition entre deux modes de fonctionnement ou de défaillances. Enfin, la dernière section du chapitres est consacrée à une généralisation des opérateurs de composition à un contexte multi-modes.

3.2 Approches modulaires en sûreté de fonctionnement

La plupart des méthodes de modélisation et d'analyse en sûreté de fonctionnement, présentées dans la deuxième partie du chapitre 1, ont été définies dans un cadre de modélisation monolithique. En effet, ces approches, qu'elles soient conduites par des raisonnements basés sur les *événements* (*e.g.* les arbres d'événements [Papazoglou, 1998]) ou sur les *états* (e.g. processus markoviens ou semi-markoviens [Csenki, 1995], [Hawkes et Sykes, 1990], [Perman *et al.*, 1997], les Boolean logic Driven Markov Process (BDMP) [Bouissou et Bon, 2003]), ne présentent pas, dans leur formulation initiale, de mécanismes natifs dédiés à la modularité.

Ce constat peut s'expliquer assez facilement par la complexité inhérente à la définition d'opérateurs de synchronisation ou de composition dans un cadre temporisé et/ou probabiliste. Néanmoins, face à la complexité croissante des systèmes et de leurs modélisation, les apports de la modularité ne sont plus à démontrer et ont justifié le développement d'approches permettant la modélisation des systèmes en les décomposant de manière fonctionnelle (selon les fonctions supportées par le système, les modes de défaillances, etc.) ou structurelle (selon l'architecture du système et des composants).

Dans le domaine des modèles non probabilistes, de nombreux travaux ont eu pour objet de définir des opérateurs de synchronisation et/ou de composition sur des modèles à états/transitions temporisés et/ou temporels. A titre d'exemple, nous pouvons citer la composition de réseaux de Petri temporels [Peres *et al.*, 2011], d'automates synchrones [Maraninchi, 1992] ou encore de logiques temporelles (CTL et LTL) couplées avec des machines de Moore pour des applications en vérification de propriétés [Grumberg et Long, 1991]. Ces approches n'abordant pas la composition de modèles probabilistes, leurs applications en sûreté de fonctionnement demeurent extrêmement limitées.

Les travaux portant sur la composition ou la synchronisation de modèles probabilistes sont beaucoup moins nombreux. Le développement des chaînes de Markov interactives [Hermanns, 2002] apportent une réponse pour l'analyse de la SdF de systèmes caractérisés par des commutations de modes en étendant les CTMC classiques en les couplant à des systèmes de transitions labellisés. Elles nécessitent cependant la description complète et globale du système au sein d'un modèle unique. Les travaux de [Delahaye *et al.*, 2010] introduisent un formalisme modulaire dédié à l'analyse de systèmes combinant des aspects stochastiques et/ou non-déterministes. Il repose

sur la formalisation du concept de contrat probabiliste, adossé à une représentation sous la forme d'une DTMC, qui permet de distinguer les hypothèses faites sur un système (les garanties) de celles faites sur son environnement (les hypothèses). Des opérateurs de composition parallèle et de conjonction sont définis sur les contrats probabilistes. Si cette approche apparaît séduisante en termes de modularité, elle reste essentiellement dédiée à la vérification formelle de propriétés pour des problèmes de *model-checking*.

D'autres approches ont proposé de contourner la difficulté inhérente à la composition formelle des modèles en proposant des approches basées sur la définition d'une sémantique opérationnelle autorisant la synchronisation des modèles (par messages, par réécriture de variables, ...) en phase de simulation [Chraibi, 2013].

Enfin, dans le cadre de la théorie des langages probabilistes, différentes opérations sur les langages et les calculs de probabilités associés ont été définis par [Garg *et al.*, 1999]. Ces opérateurs de composition ouvrent des voies prometteuses pour lever le verrou relatif à la composition des modèles et proposer une approche modulaire pour l'identification et l'évaluation probabiliste de séquences critiques. En ce sens, ils constituent un argument supplémentaire justifiant notre choix des langages probabilistes.

3.3 Approche modulaire (compositionnelle) basée sur la théorie des langages probabilistes

3.3.1 Principe général

L'approche modulaire que nous proposons s'applique à l'identification et la quantification de séquences d'événements de systèmes présentant de multiples modes de fonctionnement et de défaillances (défaillance indépendante, défaillances de cause commune, détection immédiate/différée de défaillance, etc.) ainsi que de multiples stratégies de commande ou de maintenance (réparation parfaite : le système réparé est "*as good as new*", réparation imparfaite : les caractéristiques comportementales et de sûreté de fonctionnement du composant réparé sont dégradées par rapport à celles du composant neuf). Dans la pratique industrielle, les études de sûreté de fonctionnement concernant ces systèmes sont souvent réalisées de manière distincte et découplée (par exemple en mode nominal, en mode dégradé, ...), compte tenu de la taille et de la complexité des modèles sous-jacents. Pour lever cette limite par la mise en œuvre d'une approche modulaire, il est nécessaire de disposer de mécanismes formels permettant d'intégrer les résultats obtenus localement pour chaque mode au sein d'une analyse globale. Dans ce contexte, l'identification et la quantification des séquences d'événements seront réalisées en deux étapes :

- premièrement, une évaluation **locale** de la probabilité d'occurrence de séquences d'événements critiques, correspondant à chaque mode de défaillance ou à chaque politique de maintenance, est effectuée ;
- la deuxième étape porte sur l'évaluation **globale** de la probabilité d'occurrence de séquences critiques, correspondant à l'ensemble des modes de défaillances et des comportements nominaux/dégradés, à l'aide des opérateurs de choix et de concaténation définis formellement sur les langages probabilistes [Garg *et al.*, 1999].

3.3.1.1 Modélisation et évaluation locales des séquences d'événements

La première étape de l'approche compositionnelle que nous proposons consiste à considérer séparément chaque mode du système, le modéliser et analyser les séquences d'événements qui peuvent être exécutées au sein de ce mode.

Conformément à l'approche présentée au chapitre présentée et synthétisée sur la figure (2.1), l'identification et la quantification des séquences d'événements d'un système débute par la construction d'un automate à états finis (AEF) qui représente la chaîne de Markov ou le processus semi-markovien associé à chaque mode du système.

L'identification des séquences d'événements repose alors sur la détermination, à l'aide du lemme d'Arden, des expressions régulières correspondant aux sous-langages associés à chacun des états de l'automate fini. A partir de ces sous-langages, il est possible d'extraire une ou plusieurs séquences d'événements selon les objectifs de l'étude.

Après avoir identifié les séquences d'événements, il est alors possible de procéder à l'évaluation de leur probabilité d'occurrence. En régime asymptotique, le calcul repose sur un p-automate obtenu à partir de la chaîne de Markov à temps discret immergée (construite en considérant, dans le processus stochastique continu, les instants de saut à la fin du temps de séjour dans l'état courant) et de la distribution stationnaire des probabilités d'état. Nous rappelons que la distribution stationnaire de probabilité correspondant à la CdM immergée est unique dans la mesure où la chaîne de Markov est ergodique et représentée par un graphe fortement connexe. En conséquence la probabilité p_{ij} d'un événement e_{ij} est donnée par l'équation (1.14). La probabilité d'occurrence d'une séquence d'événements est alors obtenue en appliquant l'équation (2.11) ; la somme des probabilités des séquences appartenant à un sous-langage (eq. 2.12) peut également être calculée de manière symbolique, ce qui s'avère à très utile dans les études de SdF pour borner la probabilité d'atteindre un état de panne/dangereux. En régime transitoire, la technique basée sur le calcul des transformées de Laplace des temps de premier passage depuis un état initial vers un état cible est utilisée pour obtenir la probabilité des séquences d'événements.

Pour résumer, l'identification et la quantification **locales** (c'est à dire pour un mode unique de défaillance ou de fonctionnement, modélisé et analysé de manière isolée) des séquences d'événements suit une approche en tout point identique à celle développée au chapitre précédent.

3.3.1.2 Modélisation et évaluation globales des séquences d'événements

Selon la première partie de notre approche, c'est atteignable de faire une évaluation probabiliste des séquences d'événements qui décrivent le comportement de chaque mode d'un système, étant donné que le nombre d'états et des transitions de l'automate qui représente un seul mode ne connais pas une explosion combinatoire. Sachant que dans la vie réelle, nous travaillons dans la plupart des situations avec des systèmes complexes (de grande taille) qui peuvent posséder de multiples modes de comportement.

Considérant comme acquis que l'identification et la quantification de séquences d'événements peuvent être réalisées sur des modèles de taille réduite, représentant chacun un mode particulier de fonctionnement ou de défaillance, la question posée est la suivante : comment est-il possible d'intégrer les résultats obtenus séparément pour chacun des modes afin d'obtenir une évaluation globale des séquences d'événements, c'est à dire en évitant d'identifier et de quantifier ces séquences sur un modèle global du système ?

Les opérateurs de choix et de concaténation définis dans le cadre de la théorie des langages probabilistes apportent une réponse pertinente à cette question :

- *l'opérateur de choix* sera mis en œuvre pour traiter des modes variés de défaillance ou de fonctionnement en compétition (nous proposerons dans la suite de chapitre un exemple d'application relatif à un mode de défaillance indépendante et un mode de défaillance de cause commune) ;
- *l'opérateur de concaténation* sera mis en œuvre pour traiter les commutations entre plusieurs phases ou plusieurs modes de défaillance ou de fonctionnement successivement

actifs (nous proposerons dans la suite de chapitre un exemple d'application relatif à la commutation entre des politiques de maintenance parfaite et imparfaite induisant des comportements nominaux/dégradés du système).

L'utilisation des opérateurs de choix et de concaténation n'est pas limité au cas des systèmes caractérisés par plusieurs modes de défaillance ou par de multiples politiques de maintenance. L'approche modulaire basée sur ces deux opérateurs peut s'appliquer dans le contexte plus général de tous les systèmes multi-modes et multi-phases.

3.3.2 Composition modulaire par l'opérateur de choix

L'opérateur de choix permet de réaliser un choix non déterministe entre deux comportements ou deux modes de défaillance, chaque mode ayant une influence relative sur la défaillance globale (dans notre cas les défaillances indépendantes et les défaillances de cause commune (DCC) entrant dans le calcul d'une probabilité globale de défaillance).

Définition 7. *[Garg* et al.*, 1999] : Soit deux p-langages $\mathbb{L}_1$, $\mathbb{L}_2$ et $p \in [0,1]$, l'opérateur de choix, noté $\mathbb{L}_1 +_p \mathbb{L}_2$, est défini par :*

$$\mathbb{P}(\langle s\rangle^{\mathbb{L}_1 +_p \mathbb{L}_2}) = p \cdot \mathbb{P}\left(\langle s\rangle^{\mathbb{L}_1}\right) + (1-p) \cdot \mathbb{P}\left(\langle s\rangle^{\mathbb{L}_2}\right). \tag{3.1}$$

Autrement dit, le système qui intègre les deux modes se comporte soit selon le premier mode du système représenté par un automate noté avec A_1 (qui reconnaît le p-langage $\mathbb{L}_1$) avec une probabilité p, soit comme le deuxième mode du système représenté par un automate noté avec A_2 (qui reconnaît le p-langage $\mathbb{L}_2$) avec une probabilité $(1-p)$. L'opérateur de choix peut être généralisé pour plusieurs p-langages, dans ce cas il sera exprimé comme une combinaison convexe de ses arguments (les p-langages). La définition généralisée sera présentée dans la dernière section de ce chapitre. Le théorème ci-dessous exprime quelques propriétés de l'opérateur de choix.

Théorème 3. *[Garg* et al.*, 1999] Soit deux p-langages $\mathbb{L}_1$, $\mathbb{L}_2$ et $p \in [0,1]$.*

1. *$\mathbb{L}_1 +_p \mathbb{L}_2$ est un p-langage.*
2. *$\Delta(\mathbb{L}_1 +_p \mathbb{L}_2) = p\Delta(\mathbb{L}_1) + (1-p)\Delta(\mathbb{L}_2)$.*
3. *Le choix est un opérateur continu sur ses deux arguments.*

Démonstration

1. Le fait que $\mathbb{L}_1 +_p \mathbb{L}_2$ est un p-langage est évident, étant donné que c'est la combinaison convexe de deux p-langages.
2. Pour simplifier les notations, on définit $\Delta(\mathbb{P})(s)$ et $\Lambda(\mathbb{P})(s)$ comme respectivement la probabilité de terminaison d'une séquence s et la probabilité que le système continue à évoluer au delà de se :

$$\forall s \in \Sigma^* : \Delta(\mathbb{P})(s) = \mathbb{P}(se_\Delta)\,; \Lambda(\mathbb{P})(s) = \sum_{e\in\Sigma} \mathbb{P}(se)$$

Ainsi, la deuxième propriété résulte de la série d'égalités suivante :

$$\begin{aligned}
\Delta(\mathbb{L}_1 +_p \mathbb{L}_2) &= \mathbb{L}_1 +_p \mathbb{L}_2 - \Lambda(\mathbb{L}_1 +_p \mathbb{L}_2) \\
&= (p\mathbb{L}_1 + (1-p)\mathbb{L}_2) - \Lambda(p\mathbb{L}_1 + (1-p)\mathbb{L}_2) \\
&= (p\mathbb{L}_1 + (1-p)\mathbb{L}_2) - [p\Lambda(\mathbb{L}_1) + (1-p)\Lambda(\mathbb{L}_2)] \\
&= [p\mathbb{L}_1 - p\Lambda(\mathbb{L}_1)] + [(1-p)\mathbb{L}_2 - (1-p)\Lambda(\mathbb{L}_2)] \\
&= p\Delta(\mathbb{L}_1) + (1-p)\Delta(\mathbb{L}_2)
\end{aligned}$$

3. Par la symétrie de la définition de l'opérateur de choix dans ses arguments, il suffit de montrer que pour toute la chaîne $\{\mathbb{L}_i\}$ de p-langages et $p \in [0,1]$:

$$\mathbb{K} +_p (\sqcup_i \mathbb{L}_i) = \sqcup_i (\mathbb{K} +_p \mathbb{L}_i),$$

où $\mathbb{K}$ est aussi un p-langage.

Considérons deux modes de fonctionnement ou de défaillance différents. Chaque mode est représenté par un p-automate (noté avec A_1 et A_2) qui correspond à un processus markovien ou semi-markovien. Par hypothèse, les automates décrivant ces modes comportent un nombre réduit d'états et des transitions ; l'approche proposée au chapitre précédent peut donc leur être appliquée, indépendamment, sans difficulté.

L'application de la définition 7 permet de composer les sous-langages associés aux états des automates A_1 et A_2 par l'opérateur de choix. Cet opérateur permet d'obtenir probabilités des séquences qui appartiennent aux sous-langages associés aux états du système global intégrant les deux modes de défaillance (sans construire explicitement le modèle global par composition des automates A_1 et A_2) :

$$\mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_1 +_p \mathbb{L}_2}\right) = p \cdot \mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_1}\right) + (1-p) \cdot \mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_2}\right) \tag{3.2}$$

où $\mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_1}\right)$ représente la somme de probabilités des séquences appartenant au sous-langage $L_{(x_i)}$ associé à l'état x_i dans l'automate A_1 qui reconnaît le p-langage $\mathbb{L}_1$.

Ainsi, pour n'importe quel état du système global (quelque soit son mode), nous sommes en mesure de déterminer la somme des probabilités des séquences appartenant aux sous-langages associés. Cette somme sert par la suite dans les calculs des indicateurs de Sûreté de Fonctionnement (fiabilité, disponibilité, maintenance).

3.3.3 Composition modulaire par l'opérateur de concaténation

L'opérateur de concaténation permet la modélisation du fonctionnement séquentiel de deux phases ou deux modes de fonctionnement d'un même système. Il est ainsi parfaitement adapté pour représenter l'enchaînement de comportements nominaux et dégradés.

Définition 8. *[Garg* et al.*, 1999] : Soit deux p-langages $\mathbb{L}_1$, $\mathbb{L}_2$ et $p \in [0,1]$, l'opérateur de concaténation, noté $\mathbb{L}_1 \cdot_p \mathbb{L}_2$, est défini par :*

$$\mathbb{P}(\langle s \rangle^{\mathbb{L}_1 \cdot_p \mathbb{L}_2}) = \mathbb{P}(\langle s \rangle^{\mathbb{L}_1}) + p \sum\nolimits_{t<s} \mathbb{P}(\langle t e_\Delta \rangle^{\mathbb{L}_1}) \mathbb{P}(\langle t^{-1} s \rangle^{\mathbb{L}_2}), \forall \langle s \rangle \in \Sigma^* \tag{3.3}$$

Cela signifie que le modèle global du système considéré exécute soit une séquence $\langle s \rangle$ appartenant entièrement au p-automate A_1 qui reconnaît le p-langage $\mathbb{L}_1$ soit, avec une probabilité p, exécute un préfixe t de s dans A_1 et puis le reste de la séquence $t^{-1}s$ dans le p-automate A_2 qui reconnaît le p-langage $\mathbb{L}_2$.

La définition 8 proposée par [Garg *et al.*, 1999] fait l'hypothèse que les alphabets des automates A_1 et A_2 (Σ) sont identiques et que les p-langages $\mathbb{L}_1$ et $\mathbb{L}_2$ sont des mesures de probabilités connues. Dans notre cas, ces deux alphabets (Σ_1 et Σ_2) pouvant contenir des événements privés (appartenant à l'un mais pas à l'autre), la définition 8 est complétée par :

- l'identification d'événements particuliers de commutation appartenant à $\Sigma_c \subset \Sigma$; sur occurrence d'un événement appartenant à Σ_c, une commutation d'un mode vers l'autre peut être réalisée avec une probabilité p,

- le p-langage $\mathbb{L}_2$ (mesure de probabilité) doit être redéfini à partir de l'état de l'automate A_2 activé par la commutation ; notons que la séquence $t^{-1}s$ n'est pas nécessairement incluse dans L_2 mais doit être contenue dans au moins un mot de L_2.
- selon l'appartenance de l'événement de commutation $e_c \in \Sigma_c$ aux alphabets de A_1 et A_2, deux situations sont possibles :
  - si l'événement de commutation appartient aux deux alphabets, alors le préfixe te_c sera exécuté dans A_1 puis le reste de la séquence $t^{-1}e_c s$ dans A_2 ; cette situation peut être comparée à une synchronisation sur événement partagé.
  - si l'événement de commutation n'appartient qu'à l'alphabet de A_1, le préfixe exécuté dans A_1 sera te_c et le reste de la séquence dans A_2 sera $t^{-1}s$; ce cas nécessite la représentation explicite de l'état de A_2 activé par la commutation (la commutation sera représentée par une flèche pointillée dans les modèles présentés au chapitre 4).

Le théorème ci-dessous exprime quelques propriétés de l'opérateur de concaténation. Sa démonstration est présentée dans [Garg *et al.*, 1999].

Théorème 4. *[Garg* et al.*, 1999] Soit deux p-langages $\mathbb{L}_1$, $\mathbb{L}_2$ et $p \in [0,1]$.*

1. *$\mathbb{L}_1 \cdot_p \mathbb{L}_2$ est un p-langage*
2. *$\Delta(\mathbb{L}_1 \cdot_p \mathbb{L}_2) = (1-p)\,\Delta(\mathbb{L}_1) + p\Delta(\mathbb{L}_1) \circ \Delta(\mathbb{L}_2)$*
3. *La concaténation est continue dans son deuxième argument, n'étant même pas monotone dans son premier argument.*

où $f \circ g$ représente la convolution de deux fonctions à valeurs réelles $f, g : \Sigma^* \to \mathcal{R}$, définie par :

$$f \circ g = \sum\nolimits_t f(t)\, g\left(t^{-1}s\right).$$

L'application de la définition 8 permet de déterminer les probabilités de différentes séquences d'événements caractérisant le comportement global du système dont les commutations entre modes :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot_p \mathbb{L}_2}\right) = \begin{cases} \mathbb{P}\left(s_i^{\mathbb{L}_1}\right) & \text{si } s_i \in A_1 \\ p \sum_{\substack{t < s_{i-1}e_{(i-1)i} \\ e_{(i-1)i} \in \Sigma_c}} \mathbb{P}\left((te_\Delta)^{\mathbb{L}_1}\right) \mathbb{P}\left(\left(t^{-1}s_i\right)^{\mathbb{L}_2}\right) & \text{sinon} \end{cases} \tag{3.4}$$

Si la séquence s_i appartient entièrement à l'automate A_1 qui correspond au premier mode du système, $\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot_p \mathbb{L}_2}\right)$ est donnée par la probabilité de l'occurrence de la séquence s_i dans l'automate A_1. En d'autres termes, cette situation correspond à des séquences où le système évolue uniquement dans son premier mode sans commutation sur le second. Dans les autres cas, c'est à dire pour les séquences pour lesquelles le système a évolué dans son premier mode avant de commuter sur le second, avec une probabilité p, un préfixe te_Δ de la séquence s_i correspondra à un comportement de l'automate A_1 (en spécifiant que le dernier événement du préfixe est toujours un événement de réparation appartenant à Σ_c) alors que le reste de la séquence $(t^{-1}s_i)$ correspondra à un comportement de l'automate A_2. Bien entendu, si la commutation entre les deux modes est déterministe, la probabilité p sera égale à 1 et, par conséquent, l'état source de la commutation dans le premier automate sera absorbant.

De cette manière, nous pouvons intégrer, dans l'évaluation des séquences, deux types de comportements différents que le système peut être amené à suivre après occurrence d'événements de commutation. L'objectif est d'évaluer les séquences du système global qui intègre les deux

modes. L'opérateur de concaténation permet d'atteindre cet objectif sans construire explicitement le modèle global (par composition synchrone des automates par exemple).

L'opérateur de concaténation peut être généralisé, comme l'opérateur de choix, pour plusieurs p-langages. La définition généralisée sera présentée dans la dernière section de ce chapitre.

3.4 Exemple d'application de l'opérateur de choix

Pour illustrer l'utilisation de l'opérateur de choix, considérons le système de régulation de température d'un four, présenté au chapitre précédent, et deux modes de défaillance : un mode de défaillance indépendante et un mode de défaillances de cause commune (DCC). Chaque mode ayant une influence relative sur la défaillance globale du four, l'opérateur de choix permettra de réaliser un choix non déterministe entre ces deux modes.

Dans une première étape, chaque mode de défaillance sera étudié séparément en identifiant et quantifiant, en régime asymptotique, les séquences d'événements admissibles pour chaque mode. Dans un deuxième temps, l'opérateur de choix sera appliqué afin de déterminer la probabilité d'occurrence des séquences associées au comportement global du système intégrant les deux modes de défaillance.

3.4.1 Mode de défaillance indépendante

Le premier mode de défaillance représenté par l'automate à états finis A_1 dans la figure (3.1) correspond à un processus semi-markovien du fait de la présence de transitions exponentielles et déterministes. Les transitions exponentielles caractérisent les événements de défaillance (λ_{PI} et λ_{TOR}) et de réparation (μ_{PI}, μ_{TOR}). Les transitions déterministes sont $diag\_temps1$ et $diag\_temps2$ entre les états 2 et 3 (dégradé) et entre les états 4 et 5 (défaillant). Elle correspondent à la durée constante nécessaire pour que les défaillances indépendantes soient détectées. Ce retard de détection représente le temps nécessaire pour que la température augmente à partir de sa valeur de référence jusqu'à un seuil critique mesuré par un capteur du sous-système de diagnostic. Les seuil minimal et maximal de la température ont été fixé sur ce cas d'étude à 150℃ et 240℃ et les valeurs des constantes de durée ont été définies à partir de l'observation de l'évolution des paramètres physiques ; elles sont donc supposées connues.

Connaissant l'automate à états finis qui décrit ce premier mode de défaillance, les expressions régulières des sous-langages L_i associés aux états du système peuvent être obtenues par résolution du système d'équation ci-dessous à l'aide du lemme d'Arden :

$$\begin{cases} L_1 = L_3 e_{31} + L_6 e_{61} \\ L_2 = L_1 e_{12} \\ L_3 = L_2 e_{23} + L_5 e_{53} \\ L_4 = L_3 e_{34} + L_6 e_{64} \\ L_5 = L_4 e_{45} \\ L_6 = L_5 e_{56} \end{cases} \tag{3.5}$$

Nous ne présentons ici que les expressions régulières des sous-langages L_1 et L_5 associés respectivement à l'état nominal (où les deux contrôleurs sont opérationnels) et à l'état de panne (où les deux contrôleurs sont défaillants) :

$$L_1 = \left[e_{12}e_{23}e_{31} + e_{12}e_{23}e_{34}\left(e_{45}e_{53}e_{34} + e_{45}e_{56}e_{64}\right)^*\left(e_{45}e_{53}e_{31} + e_{45}e_{56}e_{61}\right)\right]^* \tag{3.6}$$

$$L_5 = L_1 e_{12}e_{23}e_{34}\left(e_{45}e_{53}e_{34} + e_{45}e_{56}e_{64}\right)^* e_{45} \tag{3.7}$$

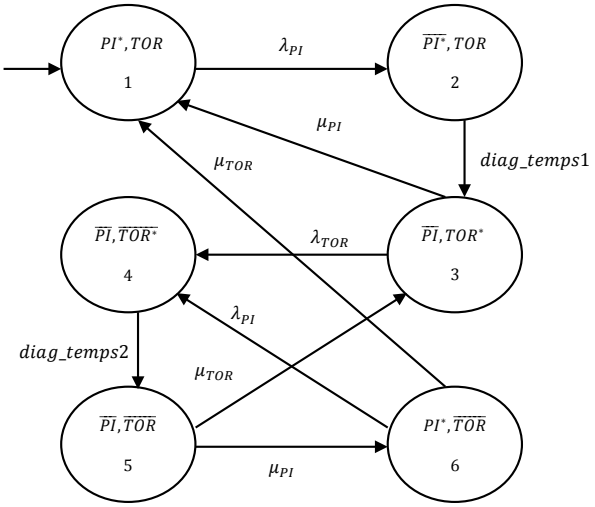


FIGURE 3.1 – Défaillances indépendantes (semi-markovien)

En appliquant la démarche proposée au chapitre 2, il est possible d'obtenir la probabilité d'une ou plusieurs séquences d'événements (eq. 2.11), extraites de l'expression des sous-langages L_1 à L_6. Toujours selon cette démarche, il est possible de déterminer, de manière analytique et numérique, la somme ($\mathbb{Q}(L_i)$) des probabilités des séquences appartenant à chaque sous-langage L_i. Compte tenu de la complexité des expressions régulières (obtenues par calcul symbolique), nous ne présentons ici que les valeurs numériques pour les sous-langages L_1 et L_5 :

$$\mathbb{Q}(L_1) = \mathbb{Q}\left(L_{(PI^*,TOR)}\right) = 1$$

$$\mathbb{Q}(L_5) = \mathbb{Q}\left(L_{\left(\overline{PI},\overline{TOR}\right)}\right) = 2.3751 \cdot 10^{-4}$$

Le résultat obtenu pour $\mathbb{Q}(L_1)$ est trivial puisqu'il représente la somme des probabilités des séquences d'événements qui, depuis l'état initial, ramènent le système dans cet état initial et que le processus stochastique, correspondant à ce premier mode de défaillance du système, est ergodique. Les autres résultats numériques n'ont pas d'interprétation particulière.

3.4.2 Défaillances de cause commune (DCC)

Le deuxième mode de défaillance, représenté par l'automate à états finis A_2 de la figure (3.2), est caractérisé par la présence d'une défaillance de cause commune (DCC). Cette DCC entraîne la défaillance simultanée des deux contrôleurs et conduit directement à l'état de panne 2 de l'automate où la température du four n'est plus contrôlée. On suppose que la DCC est détectée instantanément et qu'elle est décrite par le modèle du facteur β [Cai *et al.*, 2012]. Rappelons que selon ce modèle, la valeur du β est donnée par le rapport entre le taux de défaillance de cause commune λ_{DCC} et la somme entre le taux de défaillance indépendante λ_{ind} et le taux de défaillance de cause commune λ_{DCC} (λ_{tot}) :

$$\beta = \frac{\lambda_{DCC}}{\lambda_{tot}} = \frac{\lambda_{DCC}}{\lambda_{ind} + \lambda_{DCC}} \tag{3.8}$$

Il s'en suit que le taux de défaillance de cause commune peut être exprimée sous la forme suivante :

$$\lambda_{DCC} = \beta\lambda_{tot}, \tag{3.9}$$

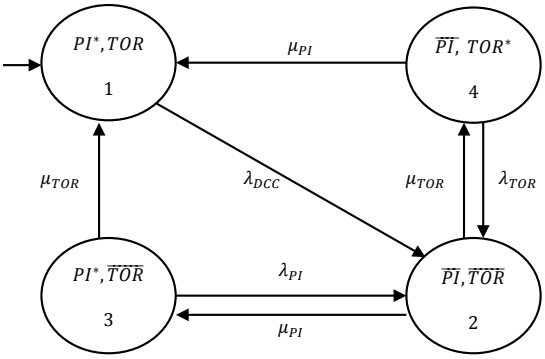


FIGURE 3.2 – Défaillances de cause commune (markovien)

et que le taux de défaillance indépendante est donné par l'équation :

$$\lambda_{ind} = (1 - \beta)\,\lambda_{tot} \tag{3.10}$$

Pour le mode de défaillance de cause commune de notre cas d'étude, le taux d'occurrence du DCC, λ_{DCC}, est obtenu en utilisant la relation suivante :

$$\lambda_{DCC} = \max\left(\beta\lambda_{PI}, \beta\lambda_{TOR}\right) \tag{3.11}$$

en considérant que $\beta = 0.05$.

Concernant les autres taux de défaillances (excepté λ_{DCC}), ils correspondent aux taux de défaillance indépendante des composants (ils ne correspondent bien sûr pas à un taux de défaillance total des composants incluant les défaillances indépendantes et les DCC puisque ces dernières sont traitées explicitement dans ce modèle).

De la même manière que pour le mode précédent, l'application de l'approche proposée au chapitre 2 permet de déterminer les expressions régulières des quatre sous-langages correspondant aux quatre états de l'automate. Nous ne présentons ci-dessous que les expressions régulières des sous-langages L_1 et L_2 associés respectivement à l'état nominal (où les deux régulateurs sont opérationnels) et à l'état de panne (où les deux régulateurs sont défaillants) :

$$L_1 = \left[e_{12}\left(e_{23}e_{32} + e_{24}e_{42}\right)^*\left(e_{23}e_{31} + e_{24}e_{41}\right)\right]^* \tag{3.12}$$

$$L_2 = L_1 e_{12}\left(e_{23}e_{32} + e_{24}e_{42}\right)^* \tag{3.13}$$

Toujours en appliquant l'approche proposée au chapitre 2, il est possible d'obtenir la probabilité d'une ou plusieurs séquences d'événements (eq. 2.11) extraites des expressions des sous-langages déterminés, ainsi que, de manière analytique et numérique, la somme ($\mathbb{Q}(L_i)$) des probabilités de toutes les séquences appartenant à un sous-langage L_i. Nous présentons ici que les valeurs numériques correspondant à la somme des probabilités des séquences appartenant aux sous-langages L_1 et L_2 :

$$\mathbb{Q}(L_1) = \mathbb{Q}\left(L_{(PI^*,TOR)}\right) = 1$$

$$\mathbb{Q}(L_2) = \mathbb{Q}\left(L_{(\overline{PI},\overline{TOR})}\right) = 1$$

Pour le $\mathbb{Q}(L_1)$, le résultat est trivial pour les mêmes raisons que pour le mode précédent. En ce qui concerne $\mathbb{Q}(L_2)$, le résultat s'explique par le fait que $\mathbb{Q}(L_1) = 1$ et que $\mathbb{P}(e_{12}) = 1$ (λ_{DCC} associé à la seule transition sortante de l'état 1).

3.4.3 Composition modulaire des modes de défaillance

Au cours de l'étape précédente, nous avons modélisé et évalué individuellement deux modes de défaillance d'un même système. L'objectif est maintenant de réaliser l'évaluation des probabilités des séquences possibles dans un modèle global qui intègre les deux modes de défaillance. Autrement, à partir de la connaissance des probabilités d'occurrence des séquences conduisant le système dans un état dangereux, qu'elles correspondent à un comportement décrit dans le cadre du mode de défaillance indépendante ou bien dans le cadre du mode de défaillance de cause commune.

En général, pour atteindre cet objectif, l'analyste dispose d'un modèle global du système qu'il construit soit de manière directe et monolithique (avec toutes les limites connues dans le cas de systèmes complexes en termes d'exactitude et d'exhaustivité) soit qu'il obtient à partir de modèles élémentaires par exemple par composition synchrone sur les automates à états finis [Cassandras et Lafortune, 2008]. Dans les deux cas, le modèle global résultant et intégrant les deux modes de défaillances est de taille importante et nous avons souligné en début de chapitre les limites de ces pratiques dans le cadre de notre approche. L'intérêt des opérateurs de composition est qu'ils évitent la construction du modèle global du système puisqu'ils s'appliquent aux langages décrits indépendamment dans chacun des deux modes.

L'opérateur de choix permet donc de réaliser l'évaluation des probabilités des séquences d'événements intégrant les deux modes de défaillance. La probabilité p, mentionnée dans la définition de l'opérateur de choix (eq. 3.1), est considérée comme égale au facteur β (du mode DCC) puisque que le facteur β est vu comme le rapport entre le taux de défaillance de cause commune λ_{DCC} et le taux de défaillance total λ_{tot}. En appliquant la définition de l'opérateur de choix (*Définition 7*), la probabilité d'occurrence d'une séquence associée à un comportement qui intègre les deux modes de défaillance, est donnée par l'équation suivante :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 +_p \mathbb{L}_2}\right) = (1-\beta)\cdot\mathbb{P}\left(s_i^{\mathbb{L}_1}\right) + \beta\cdot\mathbb{P}\left(s_i^{\mathbb{L}_2}\right) \tag{3.14}$$

De plus, il est également possible d'obtenir la somme des probabilités des séquences d'événements qui conduisent le système global vers un état fonctionnel ou vers un état indésirable ou dangereux. Ne disposant pas de modèle global avec une représentation explicite de ces états globaux, ce calcul reposera sur les probabilités associées aux sous-langages correspondant aux états de fonctionnement ou de panne dans les modèles locaux de chacun des deux modes de défaillance.

Autrement dit, la somme de probabilités de séquences qui amènent le système dans son état critique, où la température est hors contrôle, est donnée par la somme entre, d'une part la somme des probabilités des séquences appartenant au sous-langage associé à l'état de panne du premier mode de défaillance, pondérée par un coefficient $(1-\beta)$, et, d'autre part, la somme des probabilités des séquences appartenant au sous-langage associé à l'état de panne du deuxième mode de défaillance, pondérée par un coefficient β :

$$\mathbb{Q}\left(L_{(\overline{PI},\overline{TOR})}\right) = (1-\beta)\cdot\mathbb{Q}\left(L^{A_1}_{(\overline{PI},\overline{TOR})}\right) + \beta\cdot\mathbb{Q}\left(L^{A_2}_{(\overline{PI},\overline{TOR})}\right) = 0.0502$$

En ce qui concerne l'état nominal, le raisonnement est identique. La somme des probabilités de séquences appartenant au sous-langage associé à l'état nominal (dans le modèle global) est donnée par la somme entre, d'une part, la somme des probabilités des séquences appartenant au sous-langage associé à l'état nominal du premier mode de défaillance, pondérée par un coefficient $(1-\beta)$, et, d'autre part, la somme des probabilités des séquences appartenant au sous-langage associé à l'état nominal du deuxième mode de défaillance, pondérée par le coefficient β :

$$\mathbb{Q}\left(L_{(PI,TOR)}\right) = (1-\beta)\cdot\mathbb{Q}\left(L^{A_1}_{(PI,TOR)}\right) + \beta\cdot\mathbb{Q}\left(L^{A_2}_{(PI,TOR)}\right) = 1$$

Le même raisonnement peut s'appliquer pour les modes dégradés (où un seul des deux régulateurs est défaillant). Les deux équations suivantes donnent la somme des probabilités de séquences appartenant au sous-langages $L_{(\overline{PI},TOR)}$ et $L_{(PI,\overline{TOR})}$ associés aux états dégradés $\overline{PI},TOR$ et $PI,\overline{TOR}$.

$$\mathbb{Q}\left(L_{(\overline{PI},TOR)}\right) = (1-\beta)\cdot\mathbb{Q}\left(L^{A_1}_{(\overline{PI},TOR)}\right) + \beta\cdot\mathbb{Q}\left(L^{A_2}_{(\overline{PI},TOR)}\right) = 0.9779$$

$$\mathbb{Q}\left(L_{(PI,\overline{TOR})}\right) = (1-\beta)\cdot\mathbb{Q}\left(L^{A_1}_{(PI,\overline{TOR})}\right) + \beta\cdot\mathbb{Q}\left(L^{A_2}_{(PI,\overline{TOR})}\right) = 0.0223$$

De la même manière que pour chacun des modes de défaillance pris isolément, on peut remarquer que, comme attendu, la somme des probabilités des séquences amenant le système global dans un état de bon fonctionnement est $\mathbb{Q}\left(L_{(PI,TOR)}\right) = 1$. Les valeurs obtenues pour les autres sous-langages n'ont pas de signification particulière.

3.4.4 Validation analytique par la technique de Chaîne de Markov immergée

Afin de valider notre approche, nous proposons de construire l'automate global du système qui intègre les deux modes de défaillance : défaillances indépendantes et défaillances de cause commune. L'automate est construit en utilisant l'opérateur formel de *composition parallèle* sur les deux automates représentant les deux modes du système. Sa taille est de 23 états. Pour cet automate, on détermine la chaîne de Markov immergée et la distribution stationnaire des probabilités π.

Les probabilités des états x_i (nominal, dégradé où défaillant) sont obtenues, à partir de l'automate global, en effectuant la somme des probabilités stationnaires de tous les états correspondant aux couples $\left(\left(x_i^{A_1}\right),*\right)$ ou $\left(*,\left(x_i^{A_2}\right)\right)$ résultant du produit cartésien des états réalisés par la composition parallèle (autrement dit l'un ou l'autre des états du couple doit correspondre à l'état pour lequel on cherche à déterminer la probabilité) :

$$\pi\left(x_i\right) = \sum\nolimits_{x_i\in A_1}\pi\left(\left(x_i^{A_1}\right),*\right) + \sum\nolimits_{x_i\in A_2}\pi\left(*,\left(x_i^{A_2}\right)\right) \tag{3.15}$$

Nous pouvons alors vérifier que la probabilité stationnaire de chaque état x_i est égale à la probabilité stationnaire de l'état initial x_0, multipliée par la somme des probabilités des séquences appartenant au sous-langage $L_{(x_i)}$ associé à l'état considéré x_i (dernière colonne dans le tableau 3.1) :

$$\pi\left(x_i\right) = \pi\left(x_0\right)\cdot\mathbb{Q}\left(L_{(x_i)}\right), \forall x_i, i = 1..4 \tag{3.16}$$

La comparaison des résultats obtenus par calcul des probabilités stationnaires sur le modèle global et ceux obtenus à l'aide de l'équation (3.16) exploitant nos résultats concernant $\mathbb{Q}\left(L_{(x_i)}\right)$ est présentée dans le tableau (3.1) pour quatre états caractéristiques (fonctionnement normal x_1, panne x_2, fonctionnement dégradé avec un des deux régulateurs en panne x_3 et x_4) :

- la dernière colonne indique les valeurs des probabilités d'état stationnaires obtenues sur le modèle global à l'aide de l'équation (3.15) ;

TABLE 3.1 – Comparaison des probabilités d'état stationnaires $\pi(x_i)$ obtenues à partir des résultats fournis par notre approche et par calcul sur le modèle global

Etats	**APPROCHE PROPOSEE** $\mathbb{Q}(L_{(x_i)})$			$\pi(x_i)$	**MODELE GLOBAL** $\pi(x_i)$
	Mode 1 isolé	**Mode 2 isolé**	**Mode 1 et 2**		
$x_1 = (PI^*, TOR)$	1	1	1	0.4882	0.4882
$x_2 = (\overline{PI}, \overline{TOR})$	$2.3751 \cdot 10^{-4}$	1	0.0502	0.0245	0.0245
$x_3 = (\overline{PI}, TOR^*)$	1	0.5557	0.9779	0.4774	0.4774
$x_4 = (PI^*, \overline{TOR})$	$1.0556 \cdot 10^{-4}$	0.4446	0.0223	0.0108	0.0109

- pour l'approche proposée, les colonnes 2, 3 et 4 rappellent les résultats obtenus pour la somme des probabilités des séquences associées aux sous-langages, respectivement dans le mode de défaillance indépendante (mode 1), le mode de défaillance de cause commune (mode 2) et avec l'opérateur de choix (mode 1 et 2) ; la colonne 4 indique la valeur des probabilités d'état stationnaires obtenues en réutilisant les résultats de notre approche à l'aide de l'équation (3.16) avec une valeur de $\pi(x_0) = 0.4882$

La cohérence des résultats obtenus par les deux approches contribue donc à valider notre proposition relative à l'opérateur de choix.

3.5 Exemple d'application de l'opérateur de concaténation

Pour illustrer l'application de l'opérateur de concaténation, nous avons choisi un système possédant deux modes de maintenance distincts :

- un mode de maintenance parfaite qui, suite à une défaillance puis une réparation d'un composant, a pour effet de le remettre dans un état aussi bon que neuf où ses paramètres de fiabilité seront identiques à ceux d'origine,
- un mode de maintenance imparfaite qui, suite à une défaillance puis une réparation d'un composant, ne remet pas le composant dans un état neuf, c'est à dire que ses paramètres fiabilistes ont pu se dégradés (ce type de maintenance augmente souvent par exemple leur probabilité de refus au démarrage).

Le comportement d'un composant après réparation est donc différent selon le mode de maintenance. Nous allons considérer qu'à l'initialisation du système, les composants sont dans un état neuf. Leur comportement sera donc celui associé à un mode de maintenance parfaite. Suite à une réparation, le composant peut continuer à suivre un comportement correspondant à une politique de maintenance parfaite si la réparation s'est correctement déroulée, avec une probabilité $1 - p$, ou bien commuter vers un comportement correspondant à une politique de maintenance imparfaite avec une probabilité p. En d'autres termes, les deux modes de fonctionnement sont "activés" séquentiellement avec une probabilité p de faire la commutation après une réparation. Le comportement correspondant à la politique de maintenance imparfaite peut être assimilé à un comportement en mode dégradé.

Comme pour l'opérateur de choix, nous allons d'abord modéliser séparément ces deux politiques de maintenance et procéder à l'identification et la quantification des séquences d'événements dans chacun de ces modes. Sur la base de ces résultats, l'opérateur de concaténation permettra d'évaluer des séquences caractérisant le comportement global du système et de la commutation entre ces deux politiques de maintenance.

3.5.1 Politique de maintenance parfaite

Le premier comportement du système, représenté par l'automate A_1 de la figure (3.3), correspond à la politique de réparation parfaite. On part de l'hypothèse que lorsqu'un régulateur de température du four tombe en panne, il est réparé et rendu dans un état aussi bon neuf. En d'autres termes, les régulateurs sont immédiatement opérationnels après réparation pour exercer leur mission sans risque de refus à la sollicitation. On ne considère donc dans ce modèle que deux types d'événements : la défaillance et la réparation des régulateurs. Les taux qui caractérisent ces événements ont les valeurs suivantes : $\lambda_{PI} = 3.5{\cdot}10^{-5}h^{-1}$, $\lambda_{TOR} = 2{\cdot}10^{-5}h^{-1}$, $\mu_{PI} = 8{\cdot}10^{-2}h^{-1}$, $\mu_{TOR} = 10^{-1}h^{-1}$.

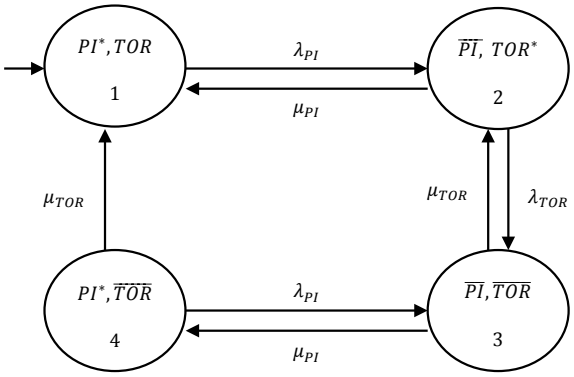


FIGURE 3.3 – Modélisation de la politique de maintenance parfaite (A_1)

A partir de ce modèle, l'objectif est d'identifier les sous-langages L_1, L_2, L_3, L_4 qui comprennent l'ensemble de toutes les séquences d'événements conduisant à ces quatre états, de calculer la probabilité d'occurrence de quelques séquences d'intérêt extraites de ces sous-langages ou bien de calculer la somme des probabilités des séquences appartenant à un sous-langage L_i, $\mathbb{Q}(L_i)$. L'approche utilisée est celle présentée dans le deuxième chapitre de cette thèse.

La résolution du système d'équations, sur les sous-langages (L_1, L_2, L_3 et L_4), obtenu à partir de l'automate A_1, et donné ci-dessous :

$$\begin{cases} L_1 = L_2e_{21} + L_4e_{41} \\ L_2 = L_1e_{12} + L_3e_{32} \\ L_3 = L_2e_{23} + L_4e_{43} \\ L_4 = L_3e_{34} \end{cases} \tag{3.17}$$

permet d'obtenir l'expression régulière qui détermine formellement chacun d'entre eux. A titre d'exemple, les sous-langages L_1 et L_3, qui conduisent le système respectivement dans son état nominal et dangereux (où la température n'est plus contrôlée car les deux régulateurs sont défaillants), sont les suivants :

$$L_1 = [e_{12}(e_{23}(e_{34}e_{43})^*e_{32})^*(e_{21} + e_{23}(e_{34}e_{43})^*e_{34}e_{41})]^* \tag{3.18}$$

$$L_3 = [e_{12}(e_{23}(e_{34}e_{43})^*e_{32})^*(e_{21} + e_{23}(e_{34}e_{43})^*e_{34}e_{41})]^*e_{12}[e_{23}(e_{34}e_{43})^*e_{32}]^*e_{23}(e_{34}e_{43})^* \tag{3.19}$$

Les probabilités des événements p_{ij} étant calculées à l'aide de la chaîne de Markov immergée (équation 1.14), les probabilités d'occurrence des séquences (mots d'un sous-langage) en régime asymptotique ainsi que la somme des probabilités des séquences appartenant à un sous-langage

TABLE 3.2 – Probabilités d'occurrence de quelques séquences conduisant à l'état 1

Séquence (s_i)	**Probabilité de la séquence** $(\mathbb{P}(s_i))$
$s_1 = e_{12}e_{21}$	0.9997
$s_2 = e_{12}e_{23}e_{32}e_{21}$	0.0001
$s_3 = e_{12}e_{23}e_{32}e_{23}e_{32}e_{21}$	$1.9282 \cdot 10^{-8}$
$s_4 = e_{12}e_{23}e_{34}e_{43}e_{32}e_{21}$	$2.1591 \cdot 10^{-8}$
$s_5 = e_{12}e_{23}e_{34}e_{43}e_{34}e_{43}e_{32}e_{21}$	$3.3574 \cdot 10^{-12}$
$s_6 = e_{12}e_{23}e_{34}e_{41}$	0.0001
$s_7 = e_{12}e_{23}e_{34}e_{43}e_{34}e_{41}$	$1.7273 \cdot 10^{-8}$
$s_8 = e_{12}e_{23}e_{32}e_{23}e_{32}e_{23}e_{34}e_{41}$	$2.1415 \cdot 10^{-12}$
$s_9 = e_{12}e_{23}e_{32}e_{23}e_{32}e_{23}e_{34}e_{43}e_{34}e_{41}$	$3.3296 \cdot 10^{-16}$
$s_{10} = e_{12}e_{21}e_{12}e_{23}e_{34}e_{41}$	0.0001
$s_{11} = e_{12}e_{21}e_{12}e_{23}e_{32}e_{21}$	0.0001
$s_{12} = e_{12}e_{21}e_{12}e_{23}e_{34}e_{43}e_{34}e_{41}$	$1.7264 \cdot 10^{-8}$

TABLE 3.3 – Probabilités d'occurrence de quelques séquences conduisant à l'état 3

Séquence (s_i)	**Probabilité de la séquence** $(\mathbb{P}(s_i))$
$s_1 = e_{12}e_{23}$	0.0002
$s_2 = e_{12}e_{23}e_{32}e_{23}$	$3.4700 \cdot 10^{-8}$
$s_3 = e_{12}e_{21}e_{12}e_{23}$	0.0002
$s_4 = e_{12}e_{23}e_{34}e_{43}$	$3.8871 \cdot 10^{-8}$
$s_5 = e_{12}e_{23}e_{32}e_{23}e_{34}e_{43}$	$5.3973 \cdot 10^{-12}$
$s_6 = e_{12}e_{21}e_{12}e_{23}e_{34}e_{43}$	$3.8862 \cdot 10^{-8}$

peuvent être calculées en appliquant respectivement les équations (2.11) et (2.12). En raison de la complexité des expressions analytiques obtenues par calcul symbolique, nous ne présentons ici que les valeurs numériques pour quelques séquences appartenant au sous-langage L_1 et au sous-langage L_3 (tableau 3.2 et tableau 3.3). Quant à la somme des séquences d'événements appartenant à ces deux sous-langages, nous avons obtenu les valeurs suivantes :

$$\mathbb{Q}(L_1) = \mathbb{Q}\left(L_{(PI,TOR)}\right) = 1$$

$$\mathbb{Q}(L_3) = \mathbb{Q}\left(L_{(\overline{PI},\overline{TOR})}\right) = 4.9990 \cdot 10^{-4}.$$

L'explication du résultat obtenu pour $\mathbb{Q}(L_1)$ est identique à celle fournie pour les états initiaux des autres modes, le résultat relatif à L_3 n'a, quant à lui, pas de signification particulière.

3.5.2 Politique de maintenance imparfaite

Le deuxième comportement considéré, représenté par l'automate A_2 de la figure (3.4), correspond à la politique de réparation imparfaite. Elle est caractérisée par le fait que, après la réparation, les régulateurs ont un comportement qui n'est pas équivalent à celui avant défaillance, soit au niveau des performances, soit au niveau de leur fiabilité et disponibilité (comme cela se produit couramment dans le cas des systèmes réels).

Nous considérons donc que, contrairement à la politique de maintenance parfaite, il existe une possibilité que la réparation d'un régulateur échoue avec une probabilité donnée (P_{refPI} et

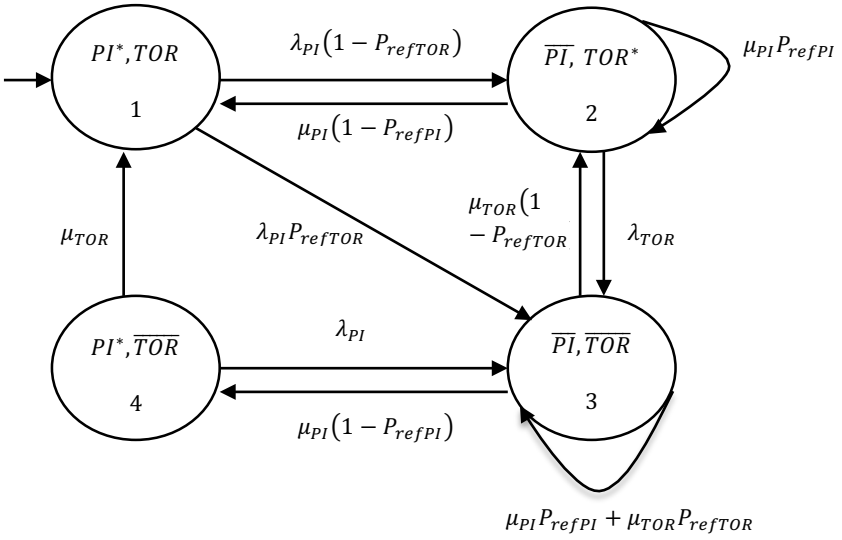


FIGURE 3.4 – Modélisation de la politique de maintenance imparfaite (A_2)

P_{refTOR}). Cela se traduit dans la chaîne de Markov à temps continu de la figure (3.4), par les transitions bouclées sur les états 2 et 3 correspondant respectivement à un échec de la réparation pour les régulateurs PI et TOR. De plus, les valeurs des taux de défaillance et de réparation sont respectivement augmentés et diminués par rapport à ceux utilisés dans le modèle de la politique de maintenance parfaite afin de prendre en compte le phénomène de dégradation : $\lambda_{PI} = 4.5 \cdot 10^{-5} h^{-1}$, $\lambda_{TOR} = 3 \cdot 10^{-5} h^{-1}$, $\mu_{PI} = 7 \cdot 10^{-2} h^{-1}$, $\mu_{TOR} = 9 \cdot 10^{-2} h^{-1}$, $P_{refPI} = 0.03$, $P_{refTOR} = 0.05$.

L'approche mise en œuvre est désormais classique : la résolution du système d'équation ci-dessous sur les sous-langages L_1, L_2, L_3 et L_4, nous fournit les expressions régulières de ces quatre sous-langages :

$$\begin{cases} L_1 = L_2 e_{21} + L_4 e_{41} \\ L_2 = L_1 e_{12} + L_2 e_{22} + L_3 e_{32} \\ L_3 = L_1 e_{13} + L_2 e_{23} + L_3 e_{33} + L_4 e_{43} \\ L_4 = L_3 e_{34} \end{cases} \tag{3.20}$$

Sachant que l'état 3 de cette politique de maintenance est un état de panne et que les séquences qui y conduisent sont considérés critiques, nous avons extrait, à titre d'exemple, deux séquences du sous-langage L_3 et nous avons calculé leur probabilités d'occurrence en obtenant les valeurs numériques suivantes :

$$s_3 = e_{13} e_{33} = 0.0021, \qquad s_4 = e_{13} e_{32} e_{23} e_{33} = 4.7215 \cdot 10^{-7}$$

Connaissant les expressions régulières des sous-langages L_1 à L_4, l'approche proposée au chapitre 2 nous permet d'obtenir la somme des probabilités des séquences leur appartenant ; à titre d'exemple, les probabilités obtenues pour L_1 et L_3 sont :

$$\mathbb{Q}(L_1) = \mathbb{Q}\left(L_{(PI^*,TOR)}\right) = 1, \qquad \mathbb{Q}(L_3) = \mathbb{Q}\left(L_{(\overline{PI},\overline{TOR})}\right) = 0.0526.$$

L'explication du résultat obtenu pour $\mathbb{Q}(L_1)$ est identique à celle fournie pour les états initiaux des autres modes, le résultat relatif à L_3 n'a, quant à lui, pas de signification particulière.

3.5.3 Composition modulaire des politiques de maintenance

Les probabilités d'occurrence des séquences ayant pu être calculées pour chacune des deux politiques de maintenance, l'objectif est maintenant de faire de même pour le système global qui intègre ces deux politiques de maintenance. L'opérateur de concaténation permet d'atteindre cet objectif sans construire explicitement le modèle global (par composition synchrone des automates par exemple).

On note que l'ensemble des événements Σ, appartenant aux automates caractérisant les deux politiques de maintenance, est exprimé comme l'union entre trois ensembles d'événements : les défaillances et les réparations des composants, et l'échec de la réparation :

$$\Sigma = \Sigma_f \cup \Sigma_r \cup \Sigma_{ref} \tag{3.21}$$

Notons, que relativement aux principes généraux énoncés dans la section 3.3.3, ce sont les événements de réparation (Σ_r) qui jouent ici le rôle d'événements potentiellement générateurs d'une commutation de mode (maintenance parfaite vers maintenance imparfaite). Par conséquent, les probabilités des différentes séquences d'événements intégrant les deux politiques de maintenance seront obtenues, via l'opérateur de concaténation, en remplaçant Σ_c par Σ_r dans l'équation (3.4) :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot_p \mathbb{L}_2}\right) = \begin{cases} \mathbb{P}\left(s_i^{\mathbb{L}_1}\right) & \text{si } s_i \in A_1 \\ p \sum_{\substack{t < s_{i-1} e_{(i-1)i} \\ e_{(i-1)i} \in \Sigma_r}} \mathbb{P}\left((te_\Delta)^{\mathbb{L}_1}\right) \mathbb{P}\left(\left(t^{-1} s_i\right)^{\mathbb{L}_2}\right) & \text{sinon} \end{cases} \tag{3.22}$$

Si la séquence s_i appartient entièrement à l'automate A_1 qui décrit la politique de maintenance parfaire, $\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot_p \mathbb{L}_2}\right)$ est donné par la probabilité de l'occurrence de la séquence s_i dans l'automate A_1. Après une réparation, le système peut, avec une probabilité p, commuter vers un comportement correspondant à la politique de maintenance imparfaite : un préfixe te_Δ de la séquence s_i suit le comportement de l'automate A_1 (le dernier événement du préfixe doit toujours être un événement de réparation appartenant à Σ_r) et le reste de la séquence $(t^{-1}s_i)$ suit le comportement de l'automate A_2. En d'autres termes, le système suit d'abord le comportement défini dans le cadre de la politique de maintenance parfaite puis, après une réparation, peut commuter vers un comportement décrit dans le cadre d'une politique de maintenance imparfaite.

Le tableau (3.4) présente les résultats numériques obtenus pour la probabilité d'occurrence de quelques séquences d'événements qui conduisent le système à l'état 3 (état de panne). Certaines séquences correspondent à des sous-ensembles de traces qui ne s'exécutent que dans l'un des deux modes, d'autres correspondent à des traces qui comprennent la commutation entre les modes (système global).

Notons que, pour obtenir des résultats équivalents dans le cadre d'une approche non modulaire, cela supposerait de définir un modèle global intégrant les deux politiques de maintenance. Cela pourrait se faire, par exemple, sous la forme d'un Processus de Markov Pilotés [Bouissou et Bon, 2003] qui est défini par deux chaînes de Markov et deux fonctions de transfert probabilistes permettant la commutation entre les états des deux chaînes. Néanmoins, nous avons souligné, en fin du chapitre 2, les limites, dues à la taille, relatives à la construction et l'exploitation de ce type de modèle non modulaire pour des systèmes réels potentiellement plus complexes que cet exemple. L'intérêt de l'opérateur de concaténation est de permettre l'évaluation des séquences relatives au système global sans nécessiter la construction du modèle global.

TABLE 3.4 – Probabilités d'occurrence des séquences d'événements appartenant au modèle global du système qui intègre les deux politiques de maintenance

Séquence (s_i)	A_1	A_2	**Système global**	**Probabilité de la séquence** $(\mathbb{P}(s_i))$
$s_1 = e_{12}e_{23}$	x			$2.4994 \cdot 10^{-4}$
$s_2 = e_{12}e_{21}e_{12}e_{22}e_{21}e_{13}$			x	$6.8982 \cdot 10^{-5}$
$s_3 = e_{12}e_{23}e_{32}e_{23}$	x			$3.4705 \cdot 10^{-8}$
$s_4 = e_{12}e_{23}e_{34}e_{43}$	x			$3.8866 \cdot 10^{-8}$
$s_5 = e_{13}e_{33}$		x		$1.0313 \cdot 10^{-4}$
$s_6 = e_{13}e_{34}e_{43}$		x		$5.3020 \cdot 10^{-7}$
$s_7 = e_{13}e_{32}e_{23}e_{33}$		x		$2.3607 \cdot 10^{-8}$
$s_8 = e_{12}e_{21}e_{12}e_{22}e_{23}e_{32}e_{23}e_{33}$			x	$5.7605 \cdot 10^{-12}$
$s_9 = e_{12}e_{23}e_{32}e_{23}e_{33}e_{34}e_{43}$			x	$2.6019 \cdot 10^{-14}$
$s_{10} = e_{12}e_{23}e_{34}e_{43}e_{33}$			x	$1.1449 \cdot 10^{-10}$

3.6 Généralisation des opérateurs de composition modulaire

L'approche modulaire, que nous avons proposée sur la bas des opérateurs de choix et de concaténation, a été définie pour intégrer deux modes de défaillances ou de fonctionnement différents. Nous avons, par exemple, illustrer l'usage de ces deux opérateurs pour intégrer, d'une part, deux modes de défaillances (indépendantes et de cause commune), et d'autre par, deux politiques de maintenance. Ces opérateurs ont en effet été initialement définis dans le cadre de la théorie de langages probabilistes [Garg *et al.*, 1999] pour deux arguments.

Dans cette section, nous proposons une généralisation des opérateurs de choix et de concaténation afin de permettre l'analyse de séquences d'événements pour des systèmes possédant n $(n > 2)$ modes.

3.6.1 Généralisation de l'opérateur de choix

Selon la définition (7), l'opérateur de choix a été défini sur deux p-langages : $\mathbb{L}_1$, $\mathbb{L}_2$. Pour prendre en considération n modes de fonctionnement ou de défaillances et n probabilités différentes qui leur sont associées, nous allons définir l'opérateur de choix comme une combinaison convexe de ses arguments (les n p-langages).

Définition 9. *: Soit un nombre n de p-langages* $\mathbb{L}_1$, $\mathbb{L}_2$, ..., $\mathbb{L}_n$ *et* $p_1, p_2, ..., p_{n-1} \in [0, 1]$*, l'opérateur de choix entre ces p-langages, noté par* $\mathbb{L}_1 +_{p_1} \mathbb{L}_2 +_{p_2} ... +_{p_{n-1}} \mathbb{L}_n$*, est défini par :*

$$\mathbb{P}(\langle s\rangle^{\mathbb{L}_1+_{p_1}\mathbb{L}_2+_{p_2}...+_{p_{n-1}}\mathbb{L}_n}) = p_1 \cdot \mathbb{P}\left(\langle s\rangle^{\mathbb{L}_1}\right) + p_2 \cdot \mathbb{P}\left(\langle s\rangle^{\mathbb{L}_2}\right) + ... + (1 - (p_1 + p_2 + ... + p_{n-1})) \cdot \mathbb{P}\left(\langle s\rangle^{\mathbb{L}_n}\right) \tag{3.23}$$

Chacun des n modes est représenté par un automate à états finis A_i, qui reconnaît le p-langage $\mathbb{L}_i$, la probabilité d'une séquence appartenant au système intégrant les n modes est donc obtenue en appliquant la définition généralisé de l'opérateur de choix (9) :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1+_{p_1}\mathbb{L}_2+_{p_2}...+_{p_{n-1}}\mathbb{L}_n}\right) = p_1 \cdot \mathbb{P}\left(s_i^{\mathbb{L}_1}\right) + p_2 \cdot \mathbb{P}\left(s_i^{\mathbb{L}_2}\right) + ... + (1 - (p_1 + p_2 + ... + p_{n-1})) \cdot \mathbb{P}\left(s_i^{\mathbb{L}_n}\right) \tag{3.24}$$

En d'autres termes, l'opérateur généralisé de choix traduit le fait qu'un système comportant n modes se comporte conformément à chacun de ses n modes avec les probabilités $p_1, p_2, ..., p_{n-1}$ qui sont supposées connues : il se comporte selon le premier mode représenté par un automate noté avec A_1 (qui reconnaît le p-langage $\mathbb{L}_1$) avec une probabilité p_1 ou bien comme le deuxième mode représenté par un automate noté avec A_2 (qui reconnaît le p-langage $\mathbb{L}_2$) avec une probabilité p_2 et ainsi de suite jusqu'au n^{eme} mode représenté par un automate noté avec A_n (qui reconnaît le p-langage $\mathbb{L}_n$) avec la probabilité $(1 - (p_1 + p_2 + ... + p_{n-1}))$.

De manière similaire à l'opérateur de choix agissant sur deux arguments, il est possible de réaliser une composition des sous-langages associés aux état des automates A_1, A_2, ..., A_n pour déterminer la somme des probabilités des séquences d'événements qui appartiennent aux sous-langages associés aux états du système global intégrant les n modes :

$$\mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_1+_{p_1}\mathbb{L}_2+_{p_2}...+_{p_{n-1}}\mathbb{L}_n}\right) = p_1 \cdot \mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_1}\right) + p_2 \cdot \mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_2}\right) + ... + (1 - (p_1 + p_2 + ... + p_{n-1})) \cdot \mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_n}\right) \tag{3.25}$$

où $\mathbb{Q}\left(L_{(x_i)}^{\mathbb{L}_k}\right)$ représente la somme des probabilités des séquences appartenant au sous-langage $L_{(x_i)}$ associé à l'état x_i dans l'automate A_k qui reconnaît le p-langage $\mathbb{L}_k$.

Notons que ces probabilités s'obtiennent sans avoir à construire le modèle global du système (dont la taille croit exponentiellement en fonction du nombre d'états des n automates). La version généralisée de l'opérateur de choix est donc tout à fait adaptée dans le cas de systèmes (industriels) complexes présentant de multiples modes de fonctionnement ou de défaillance.

3.6.2 Généralisation de l'opérateur de concaténation

L'opérateur de concaténation a été défini, initialement, sur deux arguments (définition 8). Conformément à sa définition, il permet la modélisation du fonctionnement séquentiel de deux phases ou de deux modes de fonctionnement d'un même système. De plus, il ne permet la modélisation que d'une seule commutation, d'un premier mode vers un second. Son application dans le cadre de nos travaux a concerné l'enchaînement de comportements nominaux et dégradés caractérisant des politiques de réparation respectivement parfaites et imparfaites. La généralisation de cet opérateur a pour objectif de permettre le traitement de systèmes possédant n modes de fonctionnement ou de défaillances.

Définition 10. *: Soit n p-langages $\mathbb{L}_1$, $\mathbb{L}_2$, ..., $\mathbb{L}_n$ et $p_1, p_2, ..., p_{n-1} \in [0, 1]$, l'opérateur de concaténation, noté $\mathbb{L}_1 \cdot_{p_1} \mathbb{L}_2 \cdot_{p_2} ... \cdot_{p_{n-1}} \mathbb{L}_n$, est défini par :*

$$\forall \langle s \rangle \in \Sigma^*, \mathbb{P}(\langle s \rangle^{\mathbb{L}_1 \cdot_{p_1} \mathbb{L}_2 \cdot_{p_2} ... \cdot_{p_{n-1}} \mathbb{L}_n}) = \mathbb{P}(\langle s \rangle^{\mathbb{L}_1}) +$$

$$p_1 \sum\nolimits_{t_1 < t_2} \mathbb{P}(\langle t_1 e_\Delta \rangle^{\mathbb{L}_1}) p_2 \sum\nolimits_{t_2 < t_3} \mathbb{P}(\langle t_2 e_\Delta \rangle^{\mathbb{L}_2}) ... p_{n-1} \sum\nolimits_{t_{n-1} < s} \mathbb{P}(\langle t_{n-1} e_\Delta \rangle^{\mathbb{L}_{n-1}}) \mathbb{P}(\langle t_{n-1}^{-1} s \rangle^{\mathbb{L}_n}) \tag{3.26}$$

Cette définition généralisée de l'opérateur de concaténation doit permettre de couvrir plusieurs cas de figure :

- le système possède n modes successifs de fonctionnement, toutes les commutations s'effectuant séquentiellement d'un mode vers le suivant,
- le système ne possède que deux modes mais présente de multiples commutations successives entre ces deux modes ;
- le système possède n modes et les commutations entre ces modes peuvent être multiples.

3.6.2.1 Commutations sur n modes successifs

En appliquant la définition 10, les probabilités de différentes séquences d'événements, associées au comportement global du système intégrant les n modes, sont données par :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot p_1 \mathbb{L}_2 \cdot p_2 \cdots \cdot p_{m-1} \mathbb{L}_m}\right) =$$

$$\begin{cases} \mathbb{P}\left(s_i^{\mathbb{L}_1}\right), \text{ si } s_i \in A_1 \\ p_1 \sum_{t_1<t_2} \mathbb{P}(\langle t_1 e_\Delta \rangle^{\mathbb{L}_1}) p_2 \sum_{t_2<t_3} \mathbb{P}(\langle t_2 e_\Delta \rangle^{\mathbb{L}_2}) ... p_{m-1} \sum_{t_{m-1}<s_i} \mathbb{P}(\langle t_{m-1} e_\Delta \rangle^{\mathbb{L}_{m-1}}) \mathbb{P}(\langle t_{m-1}^{-1} s_i \rangle^{\mathbb{L}_m}), \text{sinon} \end{cases} \tag{3.27}$$

où m est le nombre de modes présents dans la séquence. Autrement dit, il est possible de déterminer les probabilités d'occurrence pour des séquences qui contiennent des événements n'appartenant qu'à m modes parmi les n (en sachant que $m \leq n$).

L'équation (3.27) traduit deux possibilités pour l'exécution d'une séquence dans le cadre du modèle global :

- soit il exécute une séquence s_i appartenant entièrement au p-automate A_1 qui reconnaît le p-langage $\mathbb{L}_1$,
- soit, avec une probabilité p_1, il exécute un préfixe $t_1 e_\Delta$ de s_i dans A_1, puis, avec une probabilité p_2, il exécute un préfixe $t_2 e_\Delta$ de la séquence s_i dans A_2 et ainsi de suite jusqu'à l'exécution de la fin de la séquence $t_{m-1}^{-1} s_i$ dans le p-automate A_m avec une probabilité p_{m-1} ; comme dans le cas de l'opérateur de concaténation initial, chaque préfixe exécuté doit avoir, pour dernier événement, un événement potentiellement déclencheur d'une commutation entre modes.

3.6.2.2 Commutations multiples sur deux modes

Ce cas correspond à de multiples commutations entre deux modes 1 et 2. Une séquence d'événements finie s_i sera exécutée en suivant le comportement du premier mode, puis en suivant celui du second avant de revenir au premier mode et ainsi de suite, ce cycle de commutation pouvant se répéter à l'infini. Si on considère que la séquence finie aboutit au mode k (avec $k = 1$ ou 2), l'équation (3.27) est instanciée sur deux langages L_1 et L_2 :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot p_{12} \mathbb{L}_2 \cdot p_{21} \mathbb{L}_1 \cdot p_{12} \cdots \cdot p_{qk} \mathbb{L}_k}\right) =$$

$$\begin{cases} \mathbb{P}\left(s_i^{\mathbb{L}_1}\right), \text{ si } s_i \in A_1 \\ p_{12} \sum_{t_1<t_2} \mathbb{P}(\langle t_1 e_\Delta \rangle^{\mathbb{L}_1}) p_{21} \sum_{t_2<t_3} \mathbb{P}(\langle t_2 e_\Delta \rangle^{\mathbb{L}_2}) ... p_{qk} \sum_{t_{m-1}<s_i} \mathbb{P}(\langle t_{m-1} e_\Delta \rangle^{\mathbb{L}_q}) \mathbb{P}(\langle t_{m-1}^{-1} s_i \rangle^{\mathbb{L}_k}), \text{sinon} \end{cases} \tag{3.28}$$

où $m-1$ représente le nombre de commutations présentes dans la séquence, k représente l'indice du mode dans lequel se termine la séquence ($k = 1$ si la séquence se termine dans le premier mode, $k = 2$ si la séquence se termine dans le second mode) et q représente l'indice du mode actif avant la dernière commutation de la séquence ($q = 2$ si $k = 1$ et $q = 1$ si $k = 2$). Le calcul de cette probabilité suppose, bien entendu, de connaître le mode atteint par le dernier événement de commutation (p-langage $\mathbb{L}_1$ ou p-langage $\mathbb{L}_2$). Cette connaissance est disponible via la description de la séquence dont on cherche à calculer la probabilité.

3.6.2.3 Commutations multiples sur n modes

Le dernier cas est une généralisation des deux premiers. La définition 10 reste valable, le calcul de probabilité d'une séquence s_i sera une combinaison des équations (3.27) et (3.28). Cette combinaison tiendra compte de la connaissance des modes successivement activés lors de l'exécution de la séquence dont on cherche à calculer la probabilité. Les langages L_1, L_2, ..., L_m correspondront aux modes successivement activés au cours de cette séquence et les probabilités p_1, p_2, ..., p_{m-1} correspondront aux

probabilités associées aux commutations entre les modes parcourus. Le nombre de sommes (Σ) présentes dans l'équation correspondra au nombre de commutations présentes dans la séquence.

Pour conclure, notons que, quelque soit la configuration retenue, le calcul des probabilités des séquence est réalisé sans avoir à construire le modèle global du système, ce qui constitue un argument significatif en faveur de cette approche pour l'analyse de systèmes complexes multi-phases ou multi-modes, notamment dans le cadre d'applications industrielles.

3.7 Conclusion

Compte tenu des limites de l'approche proposée au chapitre 2 pour traiter des problèmes à échelle industrielle, notamment liées à la taille et la complexité des modèles manipulés, ce chapitre a complété le cadre formel proposé initialement par l'introduction d'opérateurs de composition. Ces derniers supportent une approche modulaire dont l'atout majeur est de permettre le calcul de la probabilité d'occurrence de séquences critiques pour des systèmes caractérisés par plusieurs modes de fonctionnement sans nécessiter la construction d'un modèle global intégrant ces différents modes.

Nous avons en particulier montré, sur la base d'exemples combinant deux modes de défaillance ou deux politiques de maintenance, que les opérateurs de choix et concaténation définis dans le cadre de la théorie de langages probabilistes [Garg *et al.*, 1999] permettait de procéder à une évaluation quantitative globale des séquences à partir des évaluations obtenues localement, mode par mode, sur des modèles de taille raisonnable. D'un point de vue très abstrait, l'opérateur de choix peut être vus comme un opérateur de composition "parallèle" pour des modes simultanément actifs alors que l'opérateur de concaténation peut être assimilé à un opérateur de composition "série" pour des modes séquentiels (ou des phases).

Nous avons ensuite généralisé les définitions des opérateurs de choix et de concaténation de telle sorte à pouvoir traiter plusieurs modes (alors que les définitions initiales se limitent à deux modes) et/ou plusieurs commutations dans le cas de l'opérateur de concaténation.

Pour conclure, l'approche proposée dans ce chapitre devrait être particulièrement pertinente pour des systèmes de grande taille aux multiples modes de fonctionnement pour lesquels l'exploitation d'un automate obtenu par composition d'automates locaux s'avère difficile. Le chapitre suivant va s'attacher à mettre en évidence son apport au travers d'un cas d'étude industriel.

Chapitre 4

Application sur un cas test industriel

4.1 Introduction

Le cadre formel défini aux chapitres 2 et 3 a été appliqué pour l'analyse qualitative et quantitative de séquences d'événements sur le cas d'étude relatif à la commande de la température d'un four qui est de taille académique. L'objectif de ce chapitre est de présenter une application de nos contributions sur un cas d'étude industriel défini dans le cadre du projet de recherche *APPRODYN* [Aubry *et al.*, 2012a] impliquant un partenaire industriel (*EDF R&D*) et trois partenaires universitaires (*CRAN-CNRS/Université de Lorraine, CQFD-INRIA/Université de Bordeaux et ICD-Université de Technologie de Troyes*). Entre 2010 et 2012, le but de ce projet a été d'expérimenter des approches de la fiabilité dynamique afin de supporter l'étude probabiliste de la sûreté de fonctionnement des systèmes de contrôle-commande critiques utilisés dans les domaines de la production d'énergie et des industries de procédé. L'indicateur analysé était la disponibilité en vue de diminuer les pertes de production et le vieillissement prématuré des composants matériels du système au travers de l'optimisation des stratégies de commande, d'exploitation et de maintenance.

Le chapitre débute par la présentation du cas test industriel, puis présente la modélisation des différents modes de fonctionnement et de défaillance sous la forme d'automates. La dernière partie est consacrée à l'identification et la quantification de séquences d'événements en régime transitoire et asymptotique.

4.2 Présentation du cas test

4.2.1 Cas test du projet APPRODYN

4.2.1.1 Architecture et principe de fonctionnement

Le système retenu par le projet APPRODYN est un système de régulation du niveau d'eau dans un générateur de vapeur (GV) d'un Réacteur nucléaire à Eau Pressurisée (REP). Un générateur de vapeur est, en pratique, un échangeur thermique entre le circuit primaire et le circuit secondaire d'une centrale de production d'électricité (figure 4.1). Afin de fixer les ordres de grandeurs, nous donnons ci-dessous quelques caractéristiques d'un GV représentatif de l'industrie nucléaire :

- Surface d'échange : 4746 m^2 (3330 tubes)
- Hauteur : 20.60 m
- Durée de vie : environ 15 ans ;
- Débit vapeur : 1820 t/hr
- Diamètre : 4.50 m
- Soupapes tarées à 76.60 b

Le circuit secondaire se compose, dans le sens de la circulation des fluides de multiples systèmes : le barillet qui concentre la vapeur en sortie des Générateurs de Vapeur (GV) sur le circuit vapeur principal (VVP), les turbines haute pression et basse pression, le condenseur, les pompes d'extraction (CEX), les réchauffeurs basse pression (ABP), la bâche alimentaire et le dégazeur (ADG), les turbopompes

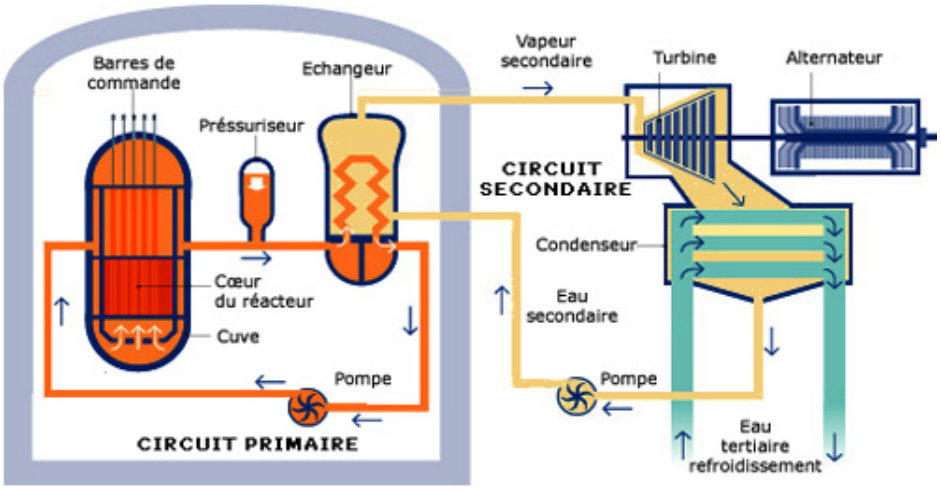


FIGURE 4.1 – Schéma de principe d'un REP

alimentaires (TPA), les réchauffeurs haute pression (AHP) et enfin les vannes réglantes (ARE). La figure (4.2) présente la partie située en avale du condenseur.

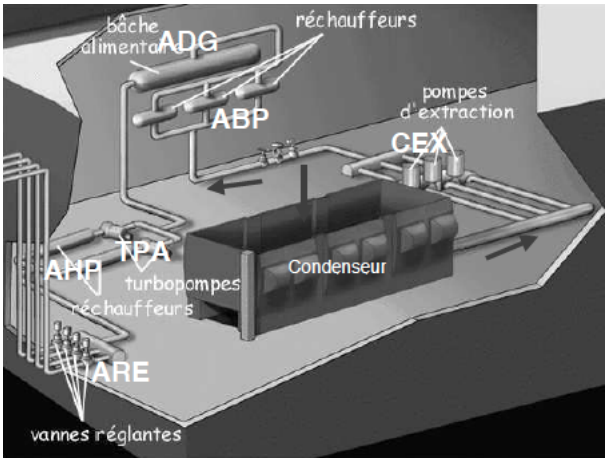


FIGURE 4.2 – Partie en aval du condenseur d'un circuit secondaire d'un REP

Le système considéré dans APPRODYN se concentre sur un sous-ensemble du circuit secondaire chargé de maintenir le niveau d'eau dans le GV, autour d'une position de référence déterminée en fonction de la puissance requise (P) exprimée sous la forme d'un pourcentage de la puissance nominale (P_n). Le cas test est composé d'un système situé en amont du condenseur (le barillet) et de trois systèmes situés en aval du condenseur (les pompes d'extraction, les turbopompes et les vannes réglantes) :

- le barillet VVP concentre la vapeur en sortie des Générateurs de Vapeur (GV) sur le circuit vapeur principal (VVP) ; il permet de maintenir le fonctionnement des turbopompes, sécheurs, *etc.* même en cas de perte d'un GV ;
- les trois pompes CEX maintiennent le vide au condenseur ;
- les deux turbopompes TPA assurent la pression commune aux trois GV, elles permettent d'assurer un débit d'eau alimentaire.
- un système de vannes réglantes ARE règle le débit pour chaque GV ; il est formé d'une vanne petit débit (0 à 400 tonnes/h) et d'une vanne gros débit (0 à 1815 tonnes/h). La vanne petit débit, plus réactive, est employée de $2\%P_n$ jusqu'à $15\%P_n$ environ, sachant qu'elle peut fournir jusqu'à $23\%P_n$. La vanne gros débit est employée à partir de $15\%P_n$ environ. Une régulation assure le

basculement entre les deux vannes. Le temps de réponse de la vanne en ouverture/fermeture permet de suivre les variations de puissance de l'installation.

Le profil de mission retenu dans le cadre du cas test est représenté sur la figure (4.3). Il se compose, en régime nominal, d'un fonctionnement à puissance variable, comprise entre 60% et 100% de la puissance nominale P_n. La phase de démarrage depuis 0% de la puissance nominale P_n suit une rampe de montée en puissance qui dure au minimum 24 heures (sans aléas) pour atteindre les 100% de P_n. La phase de descente en puissance est symétrique. Ce scénario peut subir des perturbations, en particulier des arrêts de production pour des causes diverses (arrêt turbine, arrêt du réacteur, *etc.*), autres que celles imputables à la régulation des GV. En cas d'Arrêt Automatique du Réacteur (AAR), la tranche est à l'arrêt ($0\%P_n$) pendant le temps nécessaire à l'identification et à la correction de la cause de l'AAR (estimée à 8 heures sauf en cas d'avarie grave).

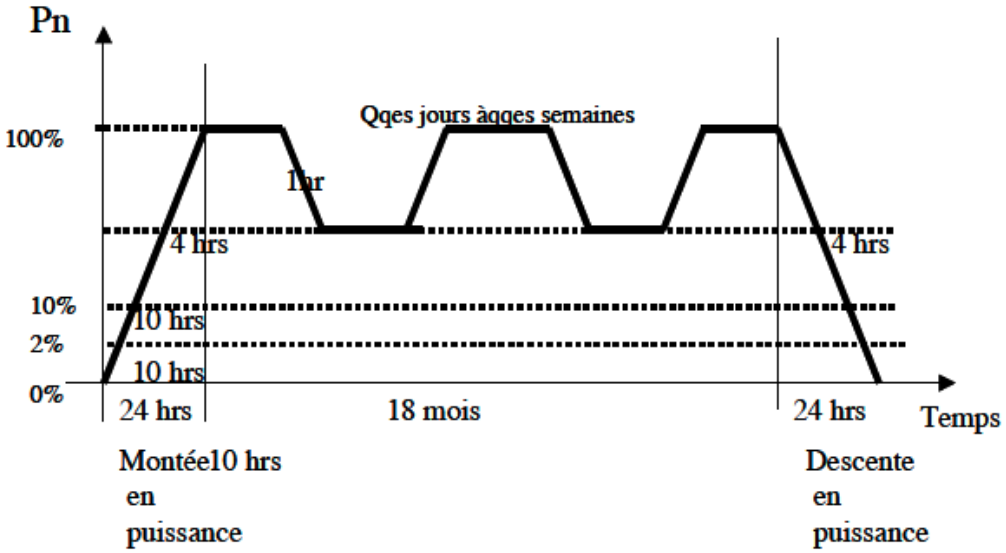


FIGURE 4.3 – Scénario de fonctionnement à puissance variable

4.2.1.2 Données de sûreté de fonctionnement

D'un point de vue de la sûreté de fonctionnement, un échec de la mission se traduit par une Indisponibilité Fortuite (IF) qui entraîne l'activation de systèmes de secours, ou dans le pire des cas, l'Arrêt Automatique du Réacteur (AAR) qui est l'événement à éviter. Le diagramme de fiabilité du cas test considéré est représenté sur la figure (4.4).

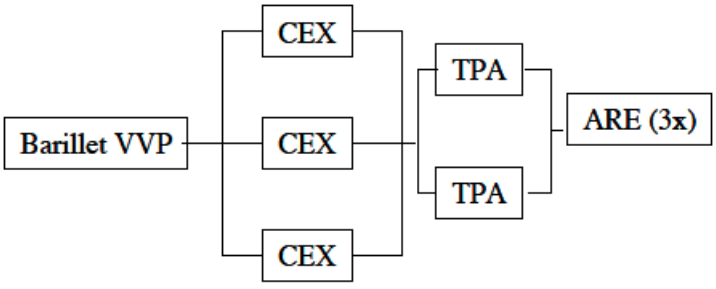


FIGURE 4.4 – Diagramme de fiabilité du cas test APPRODYN

Une rupture du barillet VVP est une défaillance de point unique ; elle représente un niveau minimal de fiabilité du système. Les défaillances du barillet VVP permettent aussi de représenter les défaillances des autres systèmes passifs (bâches, réchauffeurs, barillet en aval des vannes ARE).

Les pompes d'extraction CEX sont redondées en 2/3, la troisième pompe étant à l'arrêt, en attente. Elle est démarrée lorsqu'une des deux autres pompes tombe en panne. La pompe en panne, une fois réparée, reste en attente.

Les turbopompes fonctionnent simultanément. En cas de défaillance d'une TPA, l'autre passe en sur-vitesse et assure 60% de la charge. On considère que la puissance de l'installation ne peut dépasser 60% de sa puissance nominale ($60\%P_n$) lorsqu'une seule TPA fonctionne.

Les défaillances de ces équipements (CEX, TPA, Barillet VVP, ARE) peuvent avoir des origines diverses dues au système de commande, aux capteurs ou aux actionneurs.

En ce qui concerne la commande, la plupart des équipements sont dotés de systèmes de commande réflexe et de régulation implantés dans des automates programmables industriels. La complexité des algorithmes est sensiblement équivalente pour les CEX, les TPA ou les ARE dans la mesure où tous ces systèmes sont non-linéaires. Dans la pratique, les régulateurs sont souvent de type PID. La régulation de niveau est directement assurée par les vannes réglantes ARE, le cas test APPRODYN se concentre sur l'analyse de la commande (PID et instrumentation) de ce système.

En ce qui concerne les capteurs, le cas test considère des transmetteurs analogiques conventionnels permettant la mesure de débit vapeur (Qv), de débit eau (Qe), niveau Nge eau/vapeur en gamme étroite et niveau Ngl d'eau en gamme large. La précision et la fiabilité des capteurs de niveau évolue en fonction du temps (par effet de vieillissement) et dépend des niveaux atteints par les paramètres physiques dans le GV ; cet aspect relève de la fiabilité dynamique.

En ce qui concerne les actionneurs, le cas test du projet APPRODYN les décrit comme totalement intégrés aux systèmes auxquels ils sont associés (vannes réglantes ARE, turbopompes TPA, pompes d'extraction CEX, ...). Au delà des défaillances intrinsèques (notamment défaillance matérielle) de ces différents équipements, il faudra prendre en compte les défaillances affectant les actionneurs conventionnels (défaillance à la sollicitation, blocage, *etc.*) et les intégrer dans les défaillances à la sollicitation et la proportion de défaillances non détectables des équipements.

Pour conclure cette section consacrée aux paramètres de sûreté de fonctionnement du cas test, le tableau (4.1) fournit les valeurs numériques des défaillances associés aux composants du système. Ils reposent sur les observations sur 34 tranches, exploitées depuis 1992, selon des cycles de 18 mois, interrompus d'arrêts techniques de 1 à 3 mois mais prennent également en compte des correctifs issus du retour d'expérience en exploitation. Dans le projet APPRODYN, les taux de défaillance sont supposés constants même s'il pouvait être intéressant de prendre en compte certains phénomènes de vieillissement. Ils peuvent, en absence de maintenance adaptée, aggraver la fréquence des défaillances unitaires des AAR, le risque d'échec en réparation ou d'échec à la sollicitation.

En plus des données de fiabilité (taux de défaillance et probabilité de défaillance à la sollicitation) présentés dans le tableau 4.1, les données suivantes seront nécessaires pour caractériser la maintenabilité des composants (notamment celle des turbopompes TPA) :

- le taux de réparation de la partie hors-turbine (HT) des TPA suite à une défaillance en fonctionnement qui est égal à $0.0417h^{-1}$,
- le taux de réparation de la partie hors-turbine (HT) des TPA suite à une défaillance à la sollicitation qui est égal à $0.0357h^{-1}$,
- le taux de réparation de la partie turbine (T) des TPA suite à une défaillance en fonctionnement qui est égal à $0.0417h^{-1}$,
- le taux de réparation de la partie turbine (T) des TPA suite à une défaillance à la sollicitation qui est égal à $0.0417h^{-1}$,
- la probabilité de succès d'une opération de réparation est identique pour les quatre types de réparation présentés ci-dessus et est égale à 0.9.

Enfin, une probabilité de succès des fonctions de régulation est à considérer et elle est égale à 0.9.

TABLE 4.1 – Taux de défaillance et probabilité de défaillance à la sollicitation des composants

Composant	**Taux de défaillance (h^{-1})**	**Probabilité de défaillance à la sollicitation**
Barillet VVP	$2.17 \cdot 10^{-5}$	
Pompe CEX	$4.35 \cdot 10^{-5}$	$1.95 \cdot 10^{-3}$
TPA Hors Turbine	$1.5 \cdot 10^{-4}$	$5.5 \cdot 10^{-4}$
TPA Turbine	$4.4 \cdot 10^{-4}$	$3.90 \cdot 10^{-5}$
Vanne ARE	$3.53 \cdot 10^{-5}$	$5.85 \cdot 10^{-3}$
Capteur de mesure de niveau	$5.20 \cdot 10^{-6}$	
Capteur de mesure de débit eau alimentaire	10^{-4}	
Capteur de mesure de débit vapeur	10^{-4}	
Capteur de mesure de niveau (gamme large)	$1.70 \cdot 10^{-6}$	
Instrumentation CEX	$6.50 \cdot 10^{-6}$	$1.95 \cdot 10^{-3}$
Instrumentation TPA	$1.85 \cdot 10^{-6}$	$5.45 \cdot 10^{-4}$
Instrumentation ARE	$2 \cdot 10^{-5}$	$5.85 \cdot 10^{-3}$

4.2.2 Cas test retenu pour notre étude : sous-système de turbopompes

Dans le cadre de notre étude, nous avons retenu le sous-système des deux turbopompes (TPA) qui assurent la pression commune aux trois GV. Ce cas test a pour intérêt de représenter un système réel tout en englobant des situations plus académiques employées dans la littérature. Il peut également être assez facilement adapté à d'autres sources d'énergie ou d'autres situations rencontrées dans des industries de procédé. De plus, il présente une complexité en termes de modes de fonctionnement et de défaillances significative pour l'application de notre démarche ; sa taille est suffisante pour expérimenter le passage à l'échelle de notre approche puisque la modélisation du sous-système TPA réalisée dans le cadre du projet APPRODYN a nécessité des automates à 32 états et 54 transitions pour chaque TPA et un automate à 96 états et 145 transitions pour la coordination des deux TPA, ce qui représente un modèle global, si l'on devait le construire, d'approximativement 100.000 états [Babykina *et al.*, 2016].

Chaque TPA est constituée d'une partie turbine (T) et d'une partie hors turbine (HT) qui forment du point de vue fonctionnel un système en série (figure 4.5). Si une des partie (T ou HT) d'une TPA tombe en panne, l'autre partie de cette TPA est mise à l'arrêt (les deux parties sont nécessaires pour que la TPA fonctionne) et sa défaillance ne peut plus se produire.

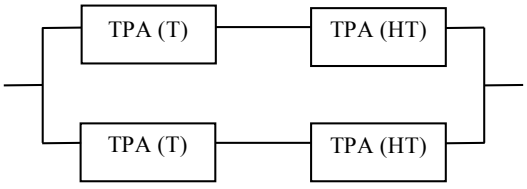


FIGURE 4.5 – Sous-système de deux TPA

Trois grandes phases de fonctionnement caractérisent le comportement du sous-système TPA : démarrage et montée en puissance, conduite en puissance, baisse de puissance.

4.2.2.1 Démarrage et montée en puissance des turbopompes (TPA)

Pour une puissances $P < 2\%P_n$, les deux TPA sont inactives. Lorsque la puissance atteint $2\%P_n$ et que le niveau des GV est à au moins 33% de la gamme étroite, une TPA est mise en marche, et les vannes ARE prennent le relais du système d'alimentation de secours pour l'alimentation en eau (le système de secours est également utilisé dans les toutes premières phases de démarrage compte tenu du dimensionnement parfaitement adapté de ses équipements).

La montée en puissance requiert deux TPA en état, une en marche, une en attente. Si la première TPA (TPA 1) ne démarre pas, la seconde (TPA 2) est mise en marche. En cas de succès de ce démarrage, des réparations sont effectuées sur la première TPA et la puissance est maintenue à $2\%P_n$. En revanche, si la TPA 2 ne démarre pas, des réparations sont effectuées sur la TPA 2 puis sur la TPA 1 à $2\%P_n$.

Pour une puissance $P < 60\%P_n$, une TPA est employée, éventuellement en sur-vitesse, la seconde reste en redondance passive. Lorsque la puissance P atteint $60\%P_n$, la seconde TPA est mise en marche. Si elle ne démarre pas, des réparations sont effectuées à $60\%P_n$.

La montée en puissance au delà de $60\%P_n$ requiert deux TPA en marche. Si une pompe TPA tombe en panne à plus de $60\%P_n$, l'autre passe en sur-vitesse et assure 60% de la charge. La puissance est maintenue à $60\%P_n$ tant qu'une seule TPA fonctionne. Des que la seconde TPA est réparée, elle est remise en marche et la puissance repart à la hausse jusqu'à $100\%P_n$. Si la seconde TPA ne démarre pas ou tombe en panne, cela conduit à l'AAR ou bien à un forçage à $2\%P_n$.

4.2.2.2 Conduite en puissance des turbopompes (TPA)

Lorsqu'une TPA tombe en panne à une puissance supérieure à $60\%P$, la consigne est de ramener la puissance à $60\%P_n$ par une action sur les fonctions de régulation (ce qui prend quelques minutes). On est alors à $60\%P_n$ avec une TPA en marche et une TPA en réparation (forçage à 60%). Cette action peut échouer à la sollicitation et cette situation mène à un AARSi la seconde TPA tombe en panne, cette situation mène aussi à un AAR. La puissance est maintenue à $60\%P_n$ (forçage à 60%) en attendant que la seconde TPA soit réparée, puis on remonte en charge. Le temps d'attente de la seconde TPA est trop court pour qu'un essai périodique soit mené.

Ce comportement détermine le profil de mission donné dans la figure (4.3). Le sous-système de turbopompes TPA est le seul sous-système parmi les sous-systèmes de la figure (4.4) (VVP, CEX, TPA et ARE) responsable de ce fonctionnement à puissance variable pendant la phase de conduite en puissance ; les autres sous-systèmes ayant un comportement binaire (marche ou panne, cette dernière déterminant un AAR) par rapport à la mission du système complet. C'est aussi une raison supplémentaire pour laquelle nous avons retenu dans notre étude le sous-système des turbopompes TPA, celui-ci permettant d'avoir un profil de mission réaliste pour le système complet.

4.2.2.3 Baisse de puissance des turbopompes TPA

Dans ce mode, les régulations de la source thermique sont utilisées pour faire baisser la puissance. Dans le cas où la puissance est à $100\%P_n$ avec deux TPA en marche, ces régulations permettent d'atteindre le seuil de $60\%P_n$ dont le franchissement déclenche l'arrêt d'une des deux TPA est arrêtée, les régulations pour faire baisser la puissance se poursuivant sur l'autre. Si la TPA en fonctionnement tombe en panne, la seconde est démarrée et la réduction de puissance continue. La TPA en panne est mise en réparation. Si la seconde TPA ne démarre pas, cela conduit à un AAR. Lorsque la puissance atteint $2\%P_n$, la TPA encore en marche est arrêtée. Dans le cas où la puissance est déjà à $60\%P_n$ en début de mode (avec une TPA en marche), la procédure décrite précédemment s'applique sur la seule TPA en fonctionnement.

4.3 Modélisation du cas test des turbopompes (TPA)

L'objectif est d'évaluer qualitativement et quantitativement des séquences d'événements admissibles par le sous-système des deux turbopompes (TPA) en appliquant les démarches proposées dans les chapitres 2 et 3.

Cette étude repose sur la modélisation du fonctionnement du cas test et de ses différents modes de fonctionnement. Les travaux développés dans le cadre du projet APPRODYN ont produit des modèles du sous-système des TPA sous diverses formes (processus markoviens déterministes par morceaux, réseaux de Petri stochastiques et automates stochastiques hybrides). Les modèles les plus proches de ceux utilisés par notre approche sont les automates stochastiques hybrides, ils ont donc servi de référence à notre modélisation. Néanmoins, nous avons apporté d'importantes modifications car la modélisation APPRODYN proposait un automate à états finis d'une trentaine d'états pour chaque TPA et un automate de coordination d'une centaine d'états [Babykina *et al.*, 2016]. Nous avons donc été conduits à introduire

explicitement une description modulaire des différents modes de fonctionnement, de taille réduite, caractérisant les trois phases d'exploitation : montée en puissance, conduite en puissance et baisse de puissance. Comme indiqué au chapitre 2, nous utiliserons des modèles markoviens/semi-markoviens et leur représentation déterministe sous la forme d'automates à états finis en remplaçant les taux de probabilités par des événements.

La modélisation du cas test des turbopompes fait apparaître des modes de fonctionnement nominaux pour les phases de montée en puissance (modes 1 et 2), de conduite en puissance (mode 5) et de baisse de puissance (mode 8) et des modes de fonctionnement dégradés. La figure (4.6) représente les commutations potentielles entre ces modes : les modes nominaux sont représentés par un cercle en trait plein, les modes dégradés hors-turbine par un parallélogramme, les modes dégradés partie turbine par des rectangles ; les flèches pointillées représente un choix de conduite à l'initialisation ; les modes représentés en pointillés font référence aux modes déjà représentés sur l'arbre de commutation.

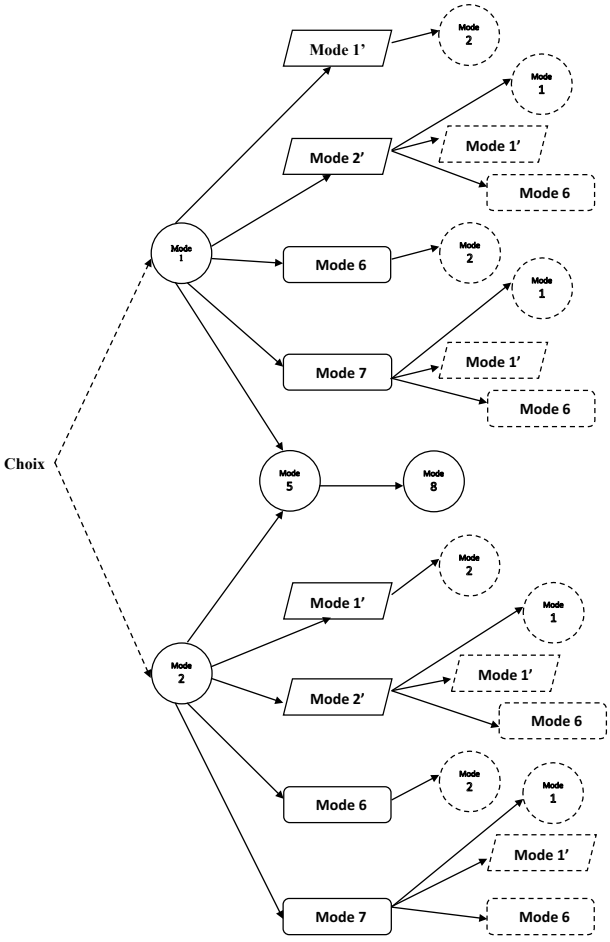


FIGURE 4.6 – Choix et commutations entre les modes considérés pour le sous-système des TPA

Notation. Les événements appartenant aux alphabets des automates à états finis présentés dans les sections suivantes ainsi que les probabilités associées sont donnés dans le tableau (4.2).

Remarque 1. Pour faciliter la lecture des modèles, les états des automates sont caractérisés par une étiquette traduisant l'état courant des TPA sous la forme de symboles : a représente un état d'attente ou d'arrêt, m un état de marche, dem une sollicitation de TPA et p un état de panne. Cette représentation est purement graphique et n'est associée à aucune représentation formelle.

Remarque 2. L'ensemble des automates décrivant les modes de fonctionnement du sous-système TPA est fourni en Annexe A.

TABLE 4.2 – Notations (événements et probabilités)

Evénement	Probabilité	Signification
soll.TPA1_HT soll.TPA2_HT	-	Sollicitation de la partie hors-turbine (déterministe)
soll.TPA1_T soll.TPA2_T	-	Sollicitation de la partie turbine (déterministe)
P=60%	-	Puissance atteinte égale à 60%P_n (déterministe)
réponse_soll.TPA1_HT	p_{ssHT} p_{esHT}	Succès à la sollicitation de la TPA1 partie hors-turbine (HT) Echec à la sollicitation de la TPA1 partie hors-turbine (HT)
réponse_soll.TPA2_HT	p_{ssHT} p_{esHT}	Succès à la sollicitation de la TPA2 partie hors-turbine (HT) Echec à la sollicitation de la TPA2 partie hors-turbine (HT)
réponse_soll.TPA1_T	p_{ssT} p_{esT}	Succès à la sollicitation de la TPA1 partie turbine (T) Echec à la sollicitation de la TPA1 partie turbine (T)
réponse_soll.TPA2_T	p_{ssT} p_{esT}	Succès à la sollicitation de la TPA2 partie hors-turbine (T) Echec à la sollicitation de la TPA2 partie hors-turbine (T)
defTPA1_HT defTPA2_HT	p_{fHT}	Défaillance de la partie hors-turbine (HT)
defTPA1_T defTPA2_T	p_{fT}	Défaillance de la partie turbine (T)
repTPA1_HT repTPA2_HT	p_{rHT}	Réparation de la partie hors-turbine (HT)
repTPA1_T repTPA2_T	p_{rT}	Réparation de la partie turbine (T)
réponse_rep	p_{srHT} p_{srT} p_{erHT} p_{erT}	Succès à la réparation de la partie hors-turbine (HT) Succès à la réparation de la partie turbine (T) Echec à la réparation de la partie hors-turbine (HT) Echec à la réparation de la partie turbine (T)
réponse_reg.	p_{sreg} p_{ereg}	Succès à la régulation de niveau de 100%P_n Echec à la régulation de niveau de 100%P_n à 60%P_n

4.3.1 Conduite en démarrage et montée en puissance

Considérant qu'à l'instant initial les deux TPA sont en attente, le fonctionnement de la phase de conduite en démarrage et montée en puissance sera réalisée selon six modes de fonctionnement :

- deux modes caractérisant la procédure nominale de démarrage pour atteindre $P = 100\%P_n$ en démarrant par la première TPA (Mode 1), soit par la seconde (Mode 2),
- deux modes correspondant aux actions réalisées suite à la défaillance et/ou à un échec à la sollicitation d'une TPA partie hors-turbine (Mode 1′ pour TPA1-HT, Mode 2′ pour TPA2-HT),
- deux modes correspondant aux actions réalisées suite à la défaillance et/ou à un échec à la sollicitation d'une TPA partie turbine (Mode 6 pour la TPA1-T, Mode 7 pour la TPA2-T).

Le premier mode (Mode 1) correspond à une montée en puissance débutant par la turbopompe TPA 1 et est représenté par l'automate A_1 à 8 états et 12 transitions de la figure (4.7). Il suit la procédure suivante : démarrage TPA 1 partie hors-turbine, démarrage TPA 1 partie turbine, démarrage TPA 2 partie hors-turbine, démarrage TPA 2 partie turbine. L'état initial du Mode 1 est l'état 1 où l'action de la partie hors-turbine du TPA 1 a été sollicitée et où la TPA 2 est en attente.

En cas d'échec au démarrage d'une TPA (partie hors-turbine ou partie turbine), le système va commuter vers un mode dégradé spécifique pour le composant en échec (le système évoluera vers l'état 1 du mode 1′ en cas d'échec à la sollicitation de la TPA 1 partie hors-turbine, vers l'état 1 du mode 6 en cas d'échec à la sollicitation de la TPA 1 partie turbine, vers l'état 7 du mode 2′ en cas d'échec à la sollicitation de la TPA 2 partie hors-turbine et vers l'état 7 du mode 7 en cas d'échec à la sollicitation de la TPA 2 partie turbine). Ces événements de commutation sont les événements de commutation décrits dans les définitions de l'opérateur de concaténation du chapitre 3. Comme indiqué dans ces définitions, ces événements ont une probabilité p de provoquer une commutation vers un autre mode (flèche pointillée dans la figure (4.7) correspondant à un échec à la sollicitation) et une probabilité $1 - p$ de rester dans le mode actif (flèche pleine correspondant à un succès à la sollicitation).

En cas de succès à toutes les sollicitations, le système commute depuis l'état 8 de l'automate A_1 vers l'état initial du mode 5 décrivant le fonctionnement en puissance.

Une représentation compacte de l'automate A_1 est donnée sous la forme du p-automate de la figure (4.8) où les événements sont remplacés par leur probabilité d'occurrence.

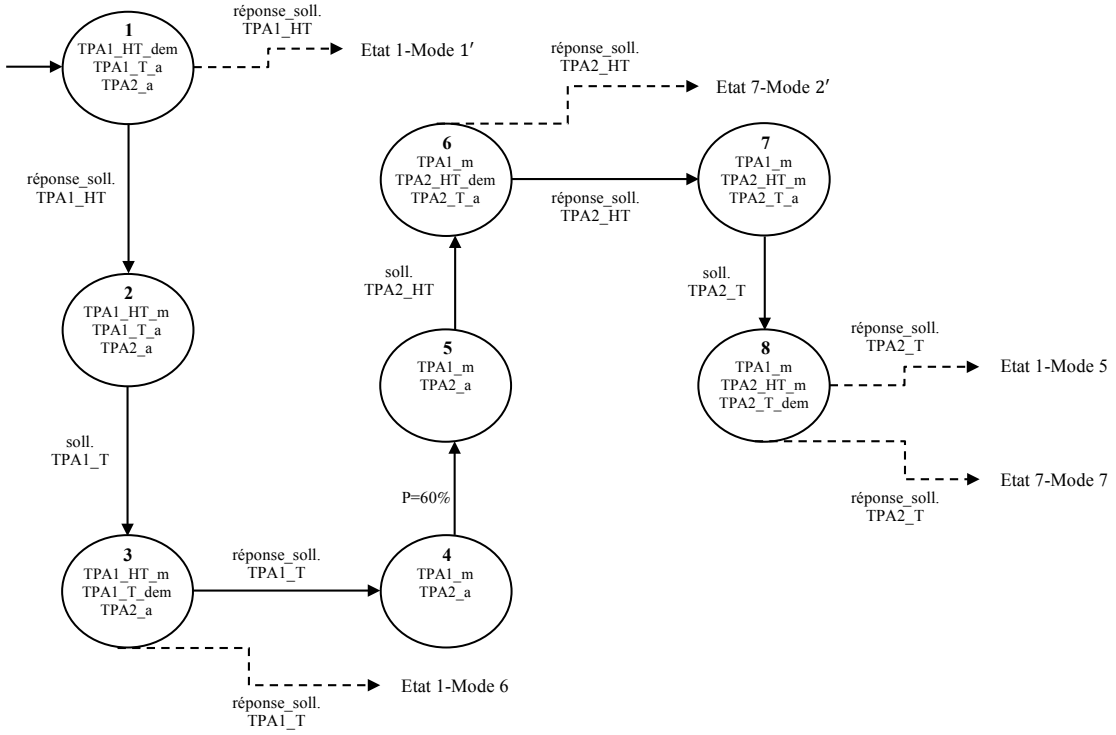


FIGURE 4.7 – Automate A_1 du Mode 1 de conduite en démarrage et montée en puissance

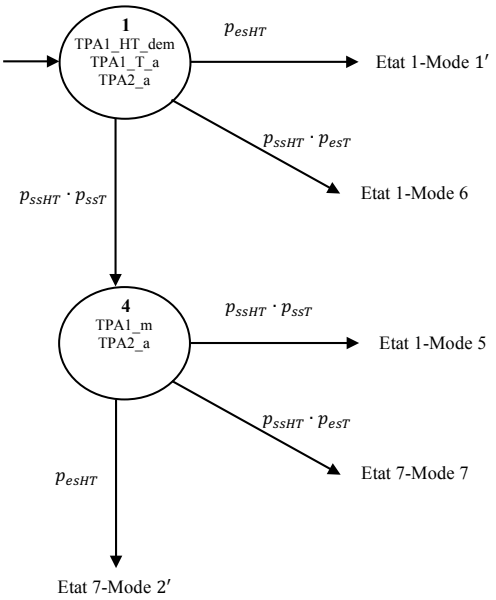


FIGURE 4.8 – Version compacte du p-automate associé à l'automate A_1

Le mode 2 est similaire au mode 1, si ce n'est que la procédure débute par la TPA 2 avant de solliciter la TPA 1. L'initialisation dans un de ces deux modes sera réalisée par l'opérateur de choix que nous décrirons ultérieurement.

A partir de deux modes nominaux 1 et 2, le système peut évoluer vers les modes dégradés $1'$, $2'$, 6 et 7. La structure des modèles de ces quatre modes est identiques, ils ne diffèrent que par les événements correspondant aux demandes de sollicitation (la TPA 1 est défaillante dans le mode 1, les modes $1'$ et 6 demanderons donc le démarrage de la TPA 2 alors que ce sera l'inverse pour les modes $2'$ et 7) et aux demandes de réparation effectuée après démarrage d'une TPA (le mode $1'$ sollicitera la réparation de la partie HT de la TPA 1 alors que le mode 6 sollicitera la répartie de la partie T). Compte tenu de ces similarités, nous ne décrirons dans ce chapitre que le mode $1'$, les modèles des autres modes sont fournis en Annexe A.

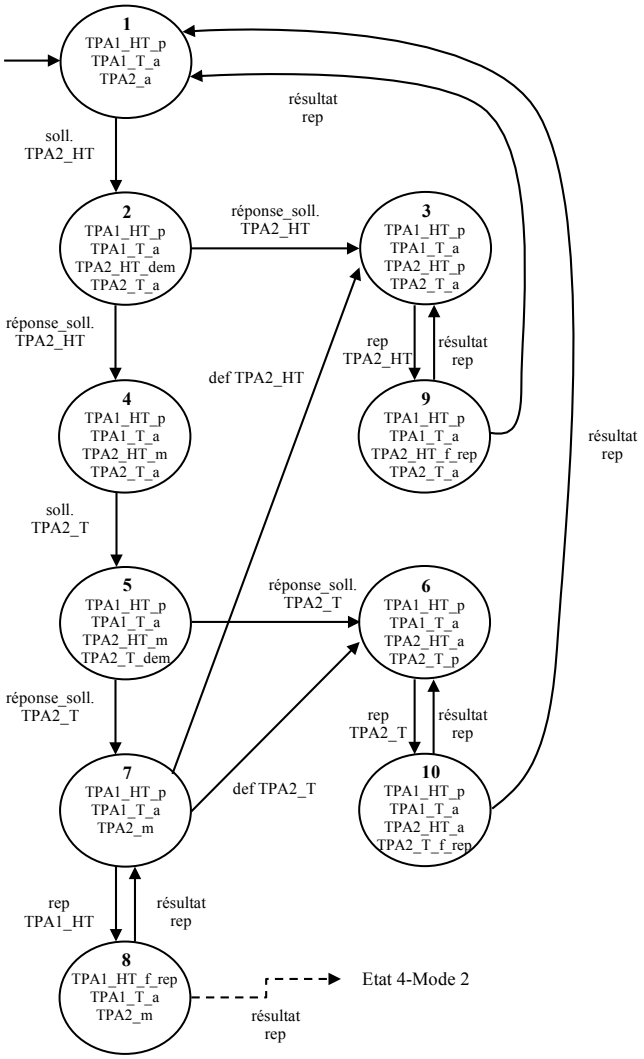


FIGURE 4.9 – Automate $A_{1'}$ du Mode $1'$ de conduite en démarrage et montée en puissance

Le modèle semi-markovien qui représente le Mode $1'$ est donné par l'automate à états finis $A_{1'}$ dans la figure (4.9). Dans l'état initial du Mode $1'$ (l'état 1), la partie HT de TPA 1 est en panne et par conséquent l'action de TPA 2 est sollicitée, en commençant par la partie hors-turbine puis la partie turbine. En cas de succès à la sollicitation des parties HT et T de TPA 2, la partie hors-turbine de la TPA 1 (dont l'échec

à la sollicitation avait conduit au mode 1′) est mise en réparation et, en cas de succès de cette réparation, le système commute vers l'état 4 du mode 2. En cas d'échec à la sollicitation des parties HT ou T de la TPA 2, le système sollicite la réparation de la partie de la TPA 2 défaillante (état 3 pour HT ou état 6 pour T). De plus, lorsque le démarrage des deux parties HT et T de la TPA 2 ont été réalisées avec succès (état 7), le système peut subir une défaillance en fonctionnement de ces deux parties de la TPA 2 qui le ramène dans les états 3 (défaillance HT de la TPA 2) ou 6 (défaillance T de la TPA 2). Comme pour le mode 1, la figure (4.10) représente une version compacte de l'automate $A_{1'}$ sous la forme d'un p-automate dans lequel les événements ont été remplacés par leur probabilité d'occurrence.

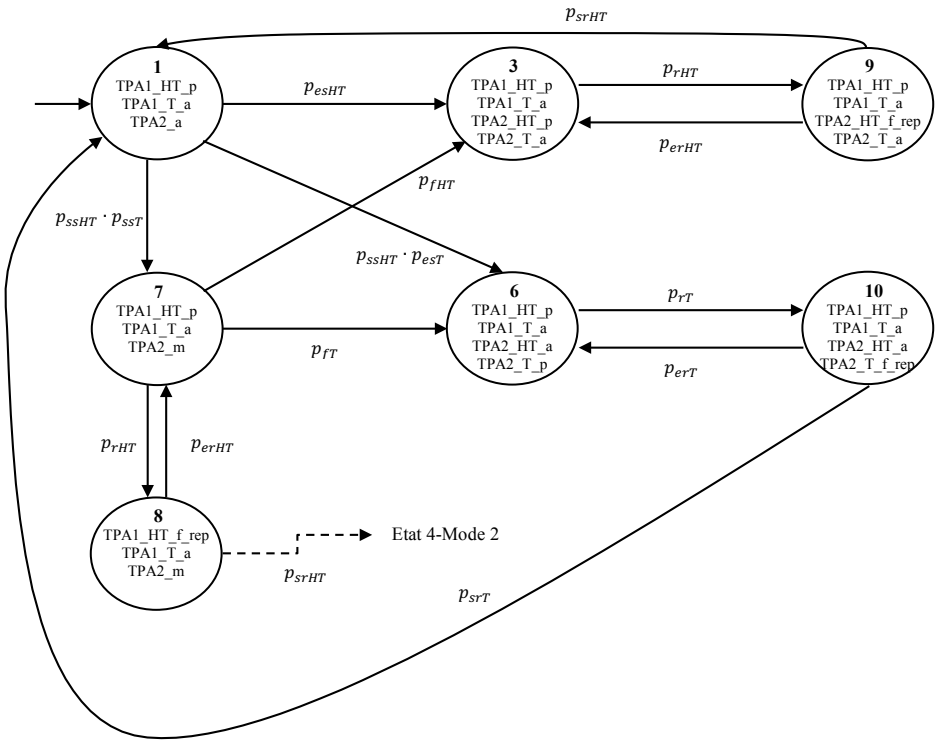


FIGURE 4.10 – Version compacte du p-automate associé à l'automate $A_{1'}$

4.3.2 Conduite en puissance

En considérant le scénario de fonctionnement décrit dans la section 4.2.2.2 correspondant au profil de mission de la figure (4.3), l'état initial de cette phase de fonctionnement est caractérisé par une puissance égale à $100\%P_n$ et les deux turbopompes TPA en marche. Pour cette phase nous avons construit un seul mode de fonctionnement (noté Mode 5 dans la figure 4.6). Le sous-système constitué par les deux TPA peut commuter vers le Mode 5 depuis l'état 8 de l'automate A_1 qui représente le Mode 1 ou depuis l'état 8 de l'automate A_2 qui représente le Mode 2. Le Mode 5 est représenté par le modèle semi-markovien donné par l'automate A_5 de la figure (4.11).

Quatre événements peuvent se produire à partir de l'état initial (état 1) : défaillances des parties HT et T de TPA 1, respectivement de TPA 2. Chacun de ces événements est caractérisé par un taux de défaillance constant caractéristique d'une loi exponentielle de distribution des probabilités. Si une de ces défaillance se produit, la turbopompe correspondante est en panne (la partie HT ou T dont la défaillance s'est produit est en panne, l'autre partie est à l'arrêt). Étant donné qu'une seule turbopompe reste en marche, la puissance doit être réduite à $60\%P_n$ par l'action des fonctions de régulation. Un échec de ces fonctions peut amener à l'Arrêt Automatique du Réacteur (état 22). Si la réduction de la

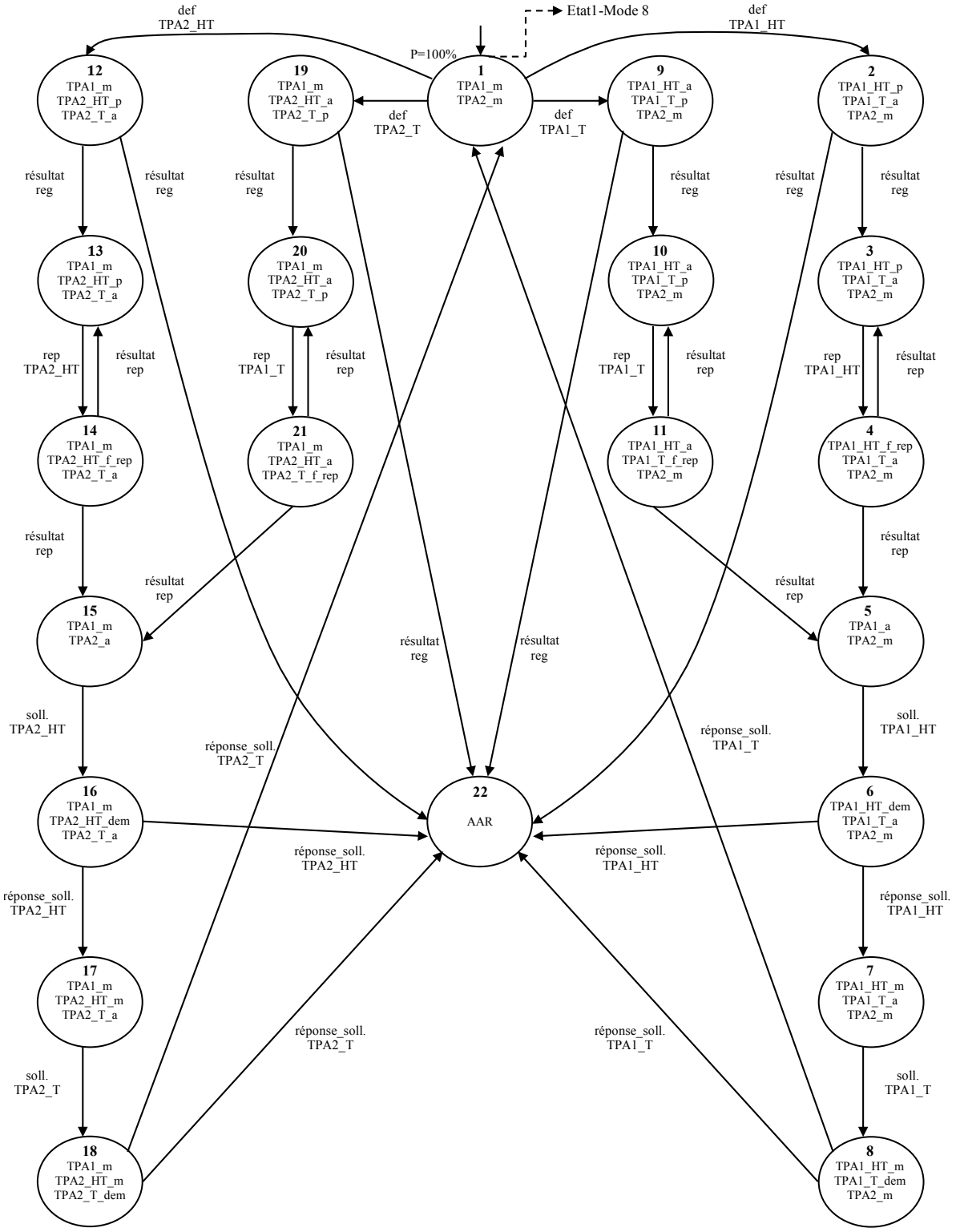


FIGURE 4.11 – L'automate A_5 qui représente le Mode 5 de conduite en puissance

puissance réussit, la partie défaillante (HT ou T) est mise en réparation. Le succès de cette réparation conduit le système vers un état où une TPA est réparée, mais à l'arrêt, et la seconde TPA en marche ; en cas d'échec de cette réparation, une nouvelle opération de réparation commence. Si la réparation s'est déroulée correctement, il est possible de remonter en charge vers $100\%P_n$ en sollicitant le démarrage de la TPA qui est à l'arrêt (d'abord la partie HT, suivi de la partie T). L'échec de la sollicitation de l'une ou l'autre de deux parties amène à l'AAR.

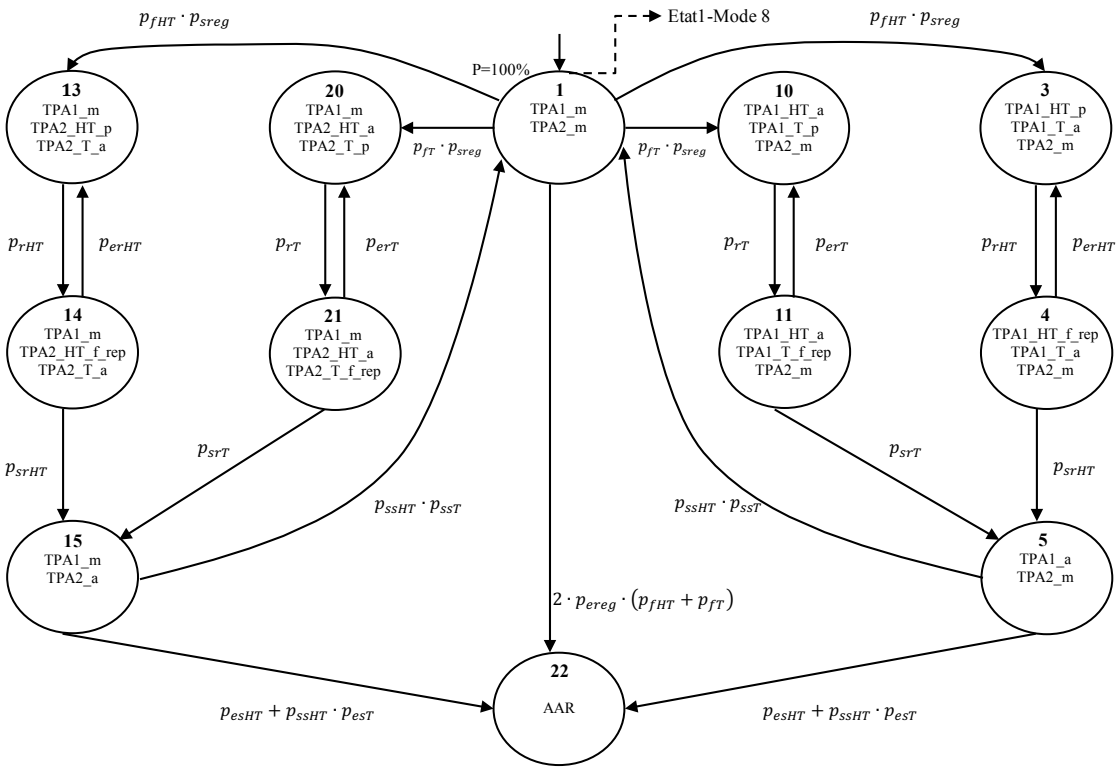


FIGURE 4.12 – Version compacte du p-automate associé à l'automate A_5

Le nombre d'états et des transitions de l'automate A_5, décrivant le comportement fonctionnel et dysfonctionnel détaillé, peut être réduit de façon analogue aux automates correspondant aux autres modes que nous avons considérés précédemment, l'objectif étant de permettre la détermination des séquences d'événements et le calcul de leurs probabilités d'occurrence. Ainsi, la représentation compacte de l'automate A_5 est donnée sous la forme d'un p-automate dans la figure (4.12) où les événements sont remplacés par leur probabilité d'occurrence.

Tel que nous remarquons dans le scénario de fonctionnement du générateur de vapeur (figure 4.3), la conduite en plein puissance peut durer jusqu'à 18 mois, cette étape étant suivie par la descente en puissance qui se réalise en 24 heures. Le passage entre la phase de conduite en puissance et celle en baisse de puissance est représenté par la commutation du sous-système entre le Mode 5 (depuis son état 1) et le Mode 8 (qui sera décrit dans la section suivante).

4.3.3 Conduite en baisse de puissance

En considérant le scénario de fonctionnement décrit dans la section 4.2.2.3 dont la puissance est égale à $100\%P_n$ avec les deux TPA en marche lorsque la phase de baisse de puissance démarre, nous avons considéré un seul mode de fonctionnement pour cette phase (noté Mode 8 dans la figure 4.6). Nous avons construit l'automate à états finis A_8 de la figure (4.13) représentant le modèle semi-markovien décrivant le comportement du sous-système des turbopompes TPA pendant la baisse en puissance du réacteur.

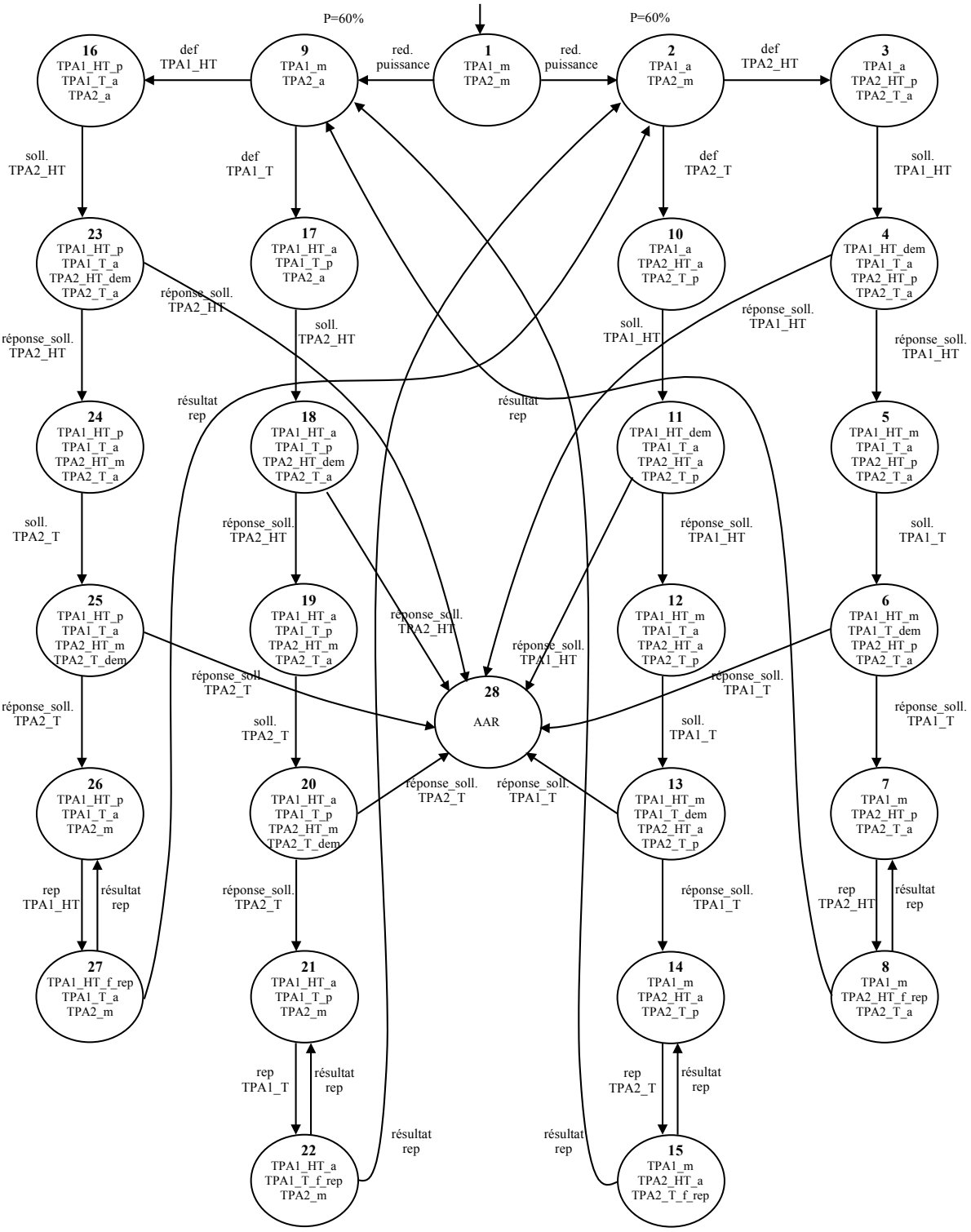


FIGURE 4.13 – L'automate A_8 correspondant au Mode 8 de conduite en baisse de puissance

Dans l'état initial (état 1) du mode 8 les deux TPA sont en marche et la réduction de la puissance se fait par l'action des fonctions de régulation. Lorsque la puissance atteint la valeur de $60\%P_n$, une de deux TPA est arrêtée. Le choix entre les deux TPA est réalisé en se basant sur une probabilité (connue) qui indique quelle est la première TPA qui sera arrêtée. La réduction de la puissance du réacteur est faite par la suite en utilisant une seule TPA. Deux événements peuvent se produire : la défaillance de la partie HT ou la défaillance de la partie T de TPA utilisée. Si une de ces défaillance se produit, la partie défaillante de la TPA utilisée se trouvera en panne et l'autre partie sera mis à l'arrêt, tandis que la seconde TPA qui se trouvait à l'arrêt est sollicitée pour démarrer (d'abord la partie HT, suivi de la partie T). Le succès de ces sollicitations entraîne la continuation du processus de baisse en puissance et la partie défaillante de la pompe en panne est mise en réparation. Autrement, l'échec de la sollicitation de l'une ou l'autre de deux parties amène à l'Arrêt Automatique du Réacteur (état 28). Si la réparation de la partie défaillante de la pompe en panne réussit, cette TPA restera à l'arrêt, mais fonctionnelle, elle pouvant à nouveau être sollicitée à démarrer dans le cas où la TPA en cours d'utilisation deviendra défaillante. En cas d'échec de cette réparation, une nouvelle opération de réparation recommence.

Étant donné le nombre assez grand d'états et des transitions de l'automate A_8, décrivant le comportement fonctionnel et dysfonctionnel détaillé, nous avons construit une représentation compacte donnée sous la forme d'un p-automate dans la figure (4.14) où les événements sont remplacés par leur probabilité d'occurrence. Ce p-automate sera utilisé pour déterminer les séquences d'événements et évaluer leur probabilité d'occurrence.

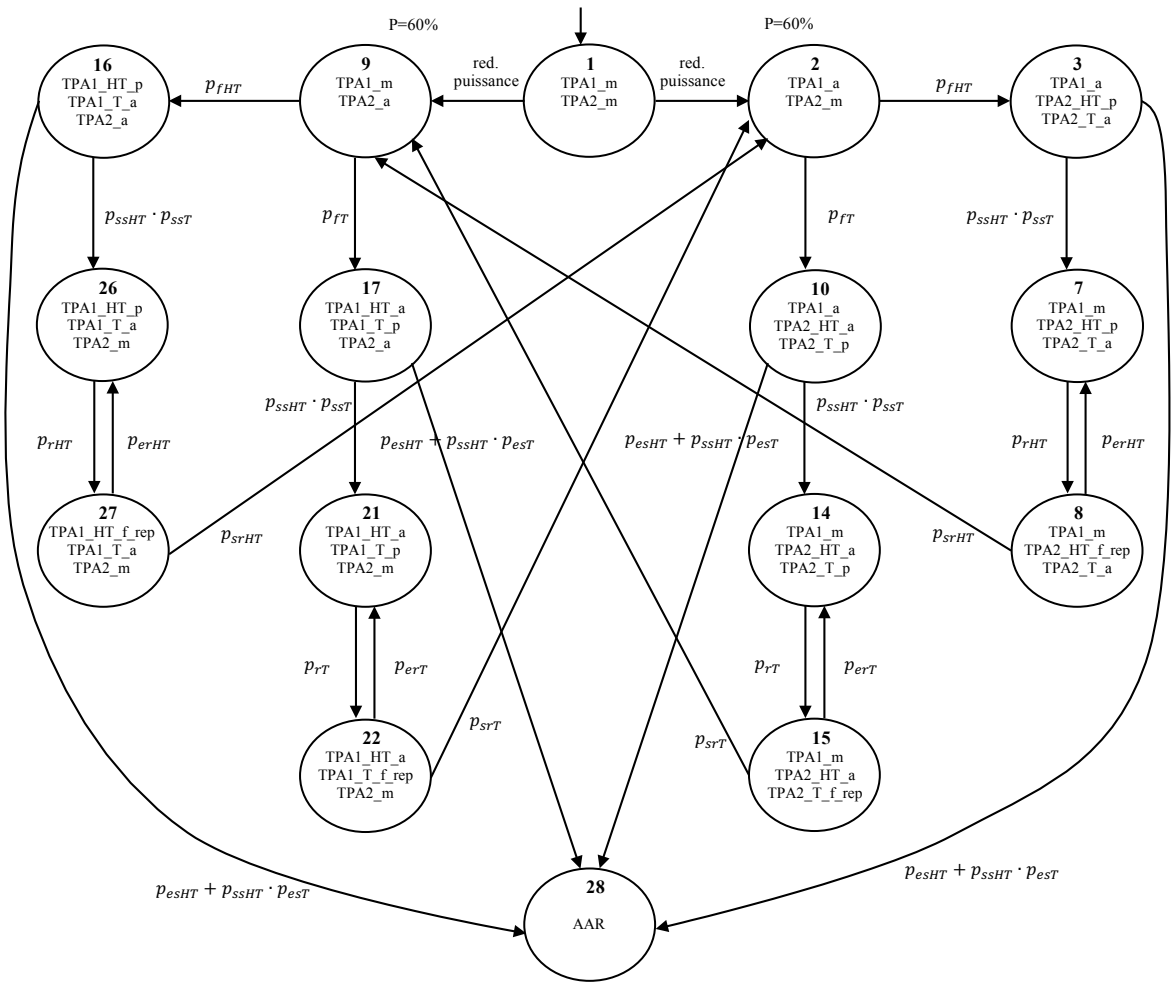


FIGURE 4.14 – Version compacte du p-automate associé à l'automate A_8

L'objectif par la suite est de réaliser une évaluation quantitative (probabiliste) des séquences d'événements appartenant au sous-système de deux turbopompes, en régime transitoire et asymptotique. Cette évaluation sera réalisée en tenant compte des différentes modes du sous-système et de leurs p-automates correspondants. Pour les calculs numériques, nous avons retenues les données de fiabilité présentées en section 4.2.1.2.

4.4 Évaluation des séquences d'événements en régime transitoire

4.4.1 Évaluation locale mode par mode

Cette section illustre la méthode de calcul des probabilités d'occurrence des séquences d'événements en régime transitoire, décrite dans la section 2.6.2 du chapitre II et basée sur la détermination de la transformée de Laplace du temps de premier passage entre un état initial et un état cible.

Nous avons choisi de présenter les résultats numériques obtenus pour quelques séquences appartenant à deux modes relatifs aux phases transitoires de montée en puissance (mode $1'$) et de baisse de puissance (mode 8). Ces résultats sont complétés par l'évaluation de séquence appartenant au mode 5 (phase de conduite en puissance) correspondant à un régime de fonctionnement stabilisé afin de montrer, ultérieurement dans ce chapitre, la convergence entre ces résultats avec ceux obtenus en régime asymptotique. Les calculs ont été effectués sur la base des p-automates correspondant aux modèles semi-markoviens de ces modes.

4.4.1.1 Mode $1'$

Les calculs présentés ci-dessous reposent sur le p-automate établi en section 4.3.1. Nous considérerons deux états cible, l'état 1 qui est aussi l'état initial et l'état 8 depuis lequel le sous-système commute vers l'état 4 du Mode 2. Pour chaque état cible considéré, nous donnons ici les résultats obtenus sur une période de 24 heures correspondant à la durée nominale de la phase de démarrage et montée en puissance.

En considérant d'abord l'état 1 (où la TPA 1 est en panne et la TPA 2 est en attente) comme cible ($\overrightarrow{j} = \{1\}$), l'équation matricielle (2.19) correspondante est la suivante :

$$\begin{pmatrix} 1 & -r^*_{13}(s) & -r^*_{16}(s) & -r^*_{17}(s) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -r^*_{39}(s) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -r^*_{6(10)}(s) \\ 0 & -r^*_{73}(s) & -r^*_{76}(s) & 1 & -r^*_{78}(s) & 0 & 0 \\ 0 & 0 & 0 & -r^*_{87}(s) & 1 & 0 & 0 \\ 0 & -r^*_{93}(s) & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -r^*_{(10)6}(s) & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L_{11}(s) \\ L_{31}(s) \\ L_{61}(s) \\ L_{71}(s) \\ L_{81}(s) \\ L_{91}(s) \\ L_{(10)1}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ r^*_{91}(s) \\ r^*_{(10)1}(s) \end{pmatrix}$$

Nous pouvons remarquer que dans la première colonne de la matrice qui contient les transformées Laplace-Stieltjes $r^*_{ij}(s)$, tous les éléments (sauf le premier qui correspond à $-r^*_{11}(s)$) ont la valeur 0. Cela s'explique par le fait que, en fonction de l'état cible choisi, les éléments de la colonne qui lui correspond dans cette matrice seront égaux à 0 sauf l'élément $-r^*_{ii}(s)$ qui aura toujours la valeur 1.

Les solutions de ce système sont les transformées de Laplace du temps de premier passage depuis chacun des 7 états de ce mode vers l'état 1. Considérons le temps du premier passage depuis l'état 1 vers lui même donné par la transformée de Laplace $L_{11}(s)$. En inversant de manière numérique $L_{11}(s)$, nous obtenons la densité de probabilité $f_{11}(t)$ représentée sur la figure (4.15).

On en déduit la valeur numérique de la fonction de répartition $F_{11}(t)$ qui représente la probabilité d'occurrence de la première séquence d'événements de l'état 1 vers lui même :

$$F_{11}(t) = 0.0041, \ \textit{pour } t = 24 \textit{ heures}$$

Du point de vue graphique $F_{11}(t)$, représente l'aire entre l'axe du temps et la courbe donnant l'évolution de la densité de probabilité $f_{11}(t)$.

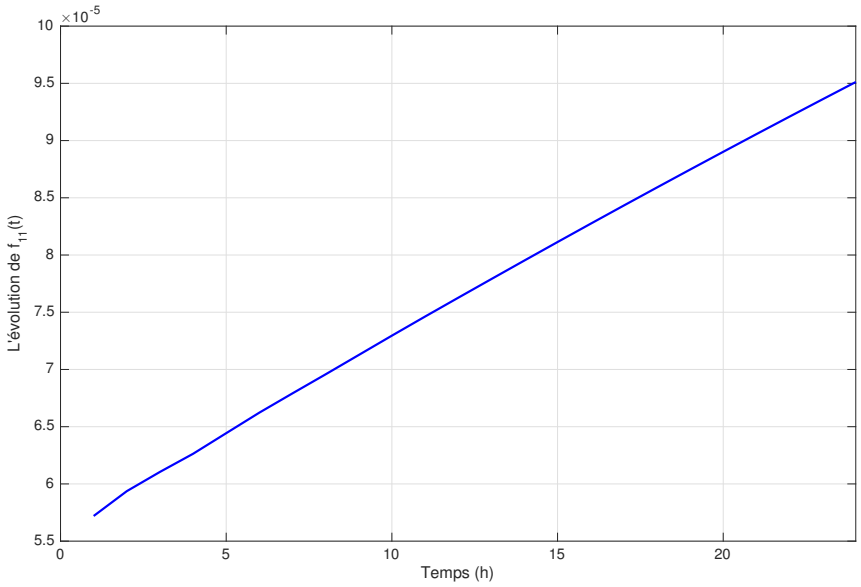


FIGURE 4.15 – Densité de probabilité du temps de premier passage de l'état 1 vers lui même (Mode 1′)

Le deuxième état cible considéré est l'état 8, ($\overrightarrow{j} = \{8\}$), où la partie HT de la TPA 1 a été réparée et la TPA 2 est en marche. L'équation matricielle (2.19) sera réécrite en fonction de cet nouvel état cible :

$$\begin{pmatrix} 1 & -r^*_{13}(s) & -r^*_{16}(s) & -r^*_{17}(s) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -r^*_{39}(s) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -r^*_{6(10)}(s) \\ 0 & -r^*_{73}(s) & -r^*_{76}(s) & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -r^*_{87}(s) & 1 & 0 & 0 \\ -r^*_{91}(s) & -r^*_{93}(s) & 0 & 0 & 0 & 1 & 0 \\ -r^*_{(10)1}(s) & 0 & -r^*_{(10)6}(s) & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L_{18}(s) \\ L_{38}(s) \\ L_{68}(s) \\ L_{78}(s) \\ L_{88}(s) \\ L_{98}(s) \\ L_{(10)8}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ r^*_{78}(s) \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Dans ce cas, la colonne correspondant à l'état 8 aura tous les éléments égaux à 0 excepté $-r^*_{88}(s)$ qui est égal à 1. Les solutions de ce système sont les transformées de Laplace du temps de premier passage depuis l'ensemble des états vers l'état 8. Parmi ces évolutions, nous nous intéressons à trois séquences d'événements particulières :

- la séquence nominale de l'état initial vers l'état 8,
- la séquence de l'état 3, où les parties HT de TPA 1 et de TPA 2 sont en panne, vers l'état 8,
- la séquence de l'état 6, où la partie HT de TPA 1 et la partie T de TPA 2 sont en panne, vers l'état 8.

Le temps de premier passage depuis l'état 1 vers l'état 8 est donné par la transformée de Laplace $L_{18}(s)$ qui a été inversée numériquement afin d'obtenir la densité de probabilité $f_{18}(t)$ représentée sur la figure (4.16) pour obtenir la fonction de répartition $F_{18}(t)$:

$$F_{18}(t) = 0.4244, \ \textit{pour } t = 24 \textit{ heures}$$

Cette valeur correspond à la probabilité d'occurrence de la séquence qui se produit pendant le temps de premier passage depuis l'état 1 à l'état 8.

De manière similaire, la solution obtenue pour $L_{38}(s)$ nous permet de calculer la densité de probabilité $f_{38}(t)$ (figure 4.17) puis la fonction de répartition $F_{38}(t)$:

$$F_{38}(t) = 0.1703, \ \textit{pour } t = 24 \textit{ heures}$$

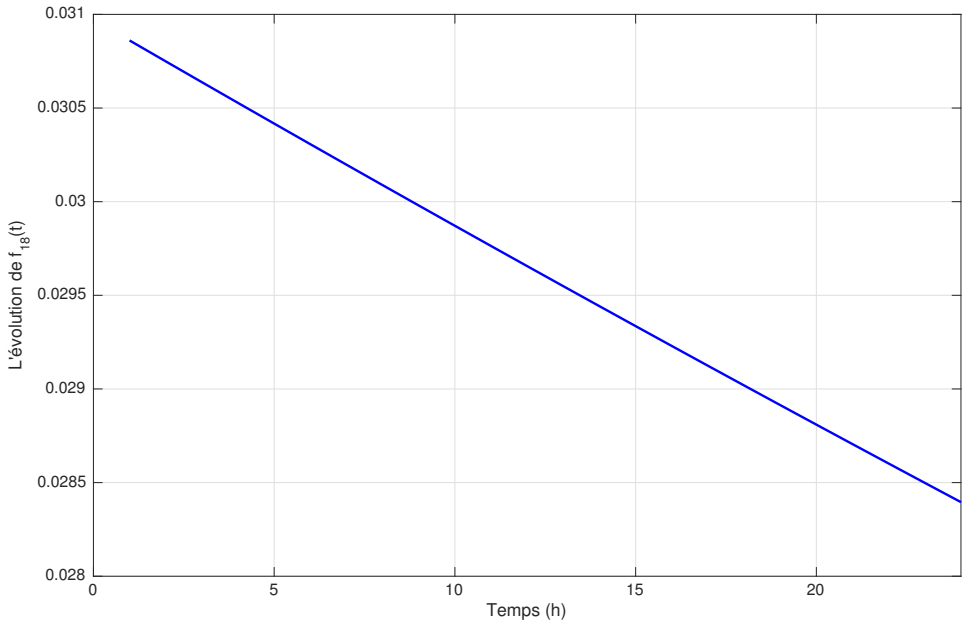


FIGURE 4.16 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 8

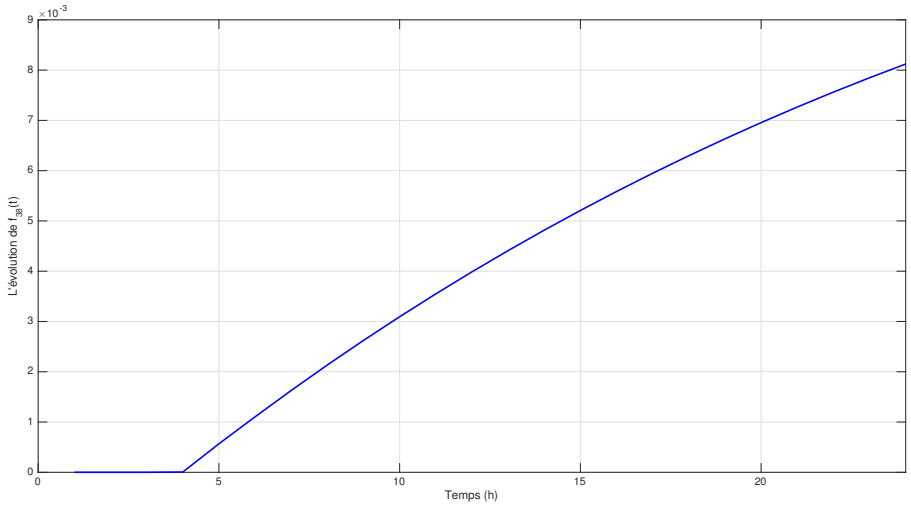


FIGURE 4.17 – Densité de probabilité du temps de premier passage de l'état 3 vers l'état 8

En dernier lieu, la densité de probabilité $f_{68}(t)$ est présentée dans la figure (4.18) et permet d'obtenir la fonction de répartition $F_{68}(t)$:

$$F_{68}(t) = 0.1703, \ \textit{pour } t = 24 \textit{ heures}$$

Nous pouvons remarquer que les probabilités d'occurrence des séquences d'événements depuis les états 1, 3 ou 6 vers l'état 8 sont largement supérieures à celle de la séquence de l'état 1 vers lui-même, ce qui traduit de fortes probabilités de sortir du mode dégradé vers le mode nominal de montée en puissance. Par ailleurs, nous pouvons observer que les valeurs de $F_{38}(t)$ et la valeur de $F_{68}(t)$ sont identiques (à 10^{-4}

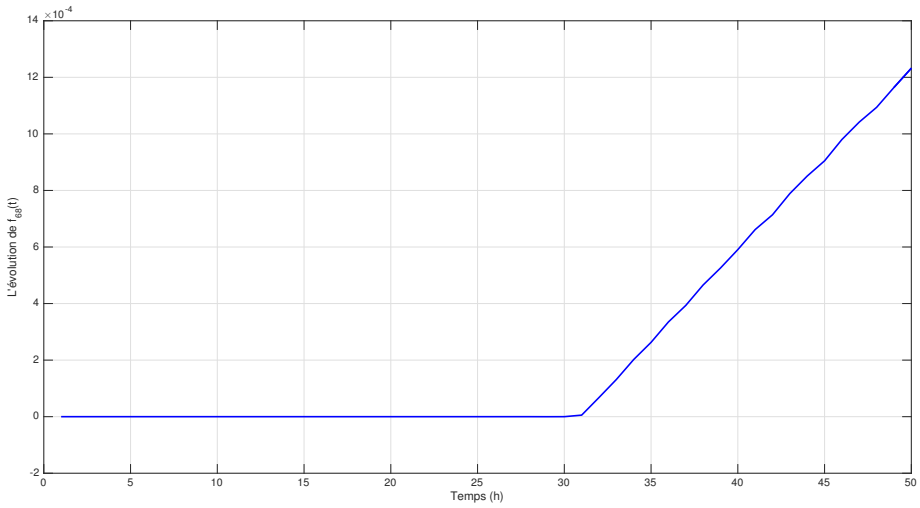


FIGURE 4.18 – Densité de probabilité du temps de premier passage de l'état 6 vers l'état 8

près) du au fait que les deux séquences d'événements depuis les états 3 et 6 vers la sortie du mode dégradé correspondent à des réparations (partie turbine et hors-turbine) ayant les mêmes caractéristiques.

4.4.1.2 Mode 5

Le Mode 5 a été conçu pour représenter la conduite en puissance du sous-système de deux TPA qui se déroule du point de vue temporel sur une période de 18 mois, l'équivalent de 13128 heures. Les calculs présentés ci-dessous reposent sur le p-automate établi en section 4.3.2. Nous considérerons trois séquences :

- la séquence de l'état 1 vers lui-même correspondant à un retour au fonctionnement à $100\%P_n$ avec les deux TPA en marche,
- la séquence de l'état 1 vers l'état 5 correspondant à une chute de puissance de $100\%P_n$ à $60\%P_n$ due à la perte d'une des deux TPA,
- la séquence de l'état 1 vers l'état 22 correspondant au déclenchement de l'Arrêt Automatique du Réacteur (AAR) dû à la perte des deux TPA.

La détermination des probabilités d'occurrence des séquences retenues est réalisée de manière identique aux calculs effectués pour le Mode $1'$.

Pour le premier état cible 1, la densité de probabilité du temps de premier passage depuis cet état vers lui même $f_{11}(t)$ est donnée sur la figure (4.19) et permet d'obtenir la fonction de répartition :

$$F_{11}(t) = 0.9921, \ pour\ t = 13128\ heures$$

Pour le deuxième état cible 5, la densité de probabilité du temps de premier passage depuis l'état initial $f_{15}(t)$ est donnée sur la figure (4.20) et permet d'obtenir la fonction de répartition :

$$F_{15}(t) = 0.8135, \ pour\ t = 13128\ heures$$

Enfin, pour le dernier cible 22, la densité de probabilité du temps de premier passage depuis l'état initial $f_{1(22)}(t)$ est donnée sur la figure (4.21) et permet d'obtenir la fonction de répartition :

$$F_{1(22)}(t) = 0.5212, \ pour\ t = 13128\ heures$$

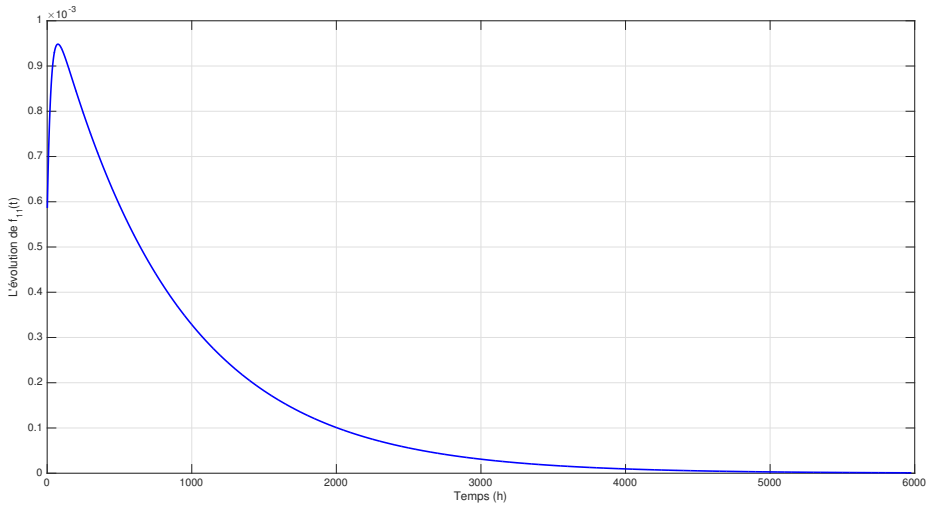


FIGURE 4.19 – Densité de probabilité du temps de premier passage de l'état 1 vers lui même (Mode 5)

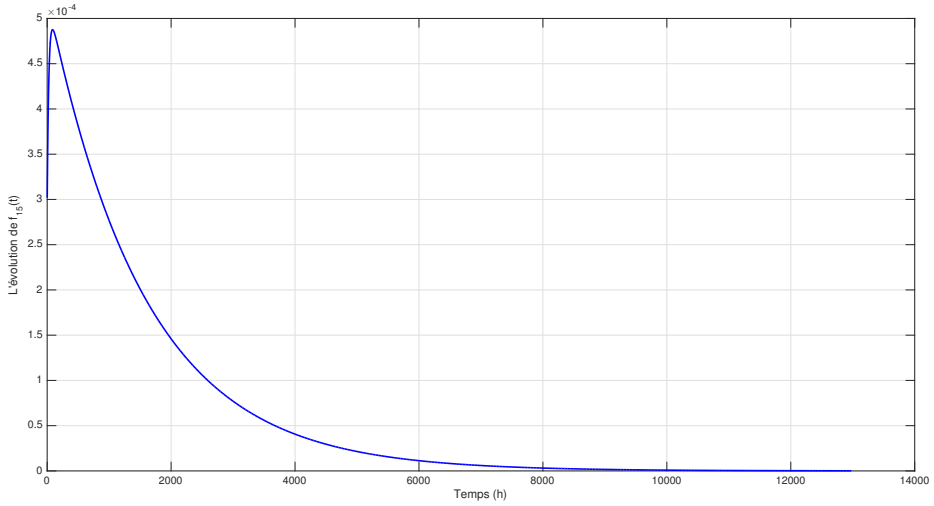


FIGURE 4.20 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 5 (Mode 5)

Les tendances observées sur des intervalles de temps relativement important (entre 1 et 2 ans selon les séquences) montrent que les probabilités décroissent en fonction de la criticité des séquences étudiées, la probabilité de revenir à $100\%P_n$ dans un laps de temps court restant évidemment très importante.

4.4.1.3 Mode 8

Le Mode 8 a été conçu pour représenter la phase de baisse de puissance qui se réalise en 24 heures. Les calculs présentés ci-dessous reposent sur le p-automate établi en section 4.3.3. Nous considérerons trois séquences :

- la séquence de l'état 1 vers l'état 2 correspondant à une procédure nominale de baisse de puissance

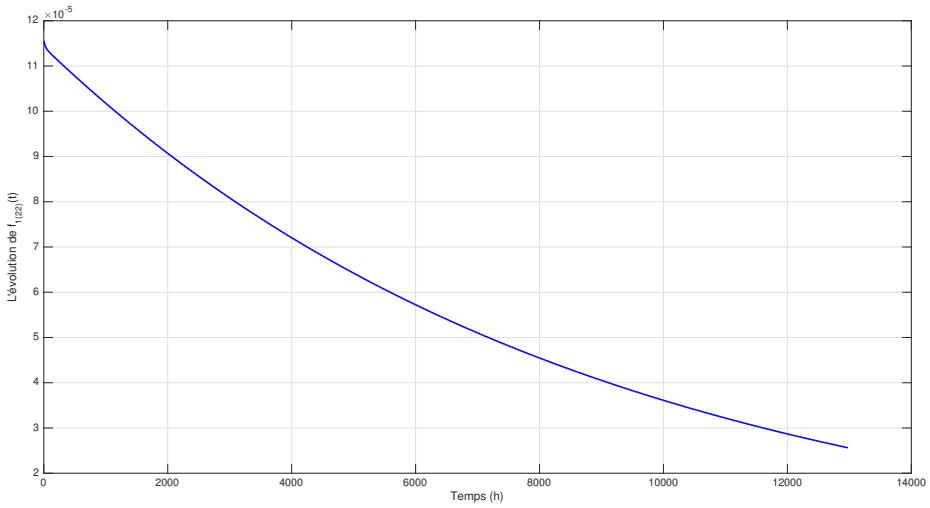


FIGURE 4.21 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 22 (Mode 5)

débutant par l'arrêt de la TPA 1,
- la séquence de l'état 1 vers l'état 9 correspondant à une procédure nominale de baisse de puissance débutant par l'arrêt de la TPA 2,
- la séquence de l'état 1 vers l'état 28 correspondant au déclenchement de l'Arrêt Automatique du Réacteur (AAR) dû à la perte des deux TPA.

Pour le premier état cible 2, la densité de probabilité du temps de premier passage depuis l'état initial (1) vers cet état $f_{12}(t)$ est donnée sur la figure (4.22) et permet d'obtenir la fonction de répartition :

$$F_{12}(t) = 0.2235, \ pour\ t = 24\ heures$$

Pour le deuxième état cible 9, la densité de probabilité du temps de premier passage depuis l'état initial (1) vers cet état $f_{19}(t)$ est la même que la densité de probabilité du temps de premier passage depuis l'état initial (1) vers l'état 2 (présentée dans la figure 4.22) car les états 2 et 9 de l'automate qui représente le mode 8 ont exactement la même signification (figure 4.14). Ainsi, la valeur obtenue pour la fonction de répartition du temps de premier passage depuis l'état initial (1) vers l'état 9 a aussi la même valeur que la fonction de répartition du temps de premier passage depuis l'état initial (1) vers l'état 2 :

$$F_{19}(t) = 0.2235, \ pour\ t = 24\ heures$$

Enfin pour l'état critique 28, la densité de probabilité du temps de premier passage depuis l'état initial (1) vers cet état $f_{1(28)}(t)$ est donnée sur la figure (4.23) et permet d'obtenir la fonction de répartition :

$$F_{1(28)}(t) = 3.5580 \cdot 10^{-6}, \ pour\ t = 24\ heures$$

Dans cette section 4.4.1, nous avons analysé séparément de manière quantitative des séquences appartenant aux modes $1'$, 5 et 8, qui correspondent aux trois types de conduite du système constitué des deux turbopompes. Néanmoins, il peut s'avérer utile d'analyser certaines séquences d'événements correspondant à des comportements ne se produisant pas exclusivement au sein d'un seul mode. Ce point fait l'objet de la section suivante.

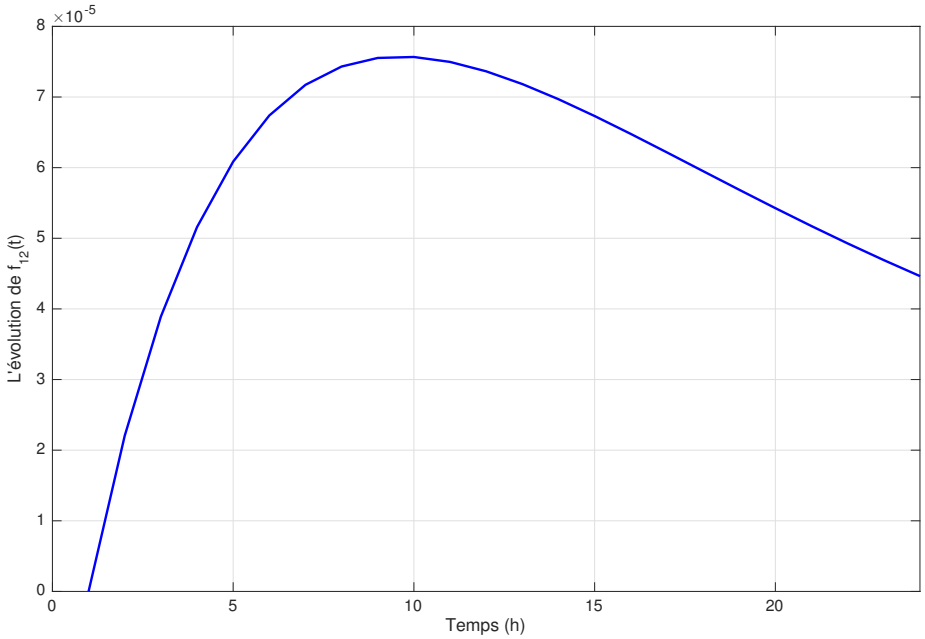


FIGURE 4.22 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 2 (Mode 8)

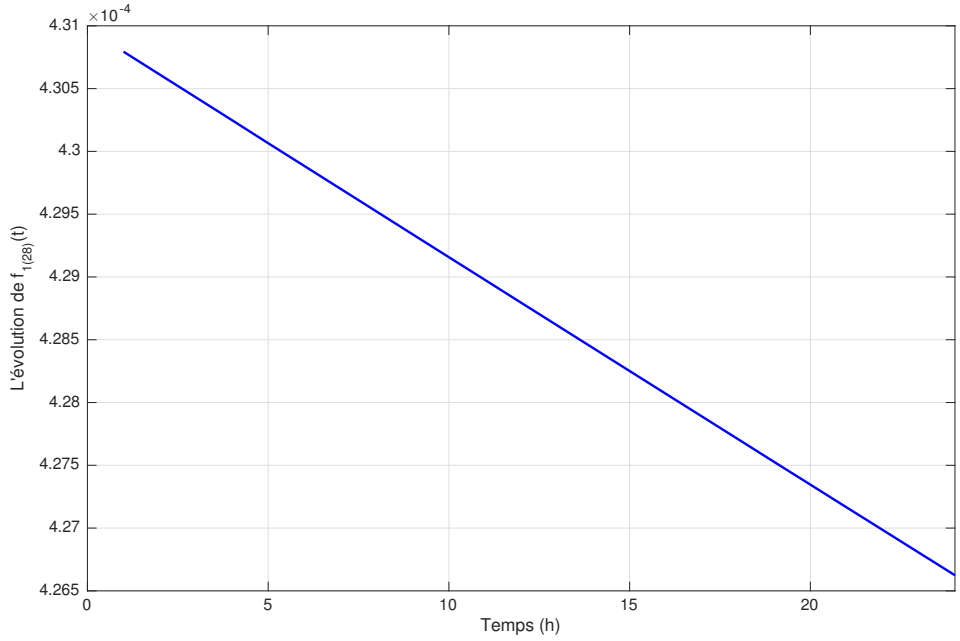


FIGURE 4.23 – Densité de probabilité du temps de premier passage de l'état 1 vers l'état 28 (Mode 8)

4.4.2 Évaluation globale par les opérateurs de composition

Cette section illustre la méthode de calcul des probabilités d'occurrence des séquences d'événements en régime transitoire, en utilisant les opérateurs de composition décrits dans le chapitre III.

4.4.2.1 Opérateur de choix

La conduite en démarrage et montée en puissance peut être réalisées selon deux alternatives équiprobables ($p_1 = 0.5$, $p_2 = 0.5$), comme indiqué sur la figure (4.6) et en section 4.3.1 :

- le Mode 1 où le démarrage débute par la mise en marche de la TPA 1 suivi de la TPA 2,
- le Mode 2 où le démarrage débute par la mise en marche de la TPA 2 suivi de la TPA 1.

En appliquant la définition généralisée donnée dans le chapitre III (définition 9), l'opérateur de choix permet de déterminer les probabilités d'occurrence des séquences d'événements en tenant compte des options de démarrage du sous-système à l'instant initial. Ainsi, les probabilités de ces séquences sont calculées en utilisant l'équation suivante :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 +_{p_2} \mathbb{L}_2}\right) = p_1 \cdot \mathbb{P}\left(s_i^{\mathbb{L}_1}\right) + p_2 \cdot \mathbb{P}\left(s_i^{\mathbb{L}_2}\right) \tag{4.1}$$

où $\mathbb{L}_1$ est le p-langage associé au Mode 1, $\mathbb{L}_2$ celui associé au Mode 2.

En considérant les p-automates correspondants au Mode 1 et au Mode 2, nous allons analyser quelques séquences d'événements en donnant leurs probabilités d'occurrence.

Pour identifier les séquences d'événements qui seront analysées, nous notons par la suite un événement $e_{i_k j_l}$ qui représente l'événemént associé à la transition entre l'état i du mode k et l'état j du mode l.

1. La première séquence considérée, $s_1 = e_{1_1 4_1} e_{4_1 1_5}$, appartient au Mode 1 où le démarrage débute par la TPA 1 :
   - L'événement noté par $e_{1_1 4_1}$ amène le sous-système depuis l'état initial à l'état 4. Il représente le succès à la sollicitation des parties HT et T de TPA 1 et est associé à une transition instantanée. Sa probabilité d'occurrence est $\mathbb{P}(e_{1_1 4_1}) = p_{ssHT} \cdot p_{ssT} = 0.9994$.
   - L'événement $e_{4_1 1_5}$ déclenche la commutation du sous-système depuis l'état 4 du Mode 1 vers l'état 1 du Mode 5 (associé à la conduite en puissance). Il a la même signification que l'événement $e_{1_1 4_1}$ (succès à la sollicitation des parties HT et T mais celles de TPA 2) et la même probabilité d'occurrence.
   - La probabilité d'occurrence de s_1 est obtenue en appliquant la définition généralisée de l'opérateur de choix donnée par l'équation 4.1 :

$$\mathbb{P}\left(s_1^{\mathbb{L}_1 +_{p_2} \mathbb{L}_2}\right) = 0.4994$$

2. La deuxième séquence $s_2 = e_{1_1 4_1} e_{4_1 7_7}$ analysée appartient aussi au Mode 1 :
   - L'événement $e_{1_1 4_1}$ a été décrit ci-dessus, suite à sa présence dans la séquence s_1.
   - $e_{4_1 7_7}$ est un événement de commutation et son occurrence détermine le passage du sous-système depuis l'état 4 du Mode 1 à l'état 7 du Mode 7. Il représente le succès à la sollicitation de la partie HT et l'échec à la sollicitation de la partie T de TPA 2 et est associé à une transition instantanée. Sa probabilité d'occurrence est $\mathbb{P}(e_{4_1 7_7}) = p_{ssHT} \cdot p_{esT} = 3.8978 \cdot 10^{-5}$.
   - La probabilité d'occurrence de s_2 est obtenue en appliquant l'équation 4.1 :

$$\mathbb{P}\left(s_2^{\mathbb{L}_1 +_{p_2} \mathbb{L}_2}\right) = 1.947 \cdot 10^{-5}$$

3. La troisième séquence analysée, $s_3 = e_{1_2 4_2} e_{4_2 7_{1'}}$, appartient au Mode 2 :
   - Le premier événement de cette séquence est identique à l'événement $e_{1_1 4_1}$ qui a été décrit dans le cadre des séquences s_1 et s_2.
   - L'événement noté par $e_{4_2 7_{1'}}$ (événement de commutation) amène le sous-système depuis l'état 4 du Mode 2 à l'état 7 du Mode $1'$ et il représente l'échec à la sollicitation de la partie HT de TPA 1. Il est associé à une transition instantanée et sa probabilité d'occurrence est $\mathbb{P}\left(e_{4_2 7_{1'}}\right) = p_{esHT} = 5.5 \cdot 10^{-4}$.

- La probabilité d'occurrence de s_3 est obtenue en appliquant l'équation 4.1 :

$$\mathbb{P}\left(s_3^{\mathbb{L}_1+_{p_2}\mathbb{L}_2}\right) = 27.483 \cdot 10^{-5}$$

4. La dernière séquence que nous présentons est $s_4 = e_{1_2 1_7}$, elle est constituée par un seul événement et elle appartient au Mode 2 :
   - L'occurrence de l'événement $e_{1_2 1_7}$ a comme conséquence la commutation du sous-système depuis l'état 1 du Mode 2 à l'état 1 du Mode 7. Il représente le succès à la sollicitation de la partie HT et l'échec à la sollicitation de la partie T de TPA 2. Sa probabilité d'occurrence est $\mathbb{P}(e_{1_2 1_7}) = p_{ssHT} \cdot p_{esT} = 3.8978 \cdot 10^{-5}$.
   - La probabilité d'occurrence de s_4 est obtenue en appliquant l'équation 4.1 :

$$\mathbb{P}\left(s_4^{\mathbb{L}_1+_{p_2}\mathbb{L}_2}\right) = 1.948 \cdot 10^{-5}$$

Si on analyse les résultats numériques obtenus pour les probabilités d'occurrence de ces quatre séquences, on observe que la probabilité de s_1 est significativement supérieure aux valeurs des trois autres séquences. Ceci s'explique par le fait qu'au contraire des autres séquences, qui conduisent vers des états où au moins une de deux TPA est en panne, cette séquence conduit à l'état 1 du Mode 5 où les deux TPA sont en marche. Autrement dit, la probabilité de se trouver dans des états défaillants est nettement inférieure à la probabilité de se trouver dans des états fonctionnels.

4.4.2.2 Opérateur de concaténation

L'opérateur de concaténation est utilisé pour évaluer des séquences d'événements comportant des événements de commutation entre modes et parcourant plusieurs modes ou phases de conduite (montée en puissance, conduite en puissance, baisse de puissance).

Nous rappelons ici que par rapport à la définition (10) donné dans le chapitre III, la probabilité d'occurrence d'une séquence qui contient des événements appartenant à m modes différentes est donnée par l'équation (3.27) :

$$\mathbb{P}\left(s_i^{\mathbb{L}_1 \cdot_{p_1} \mathbb{L}_2 \cdot_{p_2} \cdots \cdot_{p_{m-1}} \mathbb{L}_m}\right) =$$

$$\begin{cases} \mathbb{P}\left(s_i^{\mathbb{L}_1}\right), \text{ si } s_i \in A_1 \\ p_1 \sum_{t_1<t_2} \mathbb{P}(\langle t_1 e_\Delta \rangle^{\mathbb{L}_1}) p_2 \sum_{t_2<t_3} \mathbb{P}(\langle t_2 e_\Delta \rangle^{\mathbb{L}_2}) ... p_{m-1} \sum_{t_{m-1}<s_i} \mathbb{P}(\langle t_{m-1} e_\Delta \rangle^{\mathbb{L}_{m-1}}) \mathbb{P}(\langle t_{m-1}^{-1} s_i \rangle^{\mathbb{L}_m}), \text{sinon} \end{cases}$$

Rappelons que cette équation décrit deux situations : dans la première, la séquence s_i appartient exclusivement au p-automate A_1 alors que, pour la seconde, la séquence débute dans le premier mode puis commute successivement dans les modes décrits par les p-langages $\mathbb{L}_2$ à $\mathbb{L}_m$. Les événements de commutation sont les événements de réponse à une sollicitation (au démarrage, à la réparation, à la régulation, ...). Après occurrence de ces événements, la probabilité de commutation correspond à la probabilité d'échec (ou de succès) associée à la sollicitation considérée.

Pour illustrer la mise en œuvre de l'opérateur de concaténation, une dizaine de séquences contenant des commutations ont été évaluées. Les résultats sont présentés dans le tableau (4.3). Nous détaillons ci-dessous l'obtention de deux séquences critiques amenant le système dans un état de déclenchement de l'Arrêt et Automatique de Réacteur (AAR).

1. La première séquence présentée, $s_1 = e_{1_{1'} 7_{1'}} e_{7_{1'} 8_{1'}} e_{8_{1'} 4_2} e_{4_2 1_5} e_{1_5 22_5}$, amène le système depuis l'état 1 du Mode $1'$ à l'état critique 22 du Mode 5 :
   - L'événement $e_{1_{1'} 7_{1'}}$ amène le système depuis l'état initial vers l'état 7 du Mode $1'$; il représente le succès à la sollicitation des parties HT et T de TPA 2.
   - L'événement $e_{7_{1'} 8_{1'}}$ conduit le système depuis l'état 7 à l'état 8 représentant la réparation de la partie HT de TPA 1.

- L'événement $e_{8_{1'}4_2}$ conduit le système entre l'état 8 du Mode 1′ et l'état 4 du Mode 2 ; il représente le succès à la réparation de la partie HT de TPA 1 avec une probabilité $p_{1'}$.
- En cas de succès à la sollicitation des parties HT et T de la TPA 2, l'occurrence de l'événement $e_{4_21_5}$ provoque la commutation, avec une probabilité p_2, depuis l'état 4 du Mode 2 vers l'état initial du Mode 5 (en cas d'échec, le système reste dans le Mode 2).
- Le dernier événement de la séquence $e_{1_522_5}$ amène le système de l'état 1 vers l'état critique 22 du Mode 5. La transition entre ces deux états représente la combinaison entre la perte d'une turbopompe (défaillance HT ou T d'une des deux TPA) et l'impossibilité de baisser la puissance à $60\%P_n$ par les fonctions de régulation (échec à la régulation).
- La probabilité d'occurrence de s_1 est donnée par la relation suivante :

$$\mathbb{P}\left(s_1^{\mathbb{L}_{1'}\cdot p_{1'}\mathbb{L}_2\cdot p_2\mathbb{L}_5}\right) =$$

$$p_{1'}\mathbb{P}(\langle e_{1_{1'}7_{1'}}e_{7_{1'}8_{1'}}e_{8_{1'}4_2}\rangle^{\mathbb{L}_{1'}}) \cdot p_2\mathbb{P}(\langle e_{4_21_5}\rangle^{\mathbb{L}_2}) \cdot \mathbb{P}(\langle e_{1_522_5}\rangle^{\mathbb{L}_5}) = 0.2330$$

avec $e_{1_{1'}7_{1'}}e_{7_{1'}8_{1'}}e_{8_{1'}4_2}$ un préfixe de l'automate $A_{1'}$, $e_{4_21_5}$ un événement de l'automate A_2 et $e_{1_522_5}$ un événement de l'automate A_5.

2. La deuxième séquence présentée, $s_2 = e_{1_63_6}e_{3_69_6}e_{9_61_6}e_{1_67_6}e_{7_68_6}e_{8_64_2}e_{4_21_5}e_{1_522_5}$, amène le système de l'état 1 du Mode 6 jusqu'à l'état critique 22 du Mode 5 :
   - Le premier événement $e_{1_63_6}$ amène le système depuis l'état initial à l'état 3 du Mode 6 ; il représente l'échec à la sollicitation de la partie HT de la TPA 2.
   - L'occurrence de l'événement $e_{3_69_6}$ amène le système dans l'état 9 du Mode 6 et représente la réparation de la partie HT de la TPA 2.
   - L'événement $e_{9_61_6}$ ramène le système à l'état initial du Mode 6 suite à un succès de la réparation de la partie HT de la TPA 2.
   - L'événement $e_{1_67_6}$ provoque la transition depuis l'état 1 vers l'état 7 du Mode 6 suite au succès à la sollicitation au démarrage de la TPA 2.
   - L'événement $e_{7_68_6}$ conduit le système de l'état 7 à l'état 8 du Mode 6 suite à une réparation de la partie T de la TPA 1.
   - En cas de succès de la réparation précédente, l'occurrence de l'événement $e_{8_64_2}$ provoque la commutation, avec une probabilité p_6, depuis l'état 8 du Mode 6 vers l'état 4 du Mode 2 (en cas d'échec, le système reste dans le Mode 6).
   - En cas de succès à la sollicitation des parties HT et T de la TPA 1, l'occurrence de l'événement $e_{4_21_5}$ provoque la commutation, avec une probabilité p_2, depuis l'état 4 du Mode 2 vers l'état initial du Mode 5 (en cas d'échec, le système reste dans le Mode 2).
   - Le dernier événement de cette séquence, $e_{1_522_5}$, conduit à l'état de panne 22 de Mode 5 sur une combinaison de la défaillance de la partie T de TPA 1 et l'échec à la régulation de niveau.
   - La probabilité d'occurrence de la séquence s_3 est :

$$\mathbb{P}\left(s_2^{\mathbb{L}_6\cdot p_6\mathbb{L}_2\cdot p_2\mathbb{L}_5}\right) =$$

$$p_6\mathbb{P}(\langle e_{1_63_6}e_{3_69_6}e_{9_61_6}e_{1_67_6}e_{7_68_6}e_{8_64_2}\rangle^{\mathbb{L}_6}) \cdot p_2\mathbb{P}(\langle e_{4_21_5}\rangle^{\mathbb{L}_2}) \cdot \mathbb{P}(\langle e_{1_522_5}\rangle^{\mathbb{L}_5}) = 0.0012$$

avec $e_{1_63_6}e_{3_69_6}e_{9_61_6}e_{1_67_6}e_{7_68_6}e_{8_64_2}$ un préfixe de l'automate A_6, $e_{4_21_5}$ un événement de l'automate A_2 et $e_{1_522_5}$ un événement de l'automate A_5.

TABLE 4.3 – Probabilités d'occurrence des séquences d'événements décrivant l'évolution du sous-système en régime transitoire

Séquence (s_i)	**État initial**	**État final**	**Probabilité d'occurrence** ($\mathbb{P}(s_i)$)
$s_3 = e_{1_{2'}7_{2'}} e_{7_{2'}8_{2'}} e_{8_{2'}4_1} e_{4_1 1_5}$ $e_{1_5 3_5} e_{3_5 4_5} e_{4_5 5_5} e_{5_5 1_5}$	état 1, Mode $2'$	état 1, Mode 5	0.4435
$s_4 = e_{1_{2'}3_{2'}} e_{3_{2'}9_{2'}} e_{9_{2'}1_{1'}} e_{1_{1'}7_{1'}}$ $e_{7_{1'}8_{1'}} e_{8_{1'}4_2} e_{4_2 1_5}$	état 1, Mode $2'$	état 1, Mode 5	0.0015
$s_5 = e_{1_5 10_5} e_{10_5 11_5} e_{11_5 5_5} e_{5_5 1_5}$ $e_{1_5 1_8} e_{1_8 2_8} e_{2_8 3_8} e_{3_8 28_8}$	état 1, Mode 5	état 28, Mode 8	$8.0995 \cdot 10^{-6}$
$s_6 = e_{6_{1'}10_{1'}} e_{10_{1'}1_{1'}} e_{1_{1'}7_{1'}} e_{7_{1'}8_{1'}}$ $e_{8_{1'}4_2} e_{4_2 1_5} e_{1_5 13_5} e_{13_5 14_5} e_{14_5 15_5} e_{15_5 22_5}$	état 6, Mode $1'$	état 22, Mode 5	0.0718
$s_7 = e_{7_{1'}8_{1'}} e_{8_{1'}4_2} e_{4_2 1_5} e_{1_5 3_5}$ $e_{3_5 4_5} e_{4_5 5_5}$	état 7, Mode $1'$	état 5, Mode 5	0.6478

Le tableau (4.3) présente les résultats obtenus pour d'autres séquences d'événements qui conduisent le système dans des états de bon fonctionnement ou de défaillance. Les deux premières séquences (s_3 et s_4) amènent le système dans l'état 1 (nominal) du Mode 5, correspondant à la conduite en puissance à $100\%P_n$, selon deux trajectoires différentes. La séquence s_5 conduit le système depuis l'état 1 du Mode 5 jusqu'à l'état critique 28 (AAR) du Mode 8. La séquence s_6 amène le système à l'état critique 22 (AAR) du Mode 5. La dernière séquence, s_7, conduit à l'état 5 du Mode 5 où la TPA 1 est en attente et la TPA 2 est en marche (la puissance est à $60\%P_n$).

Nous pouvons remarquer que les probabilités d'occurrence des séquences critiques sont bien plus fortes en phase de conduite en puissance (s_5) qu'en phase de baisse de puissance (s_6). Ceci peut s'expliquer par le rapport des durées associées à ces deux phases. De manière attendue dans le Mode 5, les séquences non critiques (s_3, s_7) sont beaucoup plus probables que la séquence critique (s_6).

La section suivante concerne l'évaluation de séquences d'événements qui décrivent le comportement du système en régime asymptotique.

4.5 Évaluation de séquences d'événements en régime asymptotique

Conformément aux hypothèses afférentes, la conduite en puissance se déroule sur une période de temps de 18 mois, l'équivalent de 13128 heures alors que les deux autres phases (montée en puissance et baisse de puissance) ont des durée beaucoup plus courtes de l'ordre de 24 heures en régime nominal. En conséquence, l'étude en régime asymptotique n'a de sens que pour le Mode 5, correspondant à la phase de conduite en puissance, puisque la durée des autres phases est insuffisante pour que les probabilités atteignent leur valeur asymptotique.

Ce mode est représenté par le modèle semi-markovien correspondant à l'automate A_5 de la figure (4.11) ainsi que par le p-automate (chaîne de Markov immergée) de la figure (4.12). Le calcul des probabilités d'occurrence de chaque événement se fait à partir de la matrice $Q(t) = [Q_{ij}(t)]$ qui est le noyau du processus semi-markovien, en appliquant l'équation (1.21). Par la suite, les étapes de l'approche proposée dans la figure (2.1) de la section 2.3 sont suivies pour obtenir

l'évaluation quantitative de séquences en régime asymptotique. Pour réaliser une comparaison entre les résultats obtenus en régime transitoire et ceux obtenus en régime asymptotique, nous avons considéré les trois états cibles 1, 5 et 22, utilisés pour le calcul en régime transitoire.

Pour l'automate déduit du p-automate de la figure (4.12) dans lequel les probabilités sont remplacées par des événements e_{ij} où i représente l'état source de la transition et j l'état cible, les sous-langages associés aux états sont donnés par le système d'équations suivant :

$$\begin{cases} L_1 = L_5 e_{51} + L_{15} e_{(15)1} \\ L_3 = L_1 e_{13} + L_4 e_{43} \\ L_4 = L_3 e_{34} \\ L_5 = L_4 e_{45} + L_{11} e_{(11)5} \\ L_{10} = L_1 e_{1(10)} + L_{11} e_{(11)(10)} \\ L_{11} = L_{10} e_{(10)(11)} \\ L_{13} = L_1 e_{1(13)} + L_{14} e_{(14)(13)} \\ L_{14} = L_{13} e_{(13)(14)} \\ L_{15} = L_{14} e_{(14)(15)} + L_{21} e_{(21)(15)} \\ L_{20} = L_1 e_{1(20)} + L_{21} e_{(21)(20)} \\ L_{21} = L_{20} e_{(20)(21)} \\ L_{22} = L_1 e_{1(22)} + L_5 e_{5(22)} + L_{15} e_{(15)(22)} \end{cases}$$

Les expressions régulières, solutions de ce système, sont obtenues par application du lemme d'Arden (2.10). Nous présentons ci-dessous les sous-langages L_1, L_5 et L_{22} correspondant aux états cibles retenus précédemment.

$$L_1 = \{[e_{13}e_{34}\left(e_{43}e_{34}\right)^* e_{45} + e_{1(10)}e_{(10)(11)}\left(e_{(11)(10)}e_{(10)(11)}\right)^* e_{(11)5}]e_{51} + [e_{1(13)}e_{(13)(14)}\left(e_{(14)(13)}e_{(13)(14)}\right)^* e_{(14)(15)} + e_{1(20)}e_{(20)(21)}\left(e_{(21)(20)}e_{(20)(21)}\right)^* e_{(21)(15)}]e_{(15)1}\}^*$$

$$L_5 = L_1[e_{13}e_{34}\left(e_{43}e_{34}\right)^* e_{45} + e_{1(10)}e_{(10)(11)}\left(e_{(11)(10)}e_{(10)(11)}\right)^* e_{(11)5}]$$

$$L_{22} = L_1\{e_{1(22)} + [e_{13}e_{34}\left(e_{43}e_{34}\right)^* e_{45} + e_{1(10)}e_{(10)(11)}\left(e_{(11)(10)}e_{(10)(11)}\right)^* e_{(11)5}]e_{5(22)} + [e_{1(13)}e_{(13)(14)}\left(e_{(14)(13)}e_{(13)(14)}\right)^* e_{(14)(15)} + e_{1(20)}e_{(20)(21)}\left(e_{(21)(20)}e_{(20)(21)}\right)^* e_{(21)(15)}]e_{(15)(22)}\}$$

Ces sous-langages contiennent l'ensemble des séquences qui conduisent le système depuis l'état initial 1 vers l'état cible (1, 5 ou 22).

Connaissant les probabilités des événements, les probabilités des itérations, qui composent ces expressions régulières, ont été déterminées à l'aide des équations (2.39) et (2.40) du chapitre II. Sur cette base, la somme des probabilités des séquences appartenant à chaque sous-langage a été calculée numériquement :

$$\mathbb{Q}\left(L_1\right) = \sum_{s_i \in L_1} \mathbb{P}\left(s_i\right) = 1$$

$$\mathbb{Q}\left(L_5\right) = \sum_{s_i \in L_5} \mathbb{P}\left(s_i\right) = 0.8203$$

$$\mathbb{Q}\left(L_{22}\right) = \sum_{s_i \in L_{22}} \mathbb{P}\left(s_i\right) = 0.5327$$

Le résultat obtenu pour le langage L_1 s'explique par le fait que le graphe est fortement connexe et autorise toujours le retour à l'état initial. Nous pouvons également constater que les

probabilités des séquences d'événements obtenus en régime transitoire ($F_{11}(t) = 0.9921$, $F_{15}(t) = 0.8135$, $F_{1(22)}(t) = 0.5212$) convergent vers les valeurs obtenues en régime asymptotique. Ce résultat contribue à la validation de l'approche proposée.

A partir des expressions régulières associées aux sous-langages L_1, L_5 et L_{22}, il est possible d'évaluer la probabilité d'occurrence des séquences constituant les mots de ces sous-langages. A titre d'exemple, les résultats obtenus pour quelques séquences particulières sont donnés dans les tableaux 4.4, 4.5 et 4.6.

TABLE 4.4 – Probabilités d'occurrence pour quelques séquences conduisant à l'état 1

Séquence (s_i)	**Probabilité d'occurrence** $\mathbb{P}(s_i)$
$s_1 = e_{13}e_{34}e_{45}e_{51}$	0.1029
$s_2 = e_{13}e_{34}e_{43}e_{34}e_{45}e_{51}$	0.0102
$s_3 = e_{1(10)}e_{(10)(11)}e_{(11)5}e_{51}$	0.3018
$s_4 = e_{1(13)}e_{(13)(14)}e_{(14)(15)}e_{(15)1}$	0.1029

TABLE 4.5 – Probabilités d'occurrence pour quelques séquences conduisant à l'état 5

Séquence (s_i)	**Probabilité d'occurrence** $\mathbb{P}(s_i)$
$s_{11} = e_{13}e_{34}e_{45}$	0.1029
$s_{12} = e_{13}e_{34}e_{43}e_{34}e_{45}$	0.0102
$s_{13} = e_{1(10)}e_{(10)(11)}e_{(11)5}$	0.3020
$s_{14} = e_{1(20)}e_{(20)(21)}e_{(21)(15)}e_{(15)1}e_{13}e_{34}e_{45}$	0.0310

TABLE 4.6 – Probabilités d'occurrence pour quelques séquences conduisant à l'état 22

Séquence (s_i)	**Probabilité d'occurrence** $\mathbb{P}(s_i)$
$s_{21} = e_{1(22)}$	0.0999
$s_{22} = e_{13}e_{34}e_{45}e_{5(22)}$	$6.0644 \cdot 10^{-5}$
$s_{23} = e_{1(10)}e_{(10)(11)}e_{(11)5}e_{5(22)}$	$1.7789 \cdot 10^{-4}$
$s_{24} = e_{1(20)}e_{(20)(21)}e_{(21)(15)}e_{(15)(22)}$	$1.7789 \cdot 10^{-4}$

4.6 Conclusion

Ce chapitre a permis d'illustrer l'application de l'approche qui concerne la détermination et l'évaluation quantitative de séquences d'événements et de l'approche modulaire sur un sous-ensemble d'un cas test industriel (régulation du niveau d'eau dans un Générateur de Vapeur d'un Réacteur à Eau Pressurisée).

Les principes de fonctionnement du cas d'étude font apparaître trois phases de conduite (démarrage et montée en puissance, conduite en puissance et baisse de puissance) et plusieurs modes de fonctionnement ou de défaillance. Les modèles comportementaux (automates à états finis, chaîne de Markov à temps continu, processus semi-markoviens et p-automates) décrivant les différents modes de fonctionnement servent de base aux calculs de probabilités des séquences d'événements en régime transitoire et asymptotique. Les calculs effectués illustrent l'apport des

approches proposées au chapitre II, où les probabilités des séquences sont évaluées localement mode par mode, mais aussi de l'approche modulaire proposée au chapitre III.

Les résultats obtenus pour les calculs des probabilités locales mode par mode montrent la pertinence du cadre formel d'évaluation probabiliste proposé. Par ailleurs, ils ont permis de vérifier la convergence entre calculs en régime transitoire et calculs en régime asymptotique. L'application de l'approche modulaire a, quant à elle, montré la capacité des opérateurs de choix et de concaténation à traiter des problèmes de grande dimension et de forte complexité.

Conclusions & Perspectives

Dans ce mémoire, nous avons proposé un cadre formel d'identification et d'évaluation quantitative des séquences d'événements dans les études de sûreté de fonctionnement des systèmes réparables, reconfigurables, multi-modes et multi-états. Les fondements de ce cadre formel reposent sur la théorie des langages probabilistes développée par Garg, Kumar et Marcus [Garg *et al.*, 1999].

Dans un premier temps, nous avons défini une approche permettant l'identification et d'évaluer la probabilité d'occurrence de séquences d'événements en régime asymptotique (distribution stationnaire des probabilités d'états) et transitoire. La modélisation du système repose sur un automate à état fini, à partir duquel les séquences sont identifiées à l'aide de techniques classiques sur les langages rationnels. L'évaluation des probabilités d'occurrence des séquences est basée, quant à elle, sur la détermination d'un p-automate. En régime asymptotique, celui-ci est obtenu à l'aide d'une chaîne de Markov à temps discrets immergée dans un processus stochastique continu alors qu'en régime transitoire, on considère la loi de distribution du temps de premier passage vers un état donné obtenue par une transformée de Laplace (convolution des temps de passage pour tous les chemins amenant à un état donné). Cette première approche a été étendue pour l'évaluation de la criticité des séquences non nécessairement limitée à leur probabilité d'occurrence (coût et longueur des séquences).

Si ce cadre formel se révèle efficace pour des systèmes de taille et complexité réduite (tel que le cas d'étude du four traité dans le chapitre 2), il présente quelques limites relatives à l'obtention et au traitement des expressions régulières qui décrivent les séquences pour des systèmes à échelle industrielle caractérisés par plusieurs modes (modes nominaux ou dégradés de fonctionnement, modes de défaillance, *etc.*). Nous avons donc été amené à étendre ce cadre par des opérateurs de composition autorisant une approche modulaire du problème. Il a été montré, qu'à l'aide d'une généralisation des opérateurs de choix et concaténation définis dans le cadre de la théorie de langages probabilistes [Garg *et al.*, 1999], il est possible de procéder à une évaluation quantitative globale des séquences à partir des évaluations obtenues localement, mode par mode, sur des modèles de taille raisonnable.

L'avantage majeur de notre contribution est relatif au cadre formel sur lequel elle repose qui permet l'identification, à l'aide d'un calcul analytique, des séquences d'événements sans nécessiter leur modélisation explicite (comme dans un arbre d'événements) ou par opposition à une obtention par exploration du modèle (comme pour les BDMP). Dans le même ordre d'idée, le calcul des probabilités d'occurrence des séquences (ou de leur coût) est effectué de manière analytique (en régime asymptotique) et numérique (en régime transitoire) sans recours à la simulation de Monte-Carlo. Le calcul analytique n'est pas limité aux chaînes de Markov à temps continu mais permet de prendre en comte des lois non exponentielles dans le cadre d'un processus semi-markovien. Enfin, les opérateurs de composition généralisés permettent la mise en œuvre

d'une approche modulaire permettant la quantification de séquences d'un système reconfigurable et multi-modes sans nécessiter la construction d'un modèle global intégrant ces différents modes.

Ces différents points forts de notre contribution ont été mis en évidence par l'application de nos propositions sur deux cas d'études de complexité croissante : un système de régulation d'un four et un sous-ensemble d'un système d'alimentation en eau d'un Générateur de Vapeur (GV) d'un Réacteur à Eau Pressurisée. Les séquences d'événements dont les probabilités ont été évaluées concernait les états dangereux de ces systèmes : température non régulée pour le cas d'étude du four et perte de l'alimentation en eau pour le cas d'étude du GV. Plusieurs modes de fonctionnement ont été considérés (maintenance parfaire ou imparfaite pour le four, conduite en démarrage et montée en puissance, conduite en puissance et conduite en baisse de puissance pour le GV) ainsi que plusieurs modes de défaillances (défaillances indépendantes et défaillances de cause commune pour le four).

Au terme de ces travaux, nous envisageons plusieurs perspectives de recherche, d'une part en élargissant le champ des systèmes étudiés, et d'autre part, pour améliorer les méthodes formelles proposées pour l'identification et la quantification des séquences d'événements.

En ce qui concerne le premier point, les systèmes considérés dans le cadre de cette étude, font l'hypothèse que la détection de défaillance est déterministe (soit instantanée, soit à durée constante). Une première extension de notre approche consisterait à introduire des aspects probabilistes dans le processus de diagnostic et d'évaluer son impact sur la fiabilité et la disponibilité du système étudié. D'autre part, les systèmes considérés sont exclusivement des systèmes de contrôle-commande étudiés dans un contexte de disponibilité, il serait intéressant de considérer également les aspects relatifs à la sécurité en appliquant notre approche aux systèmes instrumentés de sécurité, notamment pour la prise en compte de multiples redondances ou de défaillances cachées et dormantes.

Les extensions envisagées sur notre cadre formel portent sur l'identification des séquences et les opérateurs de composition.

L'identification des séquences est réalisée dans notre étude à l'aide d'une technique classique sur les langages rationnels. Nous avons montré les limites relatives à ces techniques, notamment relatives au phénomène d'explosion des expressions à manipuler. Cette technique pourrait être couplée avec les contributions de la thèse de P.Y. Chaux [Chaux, 2013] portant sur la génération, à partir d'un modèle automate à état fini, d'un ensemble de séquences minimales qui permettent de reconstruire toutes les séquences conduisant à un état redouté. Le couplage avec cette approche permettrait, notamment pour les études de fiabilité, de limiter la construction des séquences critiques aux seules séquences minimales identifiées dans [Chaux, 2013].

Enfin, les opérateurs de composition, généralisés dans le cadre de ce mémoire, pourraient faire l'objet de développements supplémentaires. Dans un premier temps, les différents situations caractérisant le contexte d'utilisation de l'opérateur de concaténation, et notamment de la nature de l'événement de commutation, pourraient être homogénéisées. En particulier, la composition d'automates, décrivant non plus différents modes de fonctionnement mais des composants distincts dont les langages sont totalement disjoints, pourrait nécessiter une redéfinition de l'opérateur de concaténation, voire la définition d'un nouvel opérateur dédié. Par ailleurs, dans le cadre de notre approche, les opérateurs de composition permettent le calcul de probabilités de séquences globales à partir de résultats obtenus localement sur des modèles de type automate à états finis (p-automates). Il pourrait être intéressant d'étendre ces mécanismes à la composition de mo-

dèles de nature hétérogène, spécifiques à une classe d'application ou à un problème donné, et non limités aux seuls automates. A titre d'exemple, les modèles locaux relatifs à la description de comportements purement combinatoire pourraient être décrits à l'aide de diagramme de Hasse ou de graphes alors que des comportements continus pourraient être décrit à l'aide d'automates stochastiques hybrides, notamment pour des problèmes de fiabilité dynamique. Ceci permettrait d'une part, de réduire la modélisation des modes de fonctionnement, des modes de défaillances, des états, ... aux seuls éléments pertinents pour l'étude, et d'autre part, de combiner des modèles au pouvoir d'expression adapté à chacun des éléments modélisés et analysés. Une telle extension ne pourra très vraisemblablement pas se limiter à une extension des opérateurs généralisés mais nécessitera d'en introduire de nouveaux.

Bibliographie

[CEI, 1990a] (1990a). *CEI 50 191 Vocabulaire Electrotechnique International, Chapitre 191 – Sûreté de fonctionnement et qualité des services.*

[CEI, 1990b] (1990b). *CEI 50 192 Vocabulaire Electrotechnique International, Chapitre 192 – Sûreté de fonctionnement.*

[CEI, 2000] (2000). *Sécurité fonctionnelle des systèmes électriques/électroniques/éléctroniques programmables relatifs à la sûreté.*

[Acosta et Siu, 1991] Acosta, C. et Siu, N. (1991). *Dynamic event tree analysis method (DETAM) for accident sequence analysis.* MITNE-295.

[Acosta et Siu, 1993] Acosta, C. et Siu, N. (1993). Dynamic event trees in accident sequence analysis : application to steam generator tube rupture. *Reliability Engineering and System Safety, Elsevier*, 41:135–154.

[Akers, 1978] Akers, S. (1978). Binary decision diagrams. *IEEE Trans. Computers*, C-27(6): 509–516.

[Alam et Al-Saggaf, 1986] Alam, M. et Al-Saggaf, U. (1986). Quantitative reliability evaluation of repairable phased-mission systems using markov approach. *IEEE Transactions on Reliability*, 35(5):498–503.

[Aubry et Brinzei, 2016] Aubry, J. et Brinzei, N. (2016). *Systems Dependability Assessment : Benefits of Petri net models.* Wiley-ISTE.

[Aubry *et al.*, 2012a] Aubry, J.-F., Babykina, G., Barros, A., Brînzei, N., Deleuze, G., Saporta, B. D., Dufour, F., Langeron, Y. et Zhang, H. (2012a). Rapport final du projet approdyn : Approches de la fiabilité dynamique pour modéliser des systèmes critiques. Rapport technique, 3SGS.

[Aubry *et al.*, 2012b] Aubry, J.-F., Babykina, G., Brînzei, N., Medjaher, S., Barros, A., Bérenguer, C., Grall, A., Langeron, Y., Nguyen, D. N., Deleuze, G., Saporta, B. D., Dufour, F. et Zhang, H. (2012b). *Supervision and Safety of Complex Systems.* ISBN 978-1-84821-413-2, Wiley- ISTE, London.

[Aubry et Brînzei, 2015] Aubry, J.-F. et Brînzei, N. (2015). *Systems Dependability Assessment, Modeling with Graphs and Finite State Automata.* Risk Management and Dependability Series. Wiley.

[Aven, 2003] Aven, T. (2003). *Foundations of risk analysis.* Wiley, New Jersey.

[Babykina *et al.*, 2016] Babykina, G., Brînzei, N., Aubry, J.-F. et Deleuze, G. (2016). Modeling and simulation of a controlled steam generator in the context of dynamic reliability using a stochastic hybrid automaton. *Reliability Engineering and System Safety, Elsevier*, (152):115–136.

[Bedford et Cooke, 2001] Bedford, T. et Cooke, R. (2001). *Probabilistic risk analysis*. Cambridge University Press, Cambridge.

[Best *et al.*, 2001] Best, E., Devillers, R. et Koutny, M. (2001). *Petri net algebra*. Springer.

[Birnbaum *et al.*, 1961] Birnbaum, Z., Esary, J. et Saunders, S. (1961). Multi-component systems and structures and their reliability. *In Technometrics*, volume 3, pages 55–77.

[Bouissou, 1993] Bouissou, M. (1993). The figaro dependability evaluation workbench in use : case studies for fault tolerant computer systems. *In FTCS'23, Toulouse, June 1993*.

[Bouissou, 2007] Bouissou, M. (2007). A generalization of dynamic fault trees through boolean logic driven markov processes (bdmp). *In European Safety and Reliability Conference, ESREL 2007*.

[Bouissou, 2008] Bouissou, M. (2008). *Gestion de la complexité dans les études quantitatives de sûreté de fonctionnement de systèmes*. Editions Lavoisier.

[Bouissou et Bon, 2003] Bouissou, M. et Bon, J. (2003). A new formalism that combines advantages of fault-trees and markov models : Boolean logic driven markov processes. *Reliability Engineering and System Safety*, 82(2):149–163.

[Bouissou *et al.*, 1991] Bouissou, M., Villate, N. et Lucas, J.-Y. (1991). Présentation de l'atelier figaro d'études de fiabilité des systèmes. *In Congrès IA, Avignon, 1991*.

[Breeding *et al.*, 1992] Breeding, R., Helton, J., Gorham, E. et Harper, F. (1992). Summary description of the methods used in the probabilistic risk assessment for NUREG-1150. *Nuclear Engineering and Design*, 135(1):1–27.

[Bryant, 1987] Bryant, R. (1987). Graph based algorithms for boolean function manipulation. *IEEE Trans. Computers*, 35(8):677–691.

[Bryant, 1992] Bryant, R. (1992). Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318.

[Burdick *et al.*, 1977] Burdick, G.-R., Fussel, J.-B., Rasmuson, D.-M. et Wilson, J.-R. (1977). Phased mission analysis : A review of new developments and an application. *IEEE Transactions on Reliability*, R-26:43–49.

[Cai *et al.*, 2012] Cai, B., Liu, Y., Liu, Z., Tian, X., Zhang, Y. et Liu, J. (2012). Performance evaluation of subsea blowout preventer systems with common-cause failures. *Journal of Petroleum Science and Engineering*, pages 18–25.

[Carton, 2008] Carton, O. (2008). *Langages formels, calculabilité et complexité*. Vuibert (6 octobre 2008), Paris.

[Cassandras et Lafortune, 1999] Cassandras, C. et Lafortune, S. (1999). *Introduction to discrete event systems*. Kluwer Academic Publishers.

[Cassandras et Lafortune, 2008] Cassandras, C. et Lafortune, S. (2008). *Introduction to Discrete Event System*. Springer, New-York, second edition édition.

[Cepin, 2011] Cepin, M. (2011). *Assessment of Power System Reliability*. Springer.

[Cepin et Mavko, 2002] Cepin, M. et Mavko, B. (2002). A dynamic fault tree. *Reliability Engineering and System Safety*, 75:83–91.

[Chaux, 2013] Chaux, P. (2013). *Formalisation de la cohérence et calcul des séquences de coupe minimales pour les systèmes binaires dynamiques réparables*. Thèse de doctorat, ENS Cachan.

[Chiola *et al.*, 1993a] Chiola, G., Ajmone Marsan, M., Balbo, G. et Conte, G. (1993a). Generalized stochastic petri nets : A definition at the net level and its implications. *IEEE Transactions on Software Engineering*, 19(2):89–107.

[Chiola *et al.*, 1993b] Chiola, G., Marsan, M. A., Balbo, G. et Conte, G. (1993b). Generalized stochastic petri nets : a definition at the net level and its implications. *IEEE Transactions on Software Engineering*, 19(2):89–107.

[Chraibi, 2013] Chraibi, H. (2013). Dynamic reliability modeling and assessment with pycatshoo : Application to a test case. *In Proceedings of Probabilistic Safety Assessment and Management (PSAM)*, Tokyo, Japan.

[Clarhaut, 2009] Clarhaut, J. (2009). *Prise en compte des séquences de défaillances pour la conception de systèmes d'automatisation.* Thèse de doctorat, Université des Sciences et des Technologies de Lille.

[Cojazzi, 1996] Cojazzi, G. (1996). The dylam approach for the dynamic reliability analysis of systems. *Reliability Engineering and System Safety, Elsevier*, 52:279–296.

[Coudert et Madre, 1992] Coudert, O. et Madre, J. (1992). A new method to compute prime and essential prime implicants of boolean functions. *In* Knight et (Eds), S., éditeurs : *Advanced Research in VLSI and Parallel Systems*, pages 113–128. The Mit Press.

[Coudert et Madre, 1994] Coudert, O. et Madre, J. (1994). Metaprime : an interactive fault-tree analyzer. *IEEE Trans. Reliability*, 43(1):121–127.

[Crow, 1975] Crow, L.-H. (1975). Reliability analysis for complex, repairable systems. Rapport technique, Technical report, U.S. Army Material Systems Analysis Activity.

[Csenki, 1995] Csenki, A. (1995). An integral equation approach to the interval reliability of systems modelled by finite semi-markov processes. *Reliability Engineering and System Safety*, 47(1):37–45.

[Delahaye *et al.*, 2010] Delahaye, B., Caillaud, B. et Legay, A. (2010). Probabilistic contracts : A compositional reasoning methodology for the design of stochastic systems. *In Application of Concurrency to System Design (ACSD), 2010 10th International Conference on*, pages 223 – 232. IEEE.

[Dugan *et al.*, 1992] Dugan, J., Bavuso, S. et Boyd, M. (1992). Dynamic fault-tree models for fault tolerant computer systems. *IEEE Transactions on Reliability*, 41(3):363–377.

[Dugan *et al.*, 2000] Dugan, J., Sullivan, K. et Coppit, D. (2000). Developing a low-cost highquality software tool for dynamic fault trees analysis. *IEEE Transactions on Reliability*, 49(1):49–59.

[Faraut *et al.*, 2010] Faraut, G., Piétrac, L. et Niel, E. (2010). Control law synthesis and reconfiguration using sct. *In In Proc. Conference on Control and Fault-Tolerant Systems (SysTol 2010)*, pages 576–581, Nice, France.

[Flammini, 2006] Flammini, F. (2006). *Model-based dependability evaluation of complex critical control systems.* Thèse de doctorat, Universita' degli Studi di Napoli Federico II.

[Fussel *et al.*, 1976] Fussel, J.-B., Aber, E.-F. et Rahl, R.-G. (1976). On the quantitative analysis of priority-and failure logic. *IEEE Transactions on Reliability*, R(25):324–326.

[Garg, 1992] Garg, V. (1992). An algebraic approach to modeling probabilistic discrete event systems. *In Proc. On the* 31^{st} *Conf. on Decision and Control*, pages 2348–2353, Tucson, USA.

[Garg *et al.*, 1999] Garg, V., Kumar, R. et Marcus, S. (1999). A probabilistic language formalism for stochastic discrete-event systems. *IEEE Trans. on Automatic Control*, 44(2):280–293.

[Grumberg et Long, 1991] Grumberg, O. et Long, D. (1991). Model checking and modular verification. *Lecture Notes in Computer Science*, 527:250–265.

[Haas, 2010] Haas, P. J. (2010). *Stochastic Petri Nets : Modelling, Stability, Simulation.* Springer-Verlag New York.

[Habrard, 2004] Habrard, A. (2004). *Modèles et Techniques en Inference Grammaticale Probabiliste : de la Gestion du Bruit à l'Extraction de Connaissances.* Thèse de doctorat, Université Jean Monnet de Saint-Etienne.

[Hanley et Kumamoto, 1981] Hanley, E. et Kumamoto, H. (1981). *Reliability Engineering and Risk Assessment.* Prentice Hall.

[Harrison et Knottenbelt, 2002] Harrison, P. et Knottenbelt, W. (2002). Passage time distributions in large markov chains. *In* ACM New York, NY, U., éditeur : *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, volume Volume 30 Issue 1, pages 77–85.

[Hawkes et Sykes, 1990] Hawkes, A. et Sykes, A. (1990). Equilibrium distribution of finite-state markov processes. *Reliability, IEEE Transactions on*, 39(5):592 – 595.

[Hermanns, 2002] Hermanns, H. (2002). *Interactive Markov Chains.* Lecture Notes in Computer Science (Book 2428). Springer, edition (September 11, 2002).

[Howard, 1971a] Howard, R. (1971a). *Dynamic Probabilistic Systems, Vol. 2 : Semi-Markov and Decision Processes.* John Wiley and Sons.

[Howard, 1971b] Howard, R. (1971b). *Dynamic Probabilistic Systems, Volume I : Markov Models.* John Wiley and Sons.

[Hsueh et Mosleh, 1996] Hsueh, K. et Mosleh, A. (1996). The development and application of the accident dynamic simulator for probabilistic risk assessment of nuclear power plants. *Reliability Engineering and System Safety, Elsevier*, 52(279-296).

[Husson *et al.*, 2007] Husson, R., Lung, C., Aubry, J.-F., Daafouz, J. et Wolf, D. (2007). *Automatique : du cahier des charges à la réalisation de systèmes.* Sciences Sup., Dunod Ed., ISBN : 978-2-10-050397-1.

[Iosifescu *et al.*, 2007] Iosifescu, M., Limnios, N. et Oprisan, G. (2007). *Modèles stochastiques.* Lavoisier.

[Iosifescu *et al.*, 2010] Iosifescu, M., Limnios, N. et Oprisan, G. (2010). *Introduction to Stochastic Models.* Wiley-ISTE.

[J.Abate *et al.*, 2000] J.Abate, Choudhury, G. et Whitt, W. (2000). *Computational Probability*, chapitre An introduction to numerical transform inversion and its application to probability models, pages 257–323. Springer US, Kluwer, Boston.

[Kanso et Berruet, 2010] Kanso, M. et Berruet, P. (2010). Rms : une évolution des systèmes manufacturiers. Rapport technique, Techniques de l'Ingénieur.

[Kaplan et Garrick, 1981] Kaplan, S. et Garrick, B. (1981). On the quantitative definition of risk. *Risk Analysis*, 1:1–11.

[Kemeny et Snell, 1976] Kemeny, J. G. et Snell, J. L. (1976). *Finite Markov Chains.* Springer-Verlag.

[Kermisch et Labeau, 2000] Kermisch, C. et Labeau, P.-E. (2000). Approche dynamique de la fiabilité des systèmes. Rapport de projet ISdF, Report.

[Kombé, 2011] Kombé, T. (2011). *Modélisation de la propagation des fautes dans les systèmes de production.* Thèse de doctorat, Institut National des Sciences Appliquées de Lyon.

[Kumar et Garg, 2001] Kumar, R. et Garg, V. (2001). Control of stochastic discrete event systems modeled by probabilistic languages. *IEEE Trans. on Automatic Control*, 46(4):593–606.

[Kumar *et al.*, 1996] Kumar, R., Garg, V. et Marcus, S. (1996). Modeling stochastic discrete event systems using probabilistic languages. *In Mathematical Theory of Networks and Systems*, St. Louis, USA.

[Laprie, 2004] Laprie, J.-C. (2004). Sûreté de fonctionnement informatique : concepts, défis, directions. *In ACI Sécurité et Informatique*, Toulouse. CNRS, LAAS.

[Laprie *et al.*, 1995] Laprie, J.-C., Arlat, J., Blanquart, J.-P., Costes, A., Crouzet, Y., Deswarte, Y., Fabre, J.-C., Guillermain, H., Kaâniche, M., Kanoun, K., Mazet, C., Powell, D., Rabéjac, C. et Thévenod, P. (1995). *Guide de la Sûreté de fonctionnement.* Cépaduès Éditions.

[Leveson, 1995] Leveson, N. (1995). *Safeware : System Safety and Computers.* Addison-Wesley.

[Limnios, 2005] Limnios, N. (2005). *Arbres de défaillances.* Hermes Science, (2e édition revue et augmentée) édition.

[Maraninchi, 1992] Maraninchi, F. (1992). Operational and compositional semantics of synchronous automaton compositions. *Lecture Notes in Computer Science*, 630:550–564.

[Marsan *et al.*, 1984] Marsan, M. A., Conte, G. et Balbo, G. (1984). A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2:93–122.

[Meshkat *et al.*, 2003] Meshkat, L., Xing, L., Donohue, S.-K. et Ou, Y. (2003). An overview of the phase-modular fault-tree approach to phased mission system analysis. *In In Proc. International Conference on Space Mission Challenges for Information Technology (SMC-IT 2003)*, Pasadena (California-USA).

[Molloy, 1982] Molloy, M. K. (1982). Performance analysis using stochastic petri nets. *IEEE Transactions on Computers*, 31(9):913–917.

[Murata, 1989] Murata, T. (1989). Petri nets : Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

[Pantelic et Lawford, 2012] Pantelic, V. et Lawford, M. (2012). A pseudometric in supervisory control of probabilistic discrete event systems. *Discrete Event Dyn. Syst.*, (22):479–510.

[Paoli *et al.*, 2011] Paoli, A., Sartini, M. et Lafortune, S. (2011). Active fault tolerant control of discrete event systems using online diagnostics. *Automatica*, 47(4):639–649.

[Papazoglou, 1998] Papazoglou, I. (1998). Mathematical foundations of event trees. *Reliability Engineering and System Safety*, 61(3):169–183.

[Peres *et al.*, 2011] Peres, F., Berthomieu, B. et Vernadat, F. (2011). On the composition of time petri nets. *Discrete Event Dynamic Systems*, 21(3):395–424.

[Perez-Castaneda, 2009] Perez-Castaneda, G. (2009). *Evaluation par simulation de la sûreté de fonctionnement de systèmes en contexte dynamique hybride.* Thèse de doctorat, Institut National Polytechnique de Lorraine.

[Perez-Castaneda *et al.*, 2011] Perez-Castaneda, G., Aubry, J.-F. et Brinzei, N. (2011). Stochastic hybrid automata model for dynamic reliability assessment. *Proceedings of the Institution of Mechanical Engineers Part O Journal of Risk and Reliability*, 225(1):28–41.

[Perman *et al.*, 1997] Perman, M., Senegacnik, A. et Tuma, M. (1997). Semi-markov models with an application to power-plant reliability analysis. *Reliability, IEEE Transactions on*, 46(4):526 – 532.

[Piriou, 2015] Piriou, P.-Y. (2015). *Contribution à l'analyse de sûreté de fonctionnement basée sur les modèles des systèmes dynamiques, réparables et reconfigurables.* Thèse de doctorat, Université Paris-Saclay.

[Rauzy, 2011] Rauzy, A. (2011). Sequence algebra, sequence decision diagrams and dynamic fault trees. *Reliability Engineering and System Safety*, 96:785–792.

[Rozanov, 1975] Rozanov, Y. (1975). *Processus aléatoires.* Editions de Moscou.

[Signoret, 2005] Signoret, J.-P. (2005). Analyse des risques des systèmes dynamiques : approche markovienne. Rapport technique, Techniques de l'Ingénieur.

[V. S. Barbu, 2008] V. S. Barbu, N. L. (2008). *Semi-Markov Chains and Hidden Semi-Markov Models toward Applications. Their Use in Reliability and DNA Analysis.* Springer.

[Villemeur, 1988] Villemeur, A. (1988). *Sûreté de fonctionnement des systèmes industriels.* Collection de la Direction des Etudes et Recherches d'Electricité de France, Editions Eyrolles, ISBN 2-212-01615-8.

[Villemeur, 1997] Villemeur, A. (1997). *Sûreté de fonctionnement des systèmes industriels.* Edition Eyrolles.

[Wang et Ray, 2004] Wang, X. et Ray, A. (2004). A language measure for performance evaluation of discrete-event supervisory control systems. *Applied Mathematical Modelling*, 28:817–833.

[Zio, 2013] Zio, E. (2013). *The Monte Carlo Simulation Method for System Reliability and Risk Analysis.* Springer Series in Reliability Engineering. Springer-Verlag, London.

Annexe A

Modélisation des modes de fonctionnement du système des turbopompes TPA

Cette annexe présente l'ensemble des modes considérés à l'étape de modélisation du sous-système constitué par les deux turbopompes (TPA) dans le chapitre IV. Ces modes ont été construits par rapport aux trois phases de conduite que ce sous-système peut suivre.

Conduite en démarrage et montée en puissance

A l'instant initial les deux turbopompes TPA sont considérées à l'arrêt ; le premier mode de fonctionnement (Mode 1) représente l'utilisation de la turbopompe TPA 1 pour démarrer le système et le faire monter en puissance. L'automate à états finis A_1 de la figure (A.1) corresponde au Mode 1 et indique les états possibles du système et les événements déterminant les transitions entre ces états. Le p-automate associé à l'automate A_1 est donné dans la figure (A.2).

Le Mode 2 représente cas où la turbopompe TPA 2 est sollicitée premièrement pour démarrer le système et le faire monter en puissance. L'automate à états finis A_2 de la figure (A.3) modélise ce mode de fonctionnement et indique les états possibles du système et les événements déterminant les transitions entre ces états. Le p-automate associé à l'automate A_2 est donné dans la figure (A.4).

Depuis les modes 1 et 2 le système peut évoluer vers les modes $1'$, $2'$, 6 ou 7, en fonction de différentes défaillances des TPA qui peuvent se produire.

L'échec à la sollicitation de la partie hors-turbine (HT) de la TPA 1 peut conduire le système depuis l'état initial du Mode 1 vers le Mode $1'$ ou depuis l'état 6 du Mode 2 vers l'état 7 du Mode $1'$. L'automate à états finis $A_{1'}$ de la figure (A.5) modélise le Mode $1'$ et indique les états possibles du système et les événements déterminant les transitions entre ces états. Un p-automate a été associé à ce mode de fonctionnement et il est présenté dans la figure (A.6).

Le Mode $2'$ est construit de manière analogue au Mode $1'$ et il représente le mode de fonctionnement du système obtenu suite à l'échec à la sollicitation de la partie turbine hors-turbine (HT) de la TPA 2. L'automate à états finis $A_{2'}$ de la figure (A.7) indique les états possibles du

système dans ce mode de fonctionnement et les événements déterminant les transitions entre ses états. Un p-automate a été associé à ce mode de fonctionnement et il est présenté dans la figure (A.8).

Suite à l'échec à la sollicitation de la partie turbine (T) de la TPA 1 le système peut commuter dans le Mode 6 depuis l'état 3 du Mode 1 ou à l'état 7 du Mode 6 depuis l'état 8 du Mode 2. L'automate à états finis A_6 de la figure (A.9) modélise le Mode 6 et indique les états possibles du système et les événements déterminant les transitions entre ces états. Un p-automate a été associé à ce mode de fonctionnement et il est présenté dans la figure (A.10).

Le Mode 7 est construit de manière analogue au Mode 6 et il représente le mode de fonctionnement du système obtenu suite à l'échec à la sollicitation de la partie turbine (T) de la TPA 2. L'automate à états finis A_7 de la figure (A.11) indique les états possibles du système dans ce mode de fonctionnement et les événements déterminant les transitions entre ses états. Un p-automate a été associé à ce mode de fonctionnement et il est présenté dans la figure (A.12).

Conduite en puissance

Le Mode 5 représente la phase de conduite en puissance du système. Le système peut commuter vers ce mode de fonctionnement depuis l'état 8 de l'automate A_1 ou depuis l'état 8 de l'automate A_2. Le Mode 5 est représenté par l'automate à états finis A_5 (figure A.13) décrivant les états possibles du système dans ce mode et les événements déterminant les transitions entre ces états. Le p-automate associé à l'automate A_5 est donné dans la figure (A.14).

Conduite en baisse de puissance

Le Mode 8 représente la phase de conduite pour la baisse de la puissance du réacteur. Ce mode de fonctionnement est représenté par l'automate à états finis A_8 (figure A.15) décrivant les états possibles du système dans ce mode et les événements déterminant les transitions entre ces états. Le p-automate associé à l'automate A_8 est donné dans la figure (A.16).

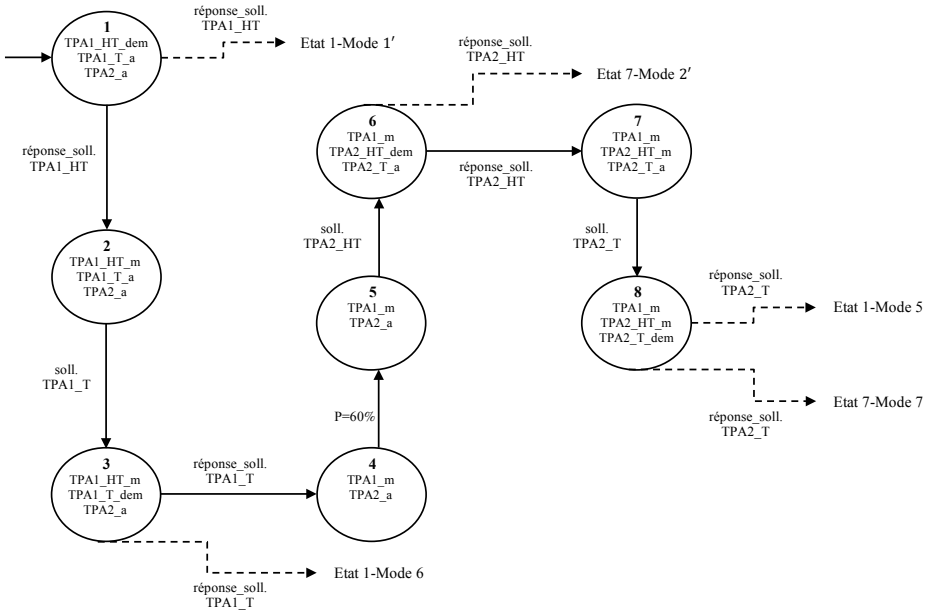


FIGURE A.1 – L'automate A_1 représentant l'utilisation de la TPA 1 pour le démarrage et la montée en puissance

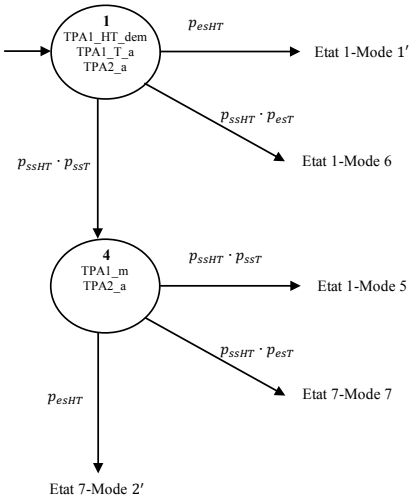


FIGURE A.2 – Version compacte du p-automate associé à l'automate A_1

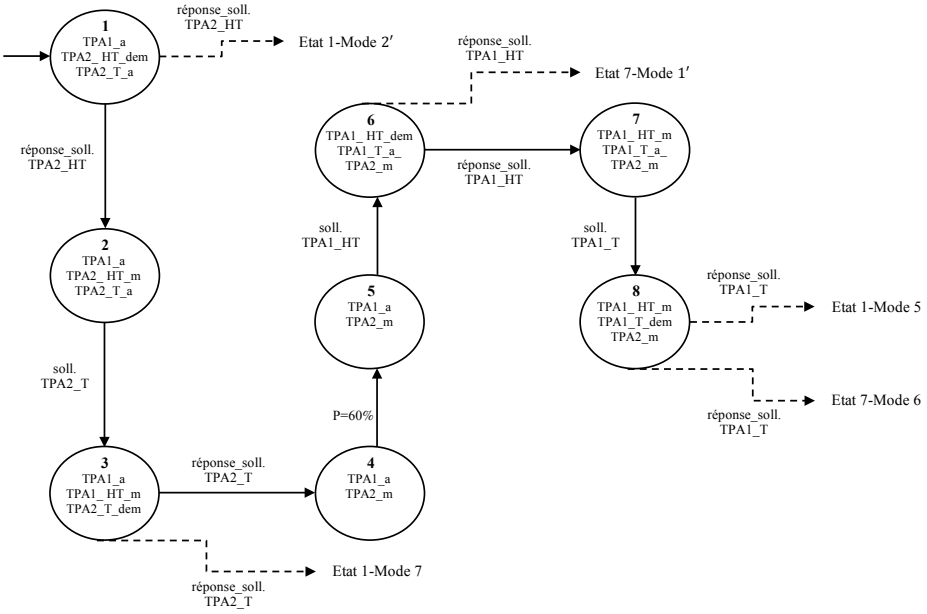


FIGURE A.3 – L'automate A_2 représentant l'utilisation de la TPA 2 pour le démarrage et la montée en puissance

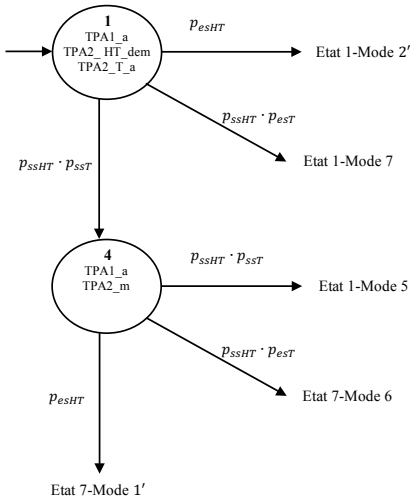


FIGURE A.4 – Version compacte du p-automate associé à l'automate A_2

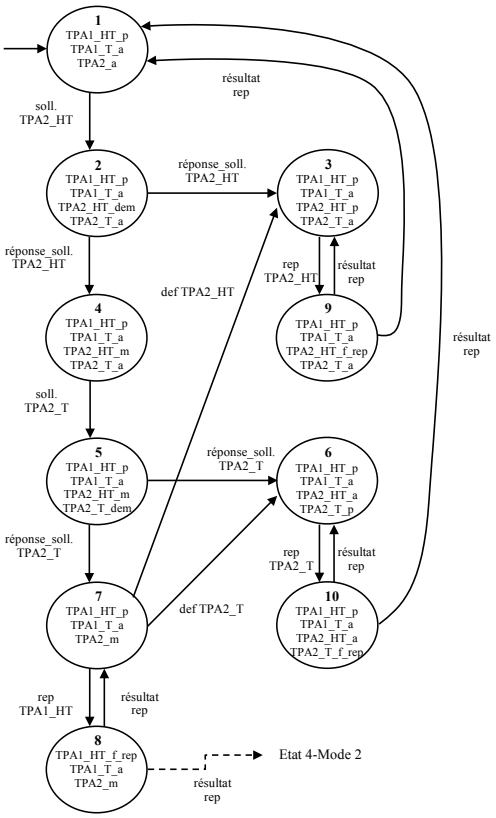


FIGURE A.5 – L'automate $A_{1'}$ obtenu suite à l'échec à la sollicitation de la partie hors-turbine (HT) de la TPA 1

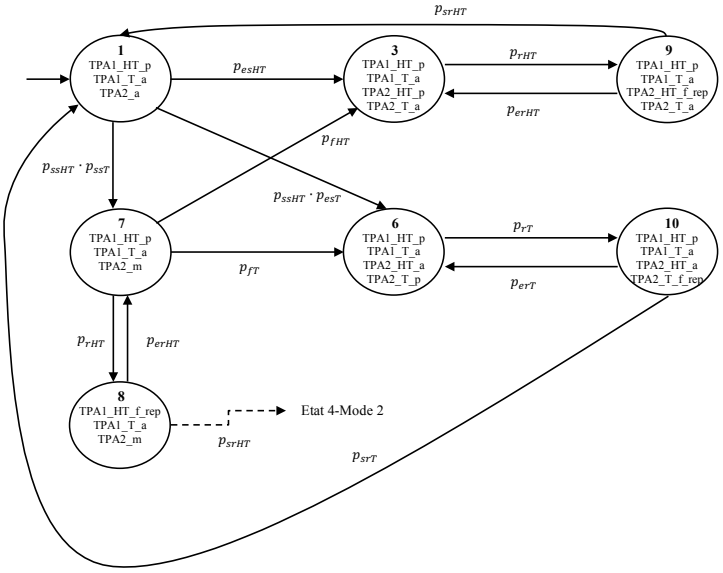


FIGURE A.6 – Version compacte du p-automate associé à l'automate $A_{1'}$

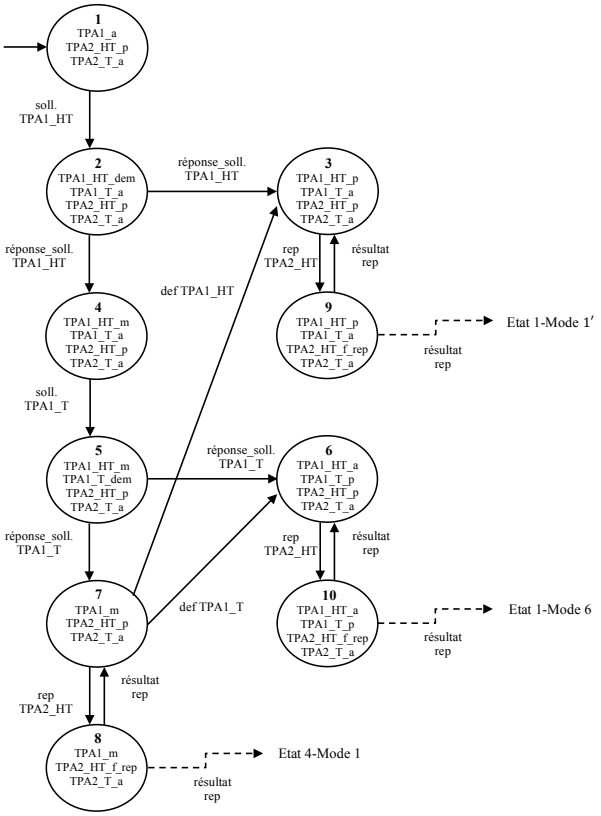


FIGURE A.7 – L'automate $A_{2'}$ obtenu suite à l'échec à la sollicitation de la partie hors-turbine (HT) de la TPA 2

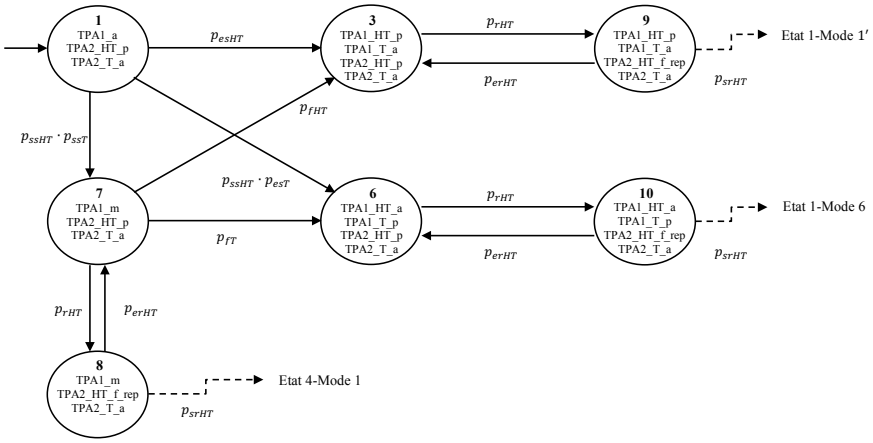


FIGURE A.8 – Version compacte du p-automate associé à l'automate $A_{2'}$

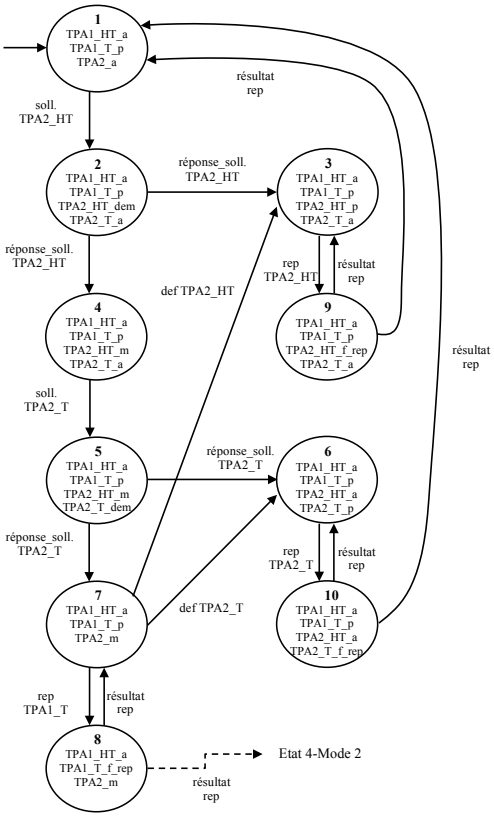


FIGURE A.9 – L'automate A_6 obtenu suite à l'échec à la sollicitation de la partie turbine (T) de la TPA 1

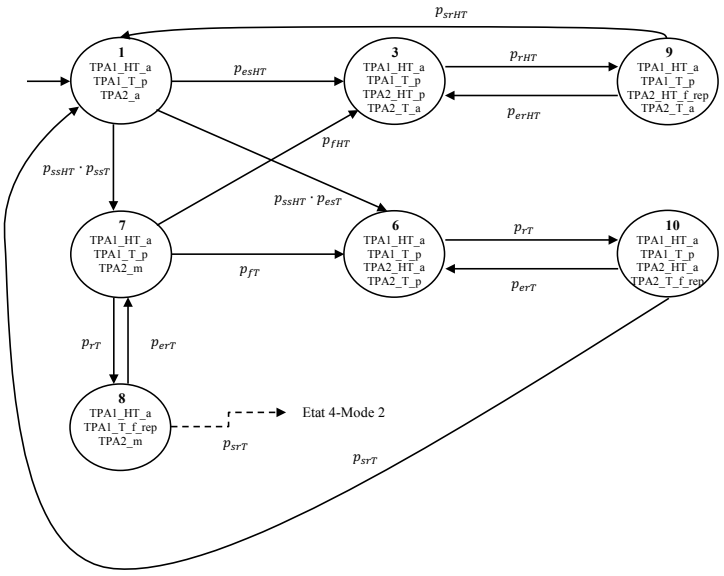


FIGURE A.10 – Version compacte du p-automate associé à l'automate A_6

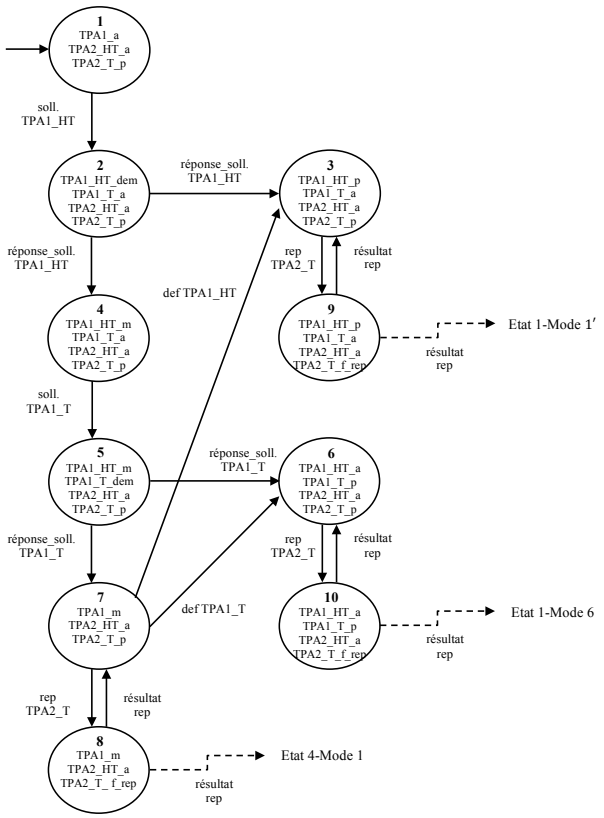


FIGURE A.11 – L'automate A_7 obtenu suite à l'échec à la sollicitation de la partie turbine (T) de la TPA 2

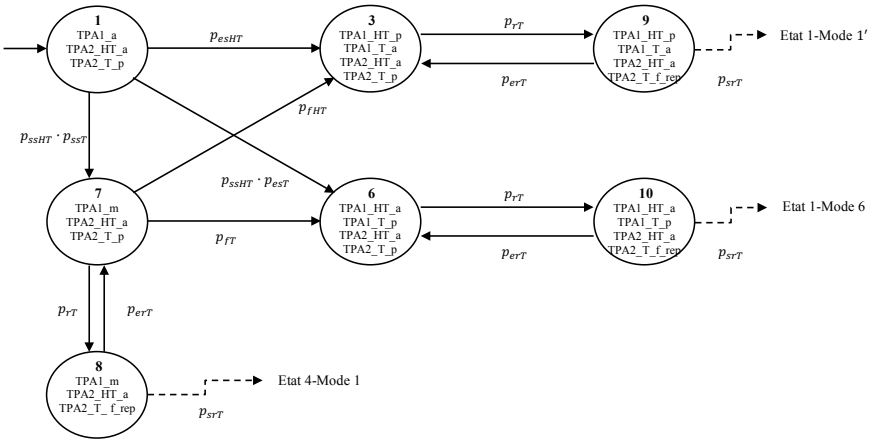


FIGURE A.12 – Version compacte du p-automate associé à l'automate A_7

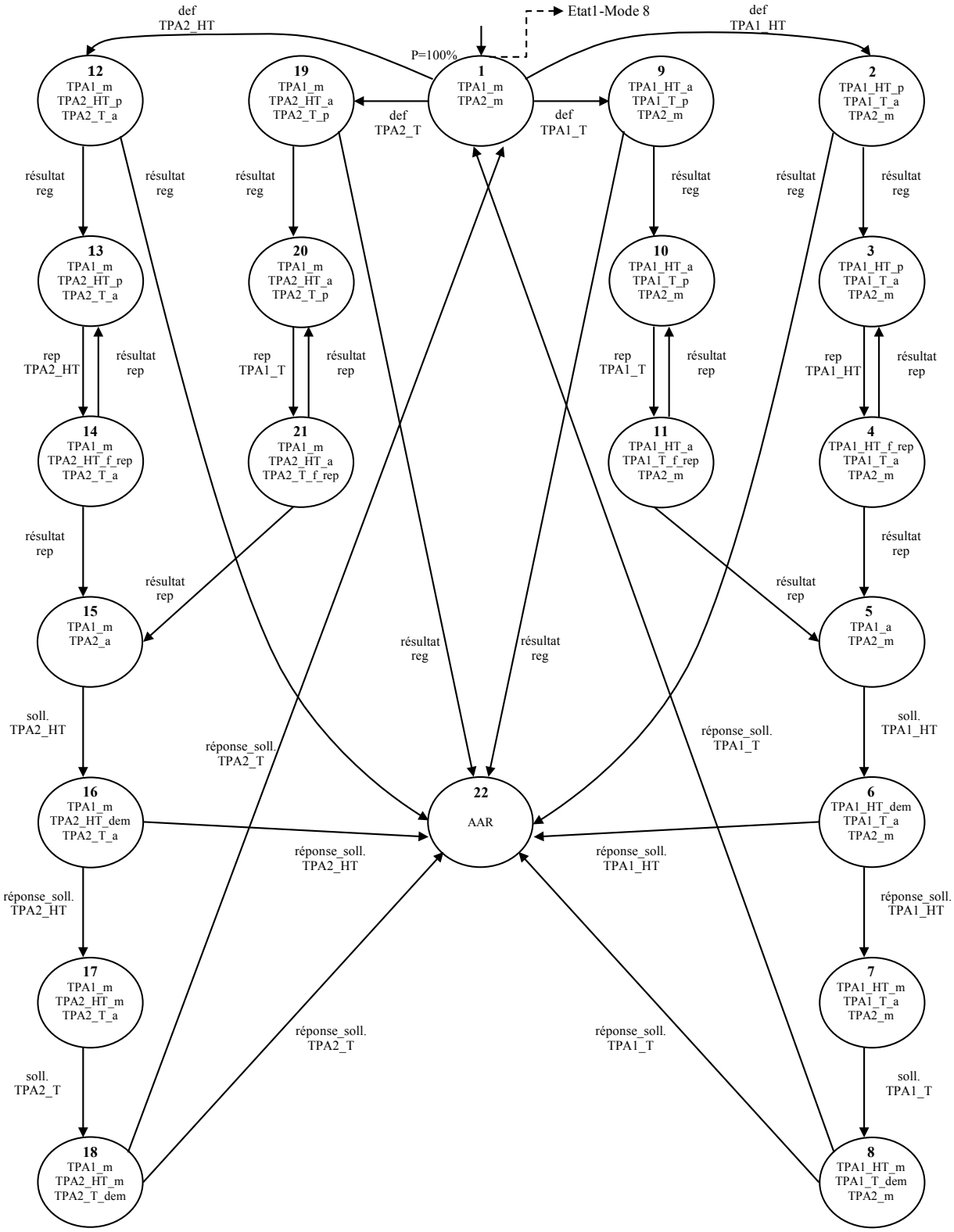


FIGURE A.13 – L'automate A_5 qui représente la phase de conduite en puissance

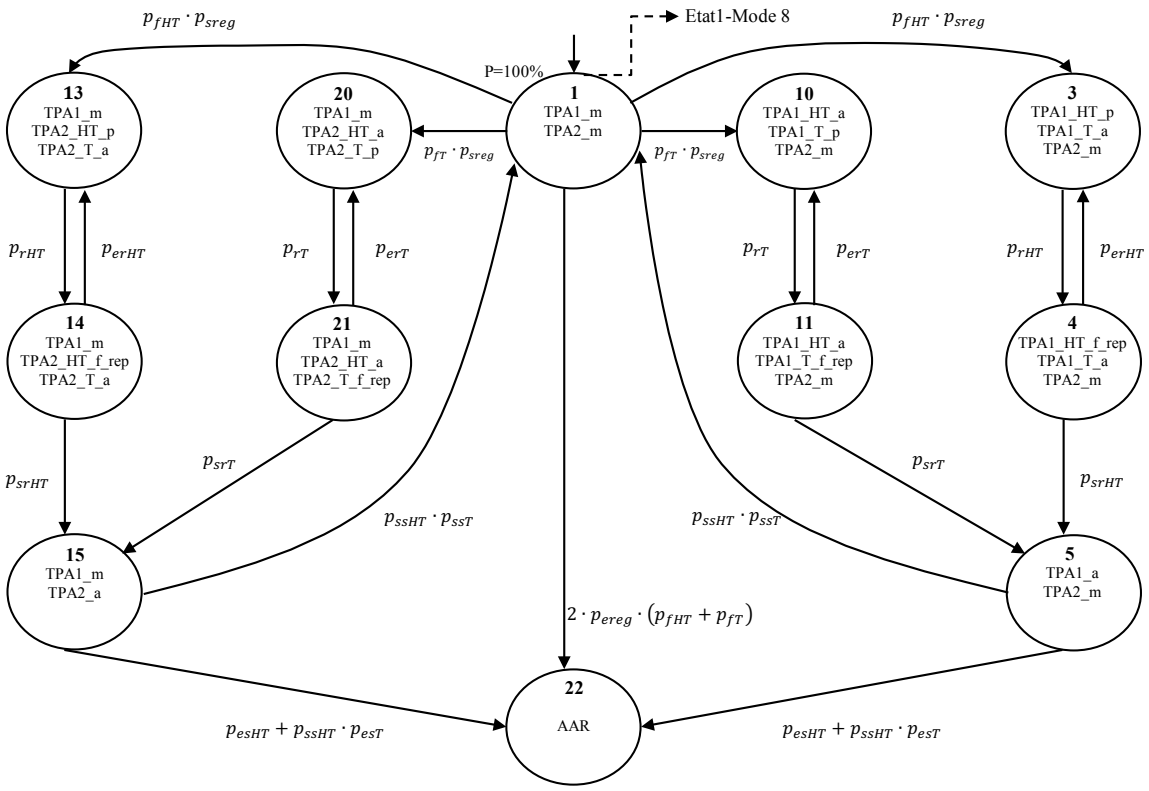


FIGURE A.14 – Version compacte du p-automate associé à l'automate A_5

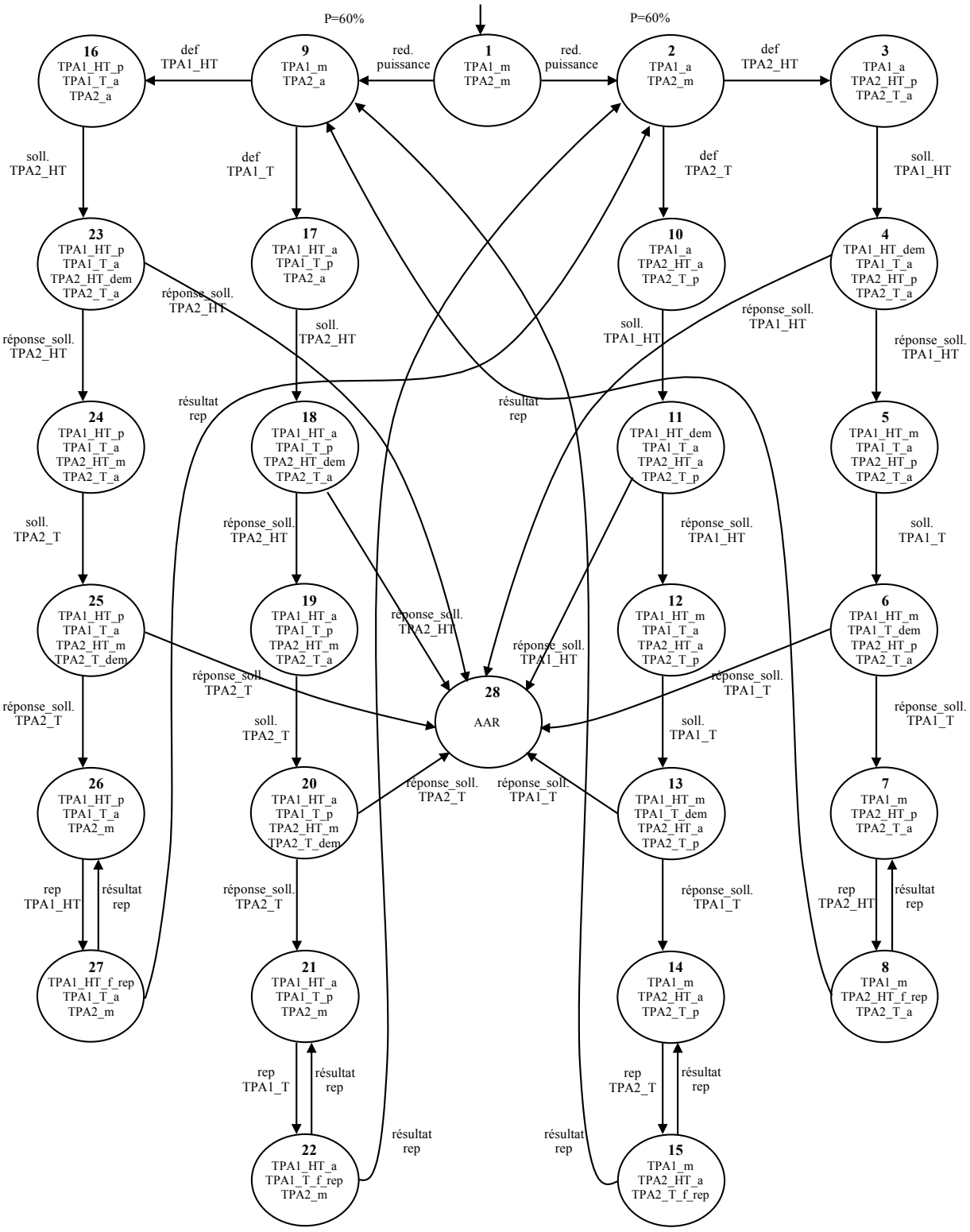


FIGURE A.15 – L'automate A_8 qui représente la phase de conduite en baisse de puissance

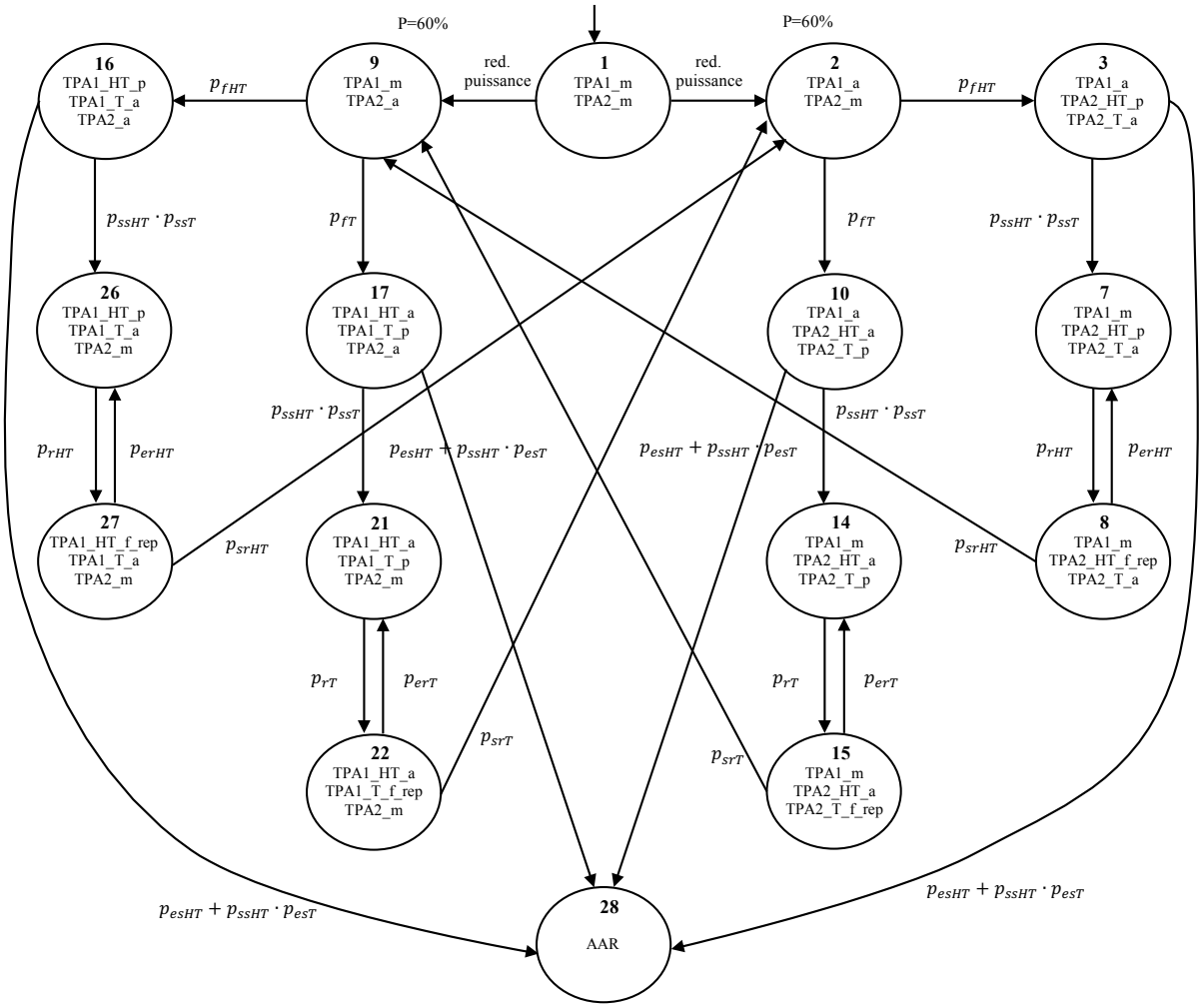


FIGURE A.16 – Version compacte du p-automate associé à l'automate A_8

Résumé

Les études de sûreté de fonctionnement (SdF) sont en général basées sur l'hypothèse d'indépendance des événements de défaillance et de réparation ainsi que sur l'analyse des coupes qui décrivent les sous-ensembles de composants entraînant la défaillance du système. Dans le cas des systèmes dynamiques pour lesquels l'ordre d'occurrence des événements a une incidence directe sur le comportement dysfonctionnel du système, il est important de privilégier l'utilisation de séquences d'événements permettant une évaluation des indicateurs de SdF plus précise que les coupes. Ainsi, nous avons proposé, dans une première partie de nos travaux, un cadre formel permettant la détermination des séquences d'événements qui décrivent l'évolution du système ainsi que leur évaluation quantitative, en recourant à la théorie de langages probabilistes et à la théorie des processus markoviens/semi-markoviens. L'évaluation quantitative des séquences intègrent le calcul de leur probabilité d'occurrence ainsi que leur criticité (coût et longueur des séquences). Pour l'évaluation des séquences décrivant l'évolution des systèmes complexes présentant plusieurs modes de fonctionnement ou de défaillance, une approche modulaire basée sur les opérateurs de composition (choix et concaténation) a été proposée. Celle-ci consiste à calculer la probabilité d'une séquence d'événements globale à partir d'évaluations réalisées localement, mode par mode. Les différentes contributions sont appliqués sur deux cas d'étude de taille et complexité croissante.

Mots-clés: langages probabilistes, sûreté de fonctionnement, chaînes de Markov, séquences d'événements, analyse de criticité, approche modulaire.

Abstract

Dependability studies are often based on the assumption of events (failures and repairs) independence but also on the analyse of cut-set which describes the subsets of components causing a system failure. In the case of dynamic systems where the events occurrence order has a direct impact on the dysfunctional behaviour, it is important to promote using event sequences instead of cut-sets for dependability assessment. In the first part, a formal framework is proposed. It helps in determining sequences of events that describe the evolution of the system and their assessment, using the theory of probabilistic languages and the theory of Markov/semi-Markov processes. The assessment integrates the calculation of the probability occurrence of the event sequences and their criticality (cost and length). For the assessment of complex systems with multiple operating/failure modes, a modular approach based on composition operators (choice and concatenation) is proposed. Evaluation of the probability of a global sequence of events is peformed from local Markov/semi-Markov models for each mode of the system. The differents contributions are applied on two case studies with a growing complexity.

Keywords: probabilistic languages, dependability, Markov chains, events sequences, criticality analysis, modular approach.