Evaluation de la sûreté de systèmes dynamiques hybrides complexes : application aux systèmes hydrauliques

Perrine Broy

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Perrine BROY

le 12 mars 2014

Evaluation de la sûreté de systèmes dynamiques hybrides complexes. Application aux systèmes hydrauliques

JURY

M. F. PÉRÈS	PROFESSEUR DES UNIVERSITES	Président
M. C. BERENGUER	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. N. BRINZEI	MAITRE DE CONFERENCES	Examinateur
M. M. CEPIN	PROFESSOR	Rapporteur
M. H. CHRAIBI	INGENIEUR CHERCHEUR	Directeur de thèse
M. Y. DIJOUX	MAITRE DE CONFERENCES	Examinateur
M. J.-M. THIRIET	PROFESSEUR DES UNIVERSITES	Rapporteur

Personnalité invitée

M. R. DONAT	INGENIEUR CHERCHEUR

Remerciements

Faire une thèse CIFRE, c'est bénéficier d'un double encadrement, académique et industriel. Mais lorsque chacun de ces encadrements est assuré par deux personnes, toutes compétentes et sympathiques, cela fait beaucoup de remerciements à rédiger, et personne ne m'a dit si j'avais le droit de quadrupler le nombre de pages de remerciements ou d'occurrences du mot « merci »...

Mes premiers remerciements vont donc à mon « comité encadrant » constitué de Christophe Bérenguer, Yann Dijoux, Hassane Chraïbi et Roland Donat. Christophe, merci d'avoir accepté la direction de cette thèse, merci pour ta disponibilité sans faille et pour ton suivi malgré l'éloignement. Yann, merci de m'avoir aidée lors des calculs analytiques douloureux, mais aussi pour la découverte d'un bon nombre de restaurants troyens ! Hassane, merci pour ta patience et ta gentillesse ; l'initiation au fonctionnement des évacuateurs de crues ou à la structure de PyCATSHOO n'était pas une affaire gagnée d'avance. Roland, merci de m'avoir guidée et motivée si souvent ! Merci à vous quatre, j'ai beaucoup appris à vos côtés et cela a été un réel plaisir de travailler avec vous !

Je remercie l'ensemble des membres du jury pour leur participation à ma soutenance et pour l'intérêt porté à mes travaux de recherche. Je remercie tout particulièrement François Pérès d'avoir endossé le rôle de président du jury. Je tiens à exprimer ma reconnaissance à Marko Cepin et Jean-Marc Thiriet pour m'avoir fait l'honneur d'être rapporteurs de ces travaux et pour leurs questions constructives. Je tiens également à remercier Nicolae Brinzei pour sa minutieuse relecture.

Alors que j'étais encore élève-ingénieur, j'ai longtemps muri ce projet de thèse avant de m'y lancer. Je voudrai remercier Frédéric et Anick de m'avoir encouragée à partir dans cette voie qu'est la recherche, et Hermann, Marine, Fabiano, Fatiha, Paul, Geoffrey, Lise et William de m'avoir fait part de leurs expériences de doctorants lors de discussions enrichissantes. Grâce à vous, j'ai construit ce projet de thèse en connaissance de cause.

Une fois ma décision prise, c'est une autre équipe qui m'a permis de concrétiser ce projet. Je tiens à remercier l'équipe hiérarchique du département MRI d'avoir initié et prolongé cette thèse mais aussi de m'avoir permis de faire de belles conférences. Quitte à être au sein du département MRI, je vais continuer en remerciant les chercheurs qui le constituent pour leur accueil, leurs conseils et leur sympathie. Je ne me risquerai pas à

vous nommer pour n'oublier personne, mais j'ai été heureuse de vous côtoyer le temps d'un café, à la cantine ou dans les vestiaires de la gym. La bonne humeur ambiante a égayé mes travaux et je vous remercie pour vos encouragements tout au long de ces trois années.

Mes remerciement vont devenir encore plus locaux avec une pensée à tous ceux qui ont eu la « chance » de partager mon bureau : Linh, Carine, Pierre-Yves, Antonello, Stéphanie, Nicolas, Martin, Tazio : merci d'avoir supporté mes bavardages et mes bougonnements !

Je tiens à remercier les membres du LM2S pour leur acceuil chaleureux à chacune de mes venues troyennes.

En parallèle de ces trois ans de recherche, j'ai eu l'occasion de progresser en course à pieds, danse et autres renforcements musculaires. Mention spéciale à tous les coachs qui m'ont permis de me défouler et de décompresser !

Merci à Jane-Marie, bonne fée viroflaysienne, de m'avoir menée à bon port chaque matin !

Bon courage à mes compatriotes de thèse, vous verrez, on finit toujours par y arriver, la preuve !

Mes derniers remerciements vont à ma famille, ma belle-famille et à mes proches. Il est temps de vous remercier d'avoir accepté soit de me voir si rarement, soit de me voir squatter la table ronde pour de longues séances de débuggage et rédaction. Même le chat Peluche en avait pris son parti et m'encourageait par quelques ronronnements...

Je ne serai pas là où j'en suis sans mes parents : merci pour votre amour, votre confiance sans faille, votre relecture de ce manuscrit. Merci aussi à toi Maxime !

Tout ceci ne serait rien sans Gaël. Tu m'as suivie au bout du monde lors des conférences, tu m'as épousée avant même de savoir à quoi ressemblait une troisième année de thèse, tu as toujours répondu placidement « oui... » à chaque fois que je te demandais « Est-ce que tu crois que je vais y arriver ? » dans mes périodes de doute, tu m'as littéralement entretenue pendant le mois de décembre, tu as scrupuleusement relu ce manuscrit et tu as géré mes pots de thèse comme un roi... Pour tout ça, le nom « Hesters » aurait bien mérité de figurer sur la première page de ce manuscrit. A défaut, je tenais à ce que les derniers mots de cette page soit pour toi... Mille mercis donc.

Table des matières

Table des figures

Principaux acronymes

I_B Indicateur de Birnbaum

TFN Taux de Faux Négatifs

SSA Seuil de Sûreté Atteint

ARSHY Analyse des Risques des Systèmes HYdrauliques

ASH Automate Stochastique Hybride

BdC Base de Connaissances

CCl Contrôle-Commande local

CPDMP *Communicating PDMP*

EdC Évacuateur de Crues

EPS Étude Probabiliste de Sûreté

ER Événement Redouté

FR Fil Rouge

GASPART *GAted Spillway System - Probabilistic Assessment of Reliability Tool*

MRI Management des Risques Industriels

PDMP *Piecewise Deterministic Markov Process*

PyCATSHOO PythoniC AuTomates Stochastiques Hybrides Orientés Objets

RMB *Receiving Message Box*

SDH Système Dynamique Hybride

SMB *Sending Message Box*

SVM *Support Vector Machine*

TTF *Time To Failure*

VTTF Vecteur de TTF

Introduction générale

L'utilisation d'eau par des aménagements hydrauliques fournit une énergie propre et renouvelable. L'hydroélectricité représente la deuxième source de production d'électricité en France en 2012. EDF est exploitant de 435 centrales hydroélectriques. A ce titre, il participe au programme de rénovation et de modernisation du parc hydraulique. Afin d'améliorer la sûreté des ouvrages hydrauliques, des études de danger sont réalisées en confrontant leur dimensionnement à des crues exceptionnelles ou au dysfonctionnement des évacuateurs de crues vannés.

En cas de crue, il est nécessaire d'évacuer le volume d'eau déversé en amont du dispositif afin de maintenir le plan d'eau de la retenue sous un niveau acceptable. Les évacuateurs de crues (EdC) sont les structures dédiées au déversement des eaux en excédent. Pour cela, les EdC vannés requièrent la mobilisation de vannes. L'événement redouté (ER) est réalisé lorsqu'un seuil de sûreté est atteint par le niveau de la retenue. L'étude de la sûreté des EdC se traduit par des indications sur la fiabilité de ces dispositifs. Par exemple, les EdC sont hiérarchisés vis-à-vis du risque lié à l'ER, ou des leviers d'amélioration de la sûreté sont proposés, tels que des stratégies de maintenance.

Au sein du département Management des Risques Industriels (MRI) d'EDF R&D, le projet ARSHY (Analyse des Risques des Systèmes HYdrauliques) développe des méthodologies d'analyse de risque systèmes pour le parc hydraulique d'EDF. En particulier, la méthode d'évaluation de la fiabilité des EdC vannés est consolidée par la prise en compte du facteur temps. La dynamique du processus de crue et de son évacuation est telle que l'évolution physique et déterministe du niveau d'eau dans la retenue est intimement liée aux événements discrets aléatoires qui vont affecter l'ouverture des vannes. En ce sens, les EdC sont des Systèmes Dynamiques Hybrides (SDH) et rentrent dans le cadre de la fiabilité dynamique. La prise en compte de l'information temporelle est corrélée à l'introduction d'une variable déterministe continue dans le processus stochastique.

A travers les EdC, ce sont donc les SDH qui sont concernés par la problématique : comment estimer la sûreté des EdC ? Nous proposons une méthodologie qui accompagne l'utilisateur tout au long de la modélisation et de l'exploitation des résultats, pour des SDH de taille industrielle. Les EdC constituent un support et une illustration pour ces travaux mais la méthodologie proposée est adaptable au cadre général de la fiabilité dynamique.

En fiabilité dynamique, une classe de processus est généralement utilisée pour modéliser les SDH. Il s'agit des Processus de Markov Déterministes par Morceaux (PDMP). Cette modélisation prend en compte la dynamique induite par la dépendance au temps du fonctionnement de ce type de système. En accord avec ce cadre théorique, les Automates Stochastiques Hybrides (ASH) distribués présentent le double avantage d'un formalisme riche et d'une représentation graphique intuitive et flexible pour décrire des systèmes complexes. La complexité des EdC est due au nombre élevé de composants, aux interactions composant-composant et composant-environnement, et à l'évolution simultanée de l'état du système et de la variable déterministe continue en fonction de l'état des composants. Les ASH sont ensuite associés à la simulation de Monte Carlo pour la quantification probabiliste de la fiabilité.

La méthode d'évaluation de la fiabilité des EdC vannés, nommée GASPART (de l'anglais *Gated Spillway System - Probabilistic Assessment of Reliability Tool*) est associée à un outil du même nom. Cet outil est développé à partir d'un langage initialement conçu pour traiter les systèmes à états discrets. La prise en compte des phénomènes continus et transitoires n'est possible qu'au prix d'hypothèses de modélisation conservatives et en adoptant des méthodes simplifiées de résolution des équations différentielles. Par ailleurs, GASPART possède deux modules de quantification distincts. La conception d'un nouvel outil nommé PyCATSHOO écarte les limites identifiées de l'outil GASPART. En étant dédié à l'évaluation de la fiabilité des SDH dès sa conception, PyCATSHOO lève les hypothèses de modélisation conservatives et propose une démarche capable de caractériser les résultats en conservant l'information temporelle.

Il est important d'identifier et de quantifier l'intérêt de prendre en compte de nouvelles informations temporelles dans l'évaluation des performances fiabilistes. En effet, la débitance des vannes dépend de la hauteur de leur ouverture. La position d'une vanne dépend elle-même de la progression du processus d'ouverture, interrompue ou non par la défaillance d'un composant nécessaire à ce processus. Une défaillance précoce entraîne une débitance faible susceptible de provoquer l'événement redouté. Contrairement à un problème de fiabilité classique, l'état du système n'est pas une fonction de l'état de ses n composants. Ce sont les dates de défaillance en fonctionnement (T_1, ..., T_n) qui définissent l'occurrence ou non de l'événement redouté pour le système. Cette dépendance est effective par le biais d'une fonction $f(T_1, ..., T_n) = s$ où $s \in \{ER, \overline{ER}\}$ désigne l'état du système (occurrence ou non de l'événement redouté ER). Ainsi, pour un même ensemble de composants en panne, le système peut être en panne ou en marche, selon les dates de panne des composants. Certaines notions de sûreté comme celles de coupes ou de mesures d'importance sont à redéfinir. Ces notions doivent être adaptées aux SDH dans l'objectif d'identifier la fonction f et de caractériser les histoires de défaillances. Cette démarche est rarement associée à la fiabilité dynamique, aussi les indicateurs proposés dans cette thèse sont-ils innovants.

Les différentes étapes de cette méthodologie, exposées ci-dessous, permettent la modélisation du système puis l'exploitation des résultats obtenus.

1. La compréhension du fonctionnement du système implique la décomposition des

sous-systèmes, l'identification des composants similaires par classes et la définition d'hypothèses de modélisation.

2. Chaque classe de composants est décrite par un automate dont les états sont les différentes phases de son fonctionnement. Les transitions entre ces états sont caractérisées par un ensemble de conditions.
3. Une Base de Connaissances (BdC) répertorie les classes ainsi définies. Les boîtes à messages destinataires et expéditrices participent à la synchronisation des automates, assurant la communication entre les objets qui interagissent. Le dispositif de calcul de la variable continue fait partie de cette construction. L'élaboration d'une BdC doit être suffisamment générale pour représenter plusieurs systèmes d'une même catégorie, mais suffisamment détaillée pour être proche de la réalité.
4. Les informations spécifiques à la topologie d'un système en particulier sont regroupées dans un script principal. Chaque composant y est déclaré en tant qu'instance d'une classe PyCATSHOO. Ces objets sont ensuite reliés entre eux par des liens, qui matérialisent les boîtes à messages. L'exécution de ce script génère aléatoirement des simulations.
5. L'analyse des résultats fournit des indicateurs de fiabilité classique, tels que l'évolution de la probabilité d'occurrence de l'ER par rapport au temps. Les combinaisons d'événements les plus contributeurs dans la réalisation de l'ER sont également identifiées.
6. Le fruit des simulations est une liste d'histoires. Une histoire est la séquence des états visités par chaque automate le temps d'une crue, associés à la date de chacune de ces transitions. Nous proposons une méthode pour extraire, synthétiser et utiliser l'information issue de la simulation du modèle. La séparation des histoires par rapport à l'occurrence ou non de l'événement redouté, en fonction des durées de fonctionnement avant défaillance de chaque composant du système, est un modèle qui exploite au maximum les données temporelles contenues dans les histoires simulées. Cette classification pronostique, à partir d'un jeu de nouvelles durées de fonctionnement avant défaillance, l'issue de l'histoire associée.
7. L'estimation de l'importance dynamique permet de savoir à tout instant quel est le composant dont la défaillance à cet instant précis aurait le plus d'impact sur la probabilité de l'ER, par rapport à une situation de référence. Cette définition est généralisable à un groupe de composants.

Ce mémoire est structuré en six chapitres :

- Le chapitre 1 positionne le problème industriel et les travaux de thèse.
- Le chapitre 2 confronte un état de l'art des méthodes utilisées en fiabilité dynamique aux enjeux méthodologiques impliqués par le problème industriel.
- Le chapitre 3 détaille les différents outils de modélisation pour la sûreté de fonctionnement.
- Le chapitre 4 décrit le fonctionnement des deux EdC étudiés et présente la modélisation de ces systèmes par les ASH distribués et l'élaboration de la BdC qui en découle.

– Le chapitre 5 propose une démarche prévisionnelle fondée sur la classification des histoires et l'estimation de la probabilité d'occurrence de l'ER.
– Le chapitre 6 définit l'importance dynamique comme un indicateur de fiabilité dynamique destiné à l'aide à la décision.

Première partie

De la problématique industrielle aux enjeux méthodologiques

Chapitre 1

Position du problème et motivation industrielle

Ce chapitre positionne le problème industriel, et par conséquent, ces travaux de thèse. La section 1.1 introduit la motivation de cette thèse, c'est-à-dire l'évaluation de la sûreté de fonctionnement des évacuateurs de crues. Ces systèmes hydrauliques ont la particularité de dépendre d'événements aléatoires discrets, mais aussi de l'évolution d'une variable déterministe continue. A ce titre, ce sont des systèmes dynamiques hybrides. A travers les évacuateurs de crues, ce sont donc tous les systèmes dynamiques hybrides qui sont concernés par la problématique : comment évaluer la sûreté de fonctionnement des évacuateurs de crues ?

La section 1.2 énumère ensuite les enjeux de la thèse en dressant le « cahier des charges » de la méthodologie recherchée.

1.1 Problématique industrielle : les évacuateurs de crues et l'estimation de leur sûreté de fonctionnement

Cette section a pour but de présenter la problématique industrielle de la thèse. Après une brève présentation de l'hydroélectricité (section 1.1.1) et une introduction sur le rôle des évacuateurs de crues (section 1.1.2), la section 1.1.3 résume la structure des évacuateurs de crues. Puis la section 1.1.4 dresse un rapide état de l'art de la sûreté de fonctionnement dans le domaine de l'hydraulique. Finalement, la section 1.1.5 introduit quelques notions d'hydrologie et d'hydraulique, notamment sur les hydrogrammes des crues et sur les cotes de la retenue d'un barrage.

1.1.1 L'hydroélectricité en France

En France, l'hydroélectricité est l'une des principales énergies [EDF, 2011]. L'hydraulique, qui représente 11,7% de l'énergie électrique totale produite en France en 2012, y est ainsi la deuxième source de production d'électricité. L'utilisation du potentiel de l'eau par des aménagements hydrauliques fournit un double avantage. D'une part, c'est une énergie propre et renouvelable, sans impact sur le climat car elle émet très peu de gaz à effet de serre. D'autre part, à défaut de savoir stocker l'électricité, c'est un moyen écologique et économique de répondre rapidement aux variations de la consommation d'électricité.

En France métropolitaine, EDF exploite 435 centrales hydroélectriques. L'eau retenue derrière un barrage est amenée par une conduite forcée vers une turbine. La force de l'eau fait tourner la turbine qui entraîne à son tour un alternateur, générant ainsi un courant électrique alternatif. Le transformateur élève ensuite la tension pour faciliter le transport de l'électricité sur de longues distances [EDF, 2011].

1.1.2 Rôle des évacuateurs de crues

En cas de crue, le volume d'eau déversé en amont de la retenue peut provoquer une montée d'eau incompatible avec la capacité de stockage et d'absorption du dispositif de production hydroélectrique recevant cette eau. Le volume d'eau en excédent représente une menace pour la sécurité du barrage et il est nécessaire de l'évacuer afin de maintenir le plan d'eau de la retenue (bassin amont du barrage) sous un niveau acceptable et d'éviter la submersion de la digue. Aussi les barrages sont-ils dotés de structures dédiées au déversement des eaux en excédent. Ces dispositifs, illustrés par la figure 1.1, sont appelés **évacuateurs de crues** (EdC). Certains EdC fonctionnent par déversement naturel lorsque le niveau d'eau dépasse celui du réservoir. Ce sont des EdC passifs. En revanche, les EdC vannés requièrent la mobilisation de vannes pour déverser le volume d'eau excédentaire.

Les évacuateurs de crues vannés sont au centre de cette thèse. L'objectif de ces travaux est de modéliser les EdC et de les simuler du point de vue fiabiliste, puis d'exploiter les résultats obtenus pour l'évaluation de la sûreté de fonctionnement et la prise de décision. L'exploitation des résultats prend la forme d'indications sur la fiabilité de ces dispositifs pour donner des pistes d'amélioration de la sûreté.

L'événement redouté (ER) est défini par l'atteinte du seuil maximal par le niveau de la retenue. Cet événement indésirable sera par la suite également nommé « débordement » ou OF pour « *OverFlow* ».

FIGURE 1.1 – Évacuateur de crues. Crédit photo : EDF, Patrice Dhumes

1.1.3 Composition d'un évacuateur de crues

Un évacuateur de crues est constitué d'une installation hydromécanique, d'un contrôle-commande et d'un système d'alimentation électrique, comme l'illustre la figure 1.2. Un ou plusieurs opérateurs, alertés si besoin par un dispositif d'alarme, contrôlent le reste de l'aménagement. Par le biais d'un contrôle-commande (local ou non), l'opérateur sollicite un actionneur (moteur ou pompe). Cet actionneur déclenche le mouvement de la transmission (vérin, chaîne, crémaillère, etc.). Cette transmission communique ensuite ce mouvement aux vannes. Une passe est constituée du dispositif actionneur - transmission - vanne.

Cette description est représentée schématiquement sur la figure 1.2 et s'applique à tous les évacuateurs de crues. Des subtilités apparaissent dès que l'on considère des évacuateurs différents ou que l'on rentre dans le détail de ces sous-systèmes. Chaque composant peut tomber en panne à la sollicitation ou en fonctionnement. La plupart du temps, la structure d'un système est telle que ce composant est relayé en cas de panne. Ces redondances forment un réseau d'interactions complexes à modéliser. La réaction de l'opérateur à ces pannes se traduit par la recherche du composant de secours et la sollicitation de celui-ci. Ce type d'action de l'opérateur est associé à une probabilité d'échec. Inversement, ces actions sont parfois automatisées.

L'objectif industriel de cette thèse est de proposer une méthodologie pour la modélisation des EdC. Cette modélisation devra représenter le processus de crue, le rôle de chaque composant en cas de défaillance et les réactions de l'opérateur. L'élaboration d'un modèle suffisamment général pour représenter différents évacuateurs, mais suffi-

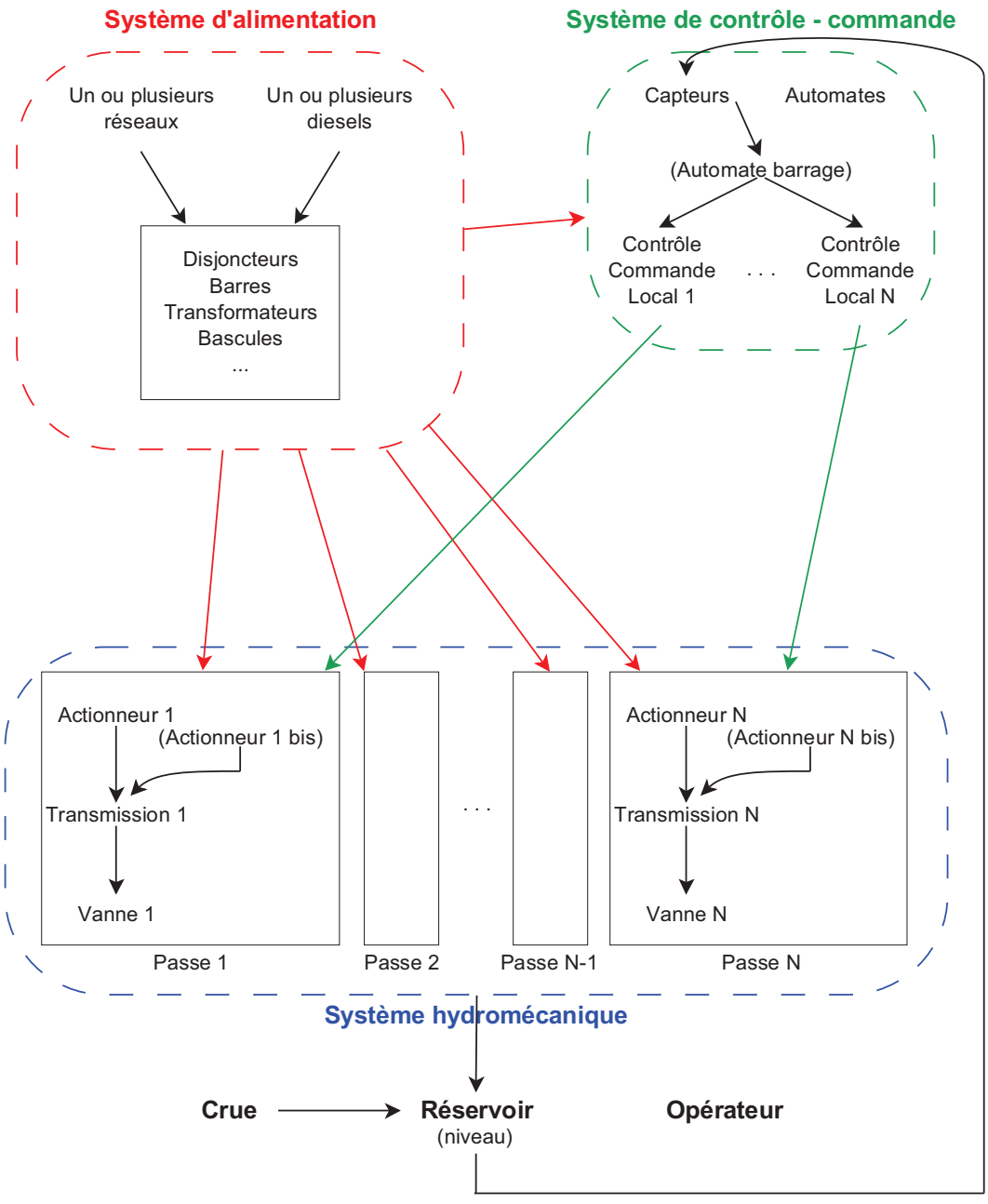

FIGURE 1.2 – Représentation schématique d'un évacuateur de crues

samment détaillé pour être réaliste, est un premier défi. La taille du système (environ 70 composants) en est un deuxième. Mais c'est un troisième verrou qui sera le principal enjeu de cette thèse : celui de la prise en compte complète de l'information temporelle tout au long de ce processus de modélisation, mais aussi lors de l'exploitation des résultats. Cet enjeu est exposé dans la section 1.2. Les caractéristiques d'un EdC sont détaillées plus précisément dans le chapitre 4.

1.1.4 La sûreté de fonctionnement dans l'hydraulique

Cette section répertorie différentes communications publiées au sujet de la sûreté de fonctionnement dans le domaine hydraulique. Le but est de positionner nos travaux par rapport aux recherches menées en dehors d'EDF R&D.

Dans le domaine hydraulique, la plupart des études [Hartford et Baecher, 2004], [Barker *et al.*, 2006] reposent sur les arbres d'événements (définis dans la section 2.1.1.5), les arbres de défaillances (introduits dans la section 2.1.1.5) ou les méthodes de type AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité). Quelques publications se basent sur les diagrammes d'influence ou réseau bayésien (définis dans le paragraphe **??**) [Hartford et Baecher, 2004], [Smith, 2006]. Dans [Estes *et al.*, 2005], l'US Army Corps évalue le risque de défaillance pour les évacuateurs de crues vannés en utilisant la méthode statique des index de condition (*Condition Indexing* ou CI). Le CI d'un composant est un pourcentage qui diminue si le composant est dégradé. Une formule relie le CI au taux de défaillance d'un composant. Le CI du système global est calculé comme une combinaison linéaire des CI de ses sous-systèmes, où les coefficients sont donnés par une mesure d'importance de chaque sous-système. Le calcul de ces facteurs d'importance statiques n'est pas précisé dans [Estes *et al.*, 2005] et n'a pas de rapport avec les mesures d'importance dynamique estimées dans le chapitre 6 de ces travaux de thèse.

Selon les objectifs visés par l'étude, chaque modèle possède des paramètres très différents. Par exemple, [Smith, 2006], [Barker *et al.*, 2006] tiennent compte de l'érosion et de la force du vent. Étant donné que nous nous intéressons uniquement à la période de crue, ce type de paramètre ne sera pas retenu dans notre modèle.

Dans ses travaux, [Parent, 1991] fait reposer l'aspect aléatoire sur le débit entrant, et non sur le fonctionnement des composants : le réservoir est considéré comme fiable et agrégé en une seule entité. La thèse de [Parent, 1991] répond donc à une problématique différente de la nôtre.

Dans [Barker *et al.*, 2006], les auteurs ont des objectifs plus proches des nôtres. Leurs travaux concernent un évacuateur de crues, et les paramètres pris en compte sont comparables aux nôtres, notamment ceux relatifs à la fiabilité humaine ou à l'architecture globale du système. Cependant, la plupart des groupes de composants sont agrégés en sous-systèmes, et le temps n'est pas pris en compte.

En considérant le nombre réel de composants d'un évacuateur de crues, et en y intégrant la composante dynamique, notre approche devient innovante dans le domaine de l'hydraulique. La prise en compte complète du facteur temps pendant l'exploitation des résultats est une nouveauté dans le cadre de la fiabilité dynamique, *a fortiori* dans celui de la sûreté hydraulique.

1.1.5 Quelques notions d'hydrologie et d'hydraulique

Les crues susceptibles d'affecter l'aménagement hydraulique étudié sont identifiées par leur temps de retour. Par exemple, les crues considérées dans ces travaux sont des crues décennales, centennales et millénales, de fréquences annuelles respectives 1/10, 1/100 et 1/1000.

1.1.5.1 Origine des hydrogrammes des crues

Dans la suite du manuscrit, chaque crue sera caractérisée par un hydrogramme. Un hydrogramme est un nuage de points représentant le débit entrant, généralement en $m^3.s^{-1}$, dans le réservoir du barrage en fonction du temps, comme l'illustre la figure 1.3. Les hydrogrammes disponibles sont ceux des crues décennales, centennales et millénales ; ils ont été élaborés par la DPIH (Division Production et Ingénierie Hydraulique) d'EDF grâce à la méthode SCHADEX [Garavaglia, 2011].

Un modèle hydrologique, ou modèle pluie-débit, utilise des données sur les précipitations (la pluie) pour prédire le débit en aval d'un bassin versant. D'autres paramètres y sont intégrés à propos des interactions avec l'atmosphère ou les nappes souterraines. Il existe plusieurs classifications des modèles pluie-débit [Gnouma, 2006]. Un modèle pluie-débit peut être global (le bassin est considéré comme une entité unique) ou distribué. Il peut être déterministe, stochastique ou mixte. Il existe des modèles événementiels, activés seulement au moment des pluies, et des modèles continus. Enfin, des modèles sont empiriques et d'autres sont conceptuels. Un modèle empirique est construit autour de relations mathématiques directes établies entre les entrées et les sorties observées sur le bassin versant considéré. Les modèles conceptuels cherchent à représenter les principaux processus de la relation pluie-débit mais peuvent s'affranchir de certaines difficultés en introduisant des paramètres qui ne représentent pas nécessairement des variables physiques.

La méthode SCHADEX, qui a fait l'objet de nombreux travaux de validation [Garavaglia, 2011], a la particularité de coupler un modèle probabiliste de pluie et un modèle hydrologique pluie-débit. Le modèle probabiliste de pluie est un simulateur d'averses. Ses paramètres sont les pluies moyennes, les pluies adjacentes (qui encadrent l'épisode pluvieux) et les pluies antérieures, pour un type de temps donné. La méthode SCHADEX repose en réalité sur deux modèles pluie-débit. Il s'agit des modèles MORDOR et GR5X. Ce sont des modèles hydrologiques globaux et conceptuels.

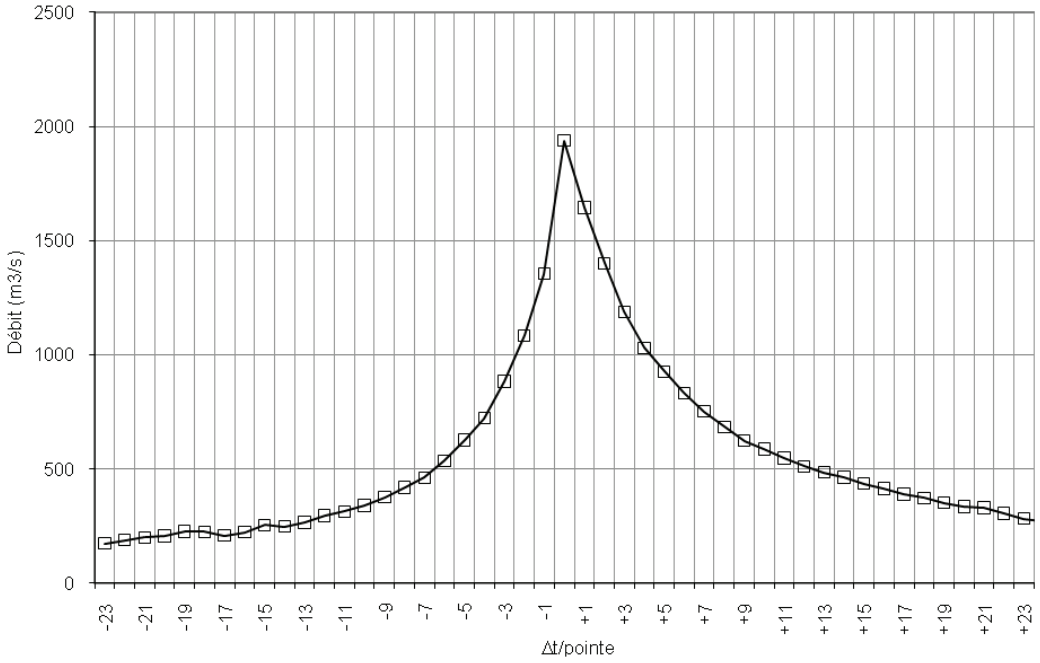

FIGURE 1.3 – Exemple d'hydrogramme de crue

1.1.5.2 Cotes de la retenue d'un barrage

Lors de la modélisation, le niveau d'eau dans la retenue sera comparé à deux niveaux de référence [Peyras *et al.*, 2006] : le niveau normal d'exploitation et le niveau des plus hautes eaux.

Le **niveau normal d'exploitation** (noté RN : Retenue Normale) correspond au niveau maximum du plan d'eau en exploitation normale, en dehors des épisodes de crue. Il s'agira de notre niveau initial.

Les **Plus Hautes Eaux** (notées PHE) correspondent au niveau de la retenue obtenu pour la crue de projet. Cette crue arrive sur un barrage dont la retenue est à la cote RN, sans dysfonctionnement de l'évacuateur de crues. La PHE est une donnée du dimensionnement du barrage. Le barrage doit fonctionner normalement en cas d'atteinte de la PHE. L'atteinte de la PHE est l'un de nos événements indésirables en cas de crue.

1.2 Enjeux industriels

Les acteurs de la sûreté hydraulique ont mis en place un vaste programme de rénovation et de modernisation du parc hydraulique. La vocation de ce programme est l'amélioration de leur performance et de leur disponibilité, et la sûreté à long terme

des aménagements hydrauliques [EDF, 2011]. En particulier, la nouvelle réglementation rend obligatoire la réalisation d'études de danger des ouvrages hydrauliques, en confrontant leur dimensionnement à des situations extrêmes telles que des crues exceptionnelles ou le dysfonctionnement des EdC vannés. L'objectif est la caractérisation des risques et l'identification des parades et des moyens de prévention et de protection permettant de les maîtriser. Ces risques peuvent être aussi bien intrinsèques à l'ouvrage hydraulique que susceptibles de se manifester à l'occasion de phénomènes exceptionnels tels que des crues ou des séismes.

Au sein du département MRI (Management des Risques Industriels) d'EDF R&D, le projet ARSHY (Analyse des Risques des Systèmes HYdrauliques), piloté par Hassane Chraibi, vise à développer des méthodologies d'analyse de risques systèmes pour le parc hydraulique d'EDF. En particulier, ce projet a pour objectif de consolider la méthode d'évaluation de la fiabilité des évacuateurs de crue vannés en développant l'outil GASPART (de l'anglais *Gated Spillway System - Probabilistic Assessment of Reliability Tool*) associé à cette méthode d'analyse de risques.

1.2.1 La méthode GASPART et l'outil associé

L'outil GASPART [Chraibi, 2013b] est une base de connaissances dédiée à la modélisation des évacuateurs de crues. Cette base de connaissances est développée sur la plate-forme logicielle KB3 reposant sur le langage de modélisation Figaro. La plate-forme outils KB3 [Flori et Donat, 2011] fournit une Interface Homme-Machine (IHM) de modélisation graphique permettant, à partir d'une base de connaissances Figaro adaptée, de saisir graphiquement la représentation des systèmes étudiés. Enfin, des outils de traitement sont également disponibles pour construire automatiquement des arbres de défaillances si le modèle est statique ou calculer divers indicateurs avec les outils YAMS (simulation de Monte-Carlo) ou Figseq (Exploration de séquences) lorsque le modèle considéré est dynamique.

Le langage de modélisation Figaro est un langage déclaratif conçu et développé par le département MRI pour l'évaluation de la fiabilité des systèmes à états discrets.

1.2.2 Réalisations et limites de l'outil GASPART

Dans le cadre de l'évaluation de la fiabilité des EdC vannés, la méthode GASPART se distingue des approches classiques par la prise en compte de deux facteurs déterminants pour la réussite du processus d'évacuation d'une crue, à savoir le facteur humain et le facteur temps.

Considérer le facteur humain revient à quantifier les actions dont l'opérateur est responsable par des probabilités d'échec. L'élaboration des données de fiabilité humaine, tout comme la définition du taux de défaillance des composants, est une part importante

du projet ARSHY. Cependant ce n'est pas l'objet de cette thèse et ces données de fiabilité seront simplement utilisées comme données d'entrée du modèle proposé.

L'outil GASPART évalue trois types de grandeurs :

1. la fréquence d'occurrence de l'événement redouté (ER) caractérisé par le dépassement d'un seuil critique par le niveau d'eau dans la retenue. Cette fréquence est donc une fréquence globale, toutes crues confondues. Cet indicateur permet de hiérarchiser les EdC vis-à-vis du risque lié à l'ER ;
2. la fiabilité de l'évacuateur vis-à-vis de chaque type de crue, afin de hiérarchiser les risques associés aux différentes crues. Par exemple, le niveau de risque le plus important peut provenir de la crue millénale ou de crues moins intenses mais plus fréquentes.
3. les séquences prépondérantes qui mènent à l'événement redouté, afin d'identifier les événements (défaillances) les plus contributeurs à la probabilité conditionnelle de l'ER. Connaître ces scénarios de défaillance associés à leur probabilité d'occurrence permet de les hiérarchiser par ordre d'importance de cette contribution probabiliste. Ces informations permettent de situer les leviers d'amélioration de la fiabilité de l'évacuateur. Ces améliorations peuvent porter sur des modifications d'architecture, des dimensionnements de composants, des actions qui augmentent la fiabilité matérielle ou humaine, ou des modifications des conditions d'exploitation.

Considérer soigneusement le facteur temps est indispensable dans ce type d'analyse. La dynamique du processus de crue et de son évacuation est telle que l'évolution physique et déterministe du niveau d'eau dans la retenue est intimement liée aux événements discrets aléatoires qui vont affecter l'ouverture des vannes. En ce sens, les EdC sont des systèmes dynamiques hybrides et rentrent dans le cadre de la fiabilité dynamique. Ces notions seront définies plus précisément dans la section 2.1.2.1.

Initialement conçu pour traiter les systèmes discrets, le langage Figaro ne permet la modélisation des systèmes dynamiques hybrides qu'au prix d'hypothèses de modélisation conservatives, évoquées dans le chapitre 4.1.6. Ces hypothèses de modélisation permettent de prendre en compte les phénomènes continus et transitoires, en adoptant des méthodes simplifiées de résolution des équations différentielles. Ces travaux de thèse d'une part et la conception d'un nouvel outil PyCATSHOO d'autre part ont pour but de lever ces hypothèses conservatives pour modéliser la dynamique des évacuateurs de crues de façon plus réaliste.

1.2.3 Objectifs de ces travaux

L'objectif de nos travaux de recherche est double :

1. proposer une démarche de modélisation pour les systèmes de type « évacuateur de crues ». Les modèles élaborés dans ce cadre se veulent les plus représentatifs

possible du fonctionnement de deux évacuateurs différents et réels, d'une part pour représenter leur caractère dynamique hybride, d'autre part pour respecter les contraintes introduites dans la section 1.1.3 ;

2. identifier et quantifier l'intérêt de prendre en compte de nouvelles informations temporelles dans l'évaluation des performances fiabilistes.

Il y a une différence significative entre une vanne qui refuse de s'ouvrir à la sollicitation, et une vanne qui ne connaît de défaillance qu'à la fin du processus de son ouverture : la progression de son processus d'ouverture est stoppée alors que la vanne est presque totalement ouverte. Dans le premier cas, la vanne ne participera pas du tout à l'évacuation de la crue, alors que dans le deuxième cas, le débit déversé par la vanne, ajouté à celui des autres vannes, pourra être suffisant pour absorber la crue. Il existe dont autant de trajectoires de l'évolution du niveau dans la retenue que de dates de défaillance possibles au cours de l'ouverture d'une vanne. Notre principal enjeu est donc d'affiner la modélisation en considérant les défaillances en fonctionnement de chacun des composants, et en exploitant ces nouvelles données que sont les instants de défaillance en fonctionnement.

Du point de vue de la modélisation, nous recherchons donc une méthode
- relevant de la fiabilité dynamique,
- capable de décrire un système de taille industrielle dans toute sa complexité,
- qui stocke et retourne des résultats sans perte d'information temporelle.

Du point de vue de l'exploitation des résultats et de cette information temporelle, nous recherchons
- un estimateur de la probabilité d'occurrence de l'événement redouté
- des indicateurs capables d'identifier les composants qui contribuent le plus à l'ER et à quel moment,
- un modèle permettant de classer les instants de défaillance des composants en fonction de l'issue de la crue.

Si le domaine de l'énergie hydraulique sert de cadre à nos recherches, les développements proposés ne sont pas spécifiques aux évacuateurs de crues et sont applicables à tout système dynamique hybride, quel que soit son domaine industriel, conditionnellement à l'adaptation du modèle à la nouvelle catégorie de système étudié.

1.3 Conclusion

Contrairement à un problème de fiabilité classique, l'état du système n'est pas une fonction de l'état de ses n composants. Ce sont seulement les dates de défaillance en fonctionnement $(T_1, ..., T_n)$ qui définissent l'occurrence ou non de l'ER pour le système. Cette dépendance est effective par le biais d'une fonction $f(T_1, ..., T_n) = s$ où $s \in \{ER, \overline{ER}\}$ désigne l'état du système (occurrence ou non de l'événement redouté

ER). Ainsi, pour un même ensemble de composants en panne, le système peut être en panne ou en marche, selon les dates de panne des composants.

Pour prendre en compte cette information temporelle jusqu'au bout du processus d'évaluation, certaines notions de sûreté comme celles de coupes ou de mesures d'importance sont à redéfinir. C'est l'objet de cette thèse. Les notions de fiabilité classique sont introduites dans le chapitre 2 avant d'être adaptées aux systèmes dynamiques hybrides dans les chapitres 5 et 6, dans l'objectif d'identifier la fonction f et de caractériser les histoires de défaillance.

Ce problème est commun aux systèmes dynamiques hybrides. La prise en compte de l'information temporelle est corrélée à l'introduction d'une variable déterministe continue dans le processus stochastique. Si cette thèse est motivée par l'évaluation de la sûreté des EdC, c'est au cadre général de la fiabilité dynamique que la méthodologie proposée cherche à s'appliquer.

Des applications industrielles mettant en jeu des systèmes dynamiques hybrides ont en effet déjà été identifiées. Les modèles proposés considèrent l'évolution d'une variable physique dans le temps pour décrire le comportement du système étudié, mais les résultats ne font pas ressortir la composante temporelle.

- Dans le domaine aéronautique, l'épaisseur de la structure en aluminium d'un missile dépend des temps de séjour de ce missile dans des environnements corrosifs [Brandejsky, 2012]. L'application de notre méthodologie permettrait de pronostiquer la rupture de cette structure directement en fonction de ces temps de séjours.
- Dans le contexte nucléaire, les EPS (Études Probabilistes de Sûreté) « dynamiques » ou « de niveau 2 » insèrent dans leurs modèles des variables déterministes continues. Ces variables, à l'image d'une pression ou d'une température, dépendent d'événements aléatoires survenus lors du processus de conduite de l'installation. Les EPS sont des méthodes d'évaluation des risques fondées sur une investigation systématique des scénarios accidentels. Ces risques sont quantifiés en termes de fréquence des événements redoutés et de leurs conséquences. Les EPS de niveau 2 ont également pour objectif d'évaluer la nature, l'importance et les fréquences des rejets hors de l'enceinte de confinement, suite à un scénario accidentel de type « fusion du cœur ».

Les EdC constituent donc un support et une illustration pour ces travaux mais ce ne sont pas les seuls systèmes susceptibles d'être concernés par la prise en compte de l'information temporelle en fiabilité dynamique.

Chapitre 2

Enjeux méthodologiques

Ce chapitre énonce les enjeux méthodologiques impliqués par le problème industriel. Si la section 2.1 présente les principales notions de sûreté de fonctionnement et introduit celles de fiabilité dynamique, la section 2.2 répertorie les différentes méthodes utilisées en fiabilité dynamique dans un état de l'art. Force est de constater qu'aucune méthode existante ne propose de levier efficace face aux verrous théoriques évoqués dans la section 1.2. Aussi, la section 2.3 annonce et justifie la méthodologie choisie et résume les contributions de ces travaux de thèse.

2.1 Principales notions de sûreté de fonctionnement et introduction à la fiabilité dynamique

Le premier but de cette section est de rappeler les principales notions de sûreté de fonctionnement et les indicateurs qui la caractérisent. Ces notions générales laissent ensuite place à une définition de la fiabilité dynamique et aux systèmes dont celle-ci étudie les performances fiabilistes : les systèmes dynamiques hybrides.

2.1.1 Principales notions de sûreté de fonctionnement

2.1.1.1 Grandeurs caractéristiques de la sûreté de fonctionnement

La sûreté de fonctionnement est caractérisée par quatre concepts : la fiabilité, la disponibilité, la maintenabilité et la sécurité. Ces quatre grandeurs sont notamment définies dans [Villemeur *et al.*, 1988] et [Rausand et Høyland, 2004]. Nous présenterons aussi brièvement les méthodes les plus fréquemment utilisées, qui permettent de calculer la fiabilité de systèmes statiques.

2.1.1.1.1 Fiabilité

Définition 1. La **fiabilité** est l'aptitude d'un système S à accomplir une mission requise, dans des conditions données, pendant l'intervalle de temps $[0, t]$. Cette aptitude est mesurée par la probabilité

$$R(t) = \mathbb{P}\left(S \text{ assure sa mission à } t',\ \forall t' \in [0, t]\right). \tag{2.1}$$

2.1.1.1.2 Disponibilité

Définition 2. La **disponibilité** est l'aptitude d'un système S à être en état d'accomplir une mission requise, dans des conditions données, à un instant t donné. Cette aptitude est mesurée par la probabilité

$$A(t) = \mathbb{P}\left(S \text{ assure sa mission à } t\right). \tag{2.2}$$

Contrairement au calcul de la fiabilité, celui de la disponibilité n'est pas affecté par l'historique des réparations éventuelles avant l'instant t. En effet, le fonctionnement à l'instant t ne nécessite pas forcément le fonctionnement sur $[0, t]$ pour un système réparable. Ainsi, pour tout système S, à tout instant t, la relation $R(t) \leq A(t)$ est vérifiée. Si le système S n'est pas réparable alors $R(t) = A(t)$.

2.1.1.1.3 Maintenabilité

Définition 3. La **maintenabilité** est l'aptitude d'un système S à être maintenu ou rétabli dans un état dans lequel il peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et des moyens prescrits. Cette aptitude est mesurée par la probabilité

$$M(t) = \mathbb{P}\left(S \text{ est réparé sur } [t_0,\ t_0 + t]\right) \tag{2.3}$$

où t_0 est l'instant de défaillance.

Les concepts de la fiabilité, de la disponibilité et de la maintenabilité sont souvent associés à celui de la sécurité. Pourtant, la sécurité n'est pas une grandeur toujours quantifiable.

Définition 4. La **sécurité** est l'aptitude d'un système à éviter l'apparition, dans des conditions données, d'événements critiques ou catastrophiques.

Les acronymes FMDS (Fiabilité / Maintenabilité / Disponibilité / Sécurité) ou RAMS (*Reliability / Availability / Maintenability / Safety*) sont souvent utilisés pour désigner les activités liées à la sûreté de fonctionnement.

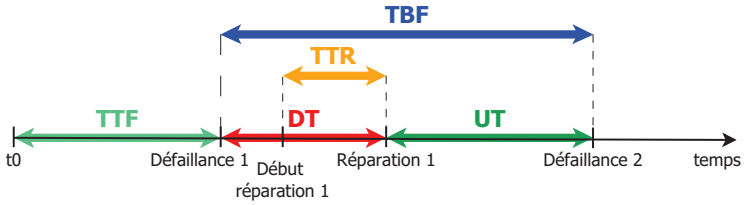

FIGURE 2.1 – Durées fondamentales en sûreté de fonctionnement

2.1.1.2 Durées fondamentales en sûreté de fonctionnement

La figure 2.1 illustre ces définitions de durées fondamentales.

Définition 5. Le **MTTF (*Mean Time To Failure*)** est la durée moyenne de fonctionnement d'une entité avant la première défaillance :

$$MTTF = \int_0^\infty R(t)dt. \tag{2.4}$$

Définition 6. Le **MTTR (*Mean Time To Repair*)** est la durée moyenne de réparation :

$$MTTR = \int_0^\infty [1 - M(t)]\, dt. \tag{2.5}$$

Définition 7. Le **MUT (*Mean Up Time*)** est la durée moyenne de fonctionnement après réparation. Si le système possède un seul bon état de fonctionnement alors $MUT = MTTF$.

Définition 8. Le **MDT (*Mean Down Time*)** est la durée moyenne d'indisponibilité. Si le système possède un seul état de panne alors $MDT = MTTR$.

Définition 9. Le **MTBF (*Mean Time Between Failure*)** est la durée moyenne entre deux défaillances consécutives d'une entité réparée :

$$MTBF = MUT + MDT. \tag{2.6}$$

2.1.1.3 Taux de défaillance et de réparation

Définition 10. Le **taux de défaillance** (instantané) est la limite, lorsque Δt tend vers 0, du quotient de la probabilité conditionnelle pour que l'instant T d'une défaillance soit compris dans l'intervalle de temps $[t,\ \Delta t]$, sachant que la défaillance n'avait pas encore eu lieu à l'instant t, et de la durée Δt :

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}\left(t < T \leq t + \Delta t / T > t\right)}{\Delta t}. \tag{2.7}$$

Définition 11. Le **taux de réparation** (instantané) est la limite, lorsque Δt tend vers 0, du quotient de la probabilité conditionnelle pour que l'instant T_R de l'achèvement d'une réparation soit compris dans l'intervalle de temps $[t,\ \Delta t]$, sachant que la panne a eu lieu pendant l'intervalle $[0, t]$, et de la durée Δt :

$$\mu(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}\left(t < T_R \leq t + \Delta t / T_R > t\right)}{\Delta t}. \tag{2.8}$$

2.1.1.4 Relations fondamentales

Des relations fondamentales permettent d'obtenir l'expression de la fiabilité à partir du taux de défaillance ou celle de la maintenabilité à partir du taux de réparation.

$$R(t) = \exp\left(-\int_0^t \lambda(u)du\right) \text{ en posant } R(0) = 1 \tag{2.9}$$

$$M(t) = 1 - \exp\left(-\int_0^t \mu(u)du\right) \text{ en posant } M(0) = 0 \tag{2.10}$$

2.1.1.5 Méthodes classiques utilisées en sûreté de fonctionnement

Les méthodes telles que les diagrammes de fiabilité, les arbres de défaillances ou les arbres d'événements sont des méthodes pour évaluer les grandeurs caractéristiques de la sûreté de fonctionnement d'un système. La fiabilité (respectivement disponibilité) d'un système est ainsi calculée à partir de la connaissance de la fiabilité (respectivement disponibilité) des composants de ce système. Ces méthodes s'appliquent dans le cadre de systèmes statiques, pour lesquels la prise en compte de l'information temporelle n'est pas nécessaire.

La méthode du **diagramme de fiabilité** est une représentation logique du système (figure 2.2). Le diagramme construit est composé de blocs (chaque bloc représente une

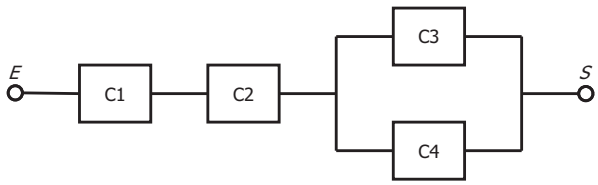

FIGURE 2.2 – Diagramme de fiabilité d'un système constitué de quatre composants. Le système accomplit sa mission si et seulement si il existe un chemin continu entre l'entrée E et la sortie S.

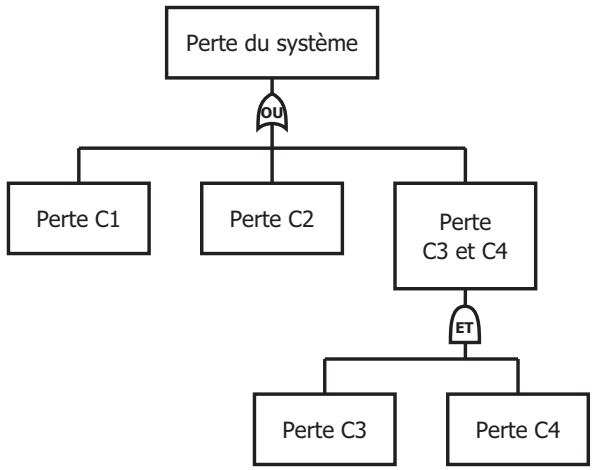

FIGURE 2.3 – Arbre de défaillances du système décrit par le diagramme de fiabilité de la figure 2.2

entité) reliés par des arcs orientés indiquant les dépendances des entités entre elles. Le comportement des entités est binaire (fonctionnement/défaillance). Cette méthode permet de représenter la structure de systèmes composés de sous-systèmes dont la structure est série ou parallèle, et d'en calculer la disponibilité. Ce calcul est réalisé sous de fortes hypothèses d'indépendance des entités et dans le cas où le système est statique.

Un **arbre de défaillances** (figure 2.3), aussi appelé arbre des causes ou arbre de fautes, de l'anglais *fault tree,* est une technique qui représente les causes d'un événement indésirable donné de manière arborescente : l'identification des événements intermédiaires puis des événements de base a lieu étape par étape, grâce à des portes élémentaires (ET / OU) ou des portes plus élaborées (PAND, XOR, etc.).

Un **arbre d'événements** (figure 2.4), de l'anglais *event tree*, aussi appelé arbre de conséquences, est une technique reposant sur le principe inverse. A partir d'un événement initiateur, l'analyste imagine les différentes parades destinées à éviter l'événement

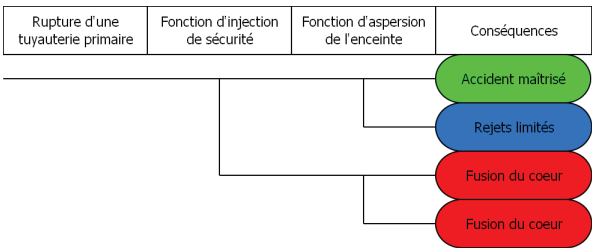

FIGURE 2.4 – Exemple d'arbre d'événements, inspiré du domaine nucléaire

indésirable et élabore des scénarios de défaillances en fonction de la réussite ou non de ces parades. La réussite de toutes les parades constitue la branche mère de l'arbre, à laquelle se greffent les autres branches représentant les scénarios alternatifs.

Dans les arbres de défaillances comme dans les arbres d'événements, une coupe est une combinaison d'événements entraînant l'événement redouté.

2.1.1.6 Mesures d'importance

En sûreté de fonctionnement, l'analyste est amené à cibler quel composant réparer en priorité lorsque le système est en panne, ou quel composant améliorer en priorité pour augmenter la fiabilité du système. C'est pour répondre à ce type de questions que les facteurs d'importance ont été définis [Lambert, 1975], [Villemeur *et al.*, 1988], [Duflot, 2007], [Rausand et Høyland, 2004].

Parmi les principales mesures d'importance, sept facteurs d'importance classiques mesurent les variations de risque liées à la réalisation ou non d'un événement de base. Notons ER la réalisation de l'événement redouté et eb_i la réalisation d'un événement de base, liée à la défaillance du composant i. $\bar{\omega}$ désigne le complémentaire de l'événement ω.

1. L'**indicateur de Birnbaum** (IB) [Birnbaum, 1968] s'exprime comme la différence entre le risque sachant que eb_i s'est produit et le risque sachant que cet événement ne peut pas se produire :

$$I_B(eb_i) = P(ER/eb_i) - P(ER/\overline{eb_i}). \tag{2.11}$$

2. Le **Facteur d'Accroissement de Risques** (FAR) est l'augmentation relative du risque provoquée par la certitude de eb_i :

$$FAR(eb_i) = \frac{P(ER/eb_i) - P(ER)}{P(ER)}. \tag{2.12}$$

3. Le **Facteur de Diminution de Risques** (FDR) est la diminution relative du risque provoquée par l'impossibilité de eb_i :

$$FDR(eb_i) = \frac{P(ER) - P(ER/\overline{eb_i})}{P(ER)}. \tag{2.13}$$

4. Le ***Risk Achievment Worth*** (RAW) est le coefficient multiplicateur du risque provoqué par la certitude de eb_i :

$$RAW(eb_i) = \frac{P(ER/eb_i)}{P(ER)}. \tag{2.14}$$

5. Le ***Risk Reduction Worth*** (RRW) est le coefficient diviseur du risque provoqué par l'impossibilité de eb_i :

$$RRW(eb_i) = \frac{P(ER)}{P(ER/\overline{eb_i})}. \tag{2.15}$$

6. Le **Facteur de Sensibilité** (FS) est l'augmentation relative du risque provoquée par une variation δp_i de la probabilité d'occurrence p_i de eb_i :

$$FS(eb_i,\, \delta p_i) = \frac{P(ER/p_i' = p_i + \delta p_i) - P(ER/p_i)}{P(ER/p_i)}. \tag{2.16}$$

7. Le **Facteur de Vesely-Fussel** (VF) est la probabilité qu'une coupe contenant eb_i soit réalisée sachant que l'événement ER s'est produit :

$$VF(eb_i) = P\left(\left(\cup_{eb_i \in C_j} C_j\right)/ER\right) \tag{2.17}$$

où C_j désigne une coupe et $\left(\cup_{eb_i \in C_j} C_j\right)$ l'ensemble des coupes contenant eb_i.

2.1.2 Introduction à la fiabilité dynamique

La dynamique d'un système est parfois telle que l'évolution physique et déterministe d'une variable environnementale est intimement liée aux événements discrets aléatoires qui vont affecter le fonctionnement des composants du système. Ce type de système fait partie de la classe des systèmes dynamiques hybrides (SDH) et rentre dans le cadre de la fiabilité dynamique. La prise en compte de l'information temporelle est corrélée à l'introduction d'une variable déterministe continue dans le processus stochastique. Les méthodes classiques d'évaluation de la sûreté de fonctionnement décrites ci-dessus présentent des limites dans la prise en compte du temps.

2.1.2.1 Définition d'un système dynamique hybride

Définition 12. Un **système dynamique hybride** (SDH) est caractérisé par le couplage d'événements aléatoires discrets d'une part, et par des phénomènes déterministes continus et transitoires d'autre part.

Terminologie liée aux variables aléatoires discrètes
Les variables aléatoires (v.a.) discrètes caractérisent la configuration du système. Chaque composant est décrit par une ou plusieurs v.a., qui correspondent par exemple aux transitions entre différents modes de défaillances, différentes options d'alimentation ou différents états de marche. S'il s'agit d'un système simple, une variable aléatoire discrète sera nommée « état du système » ou encore « mode ». Dans le cadre de systèmes complexes, constitué de plusieurs composants, la configuration du système sera une combinaison des états de chacun de ses composants.

Terminologie liée aux variables déterministes continues
Une ou plusieurs variables déterministes continues, appelées également variables environnementales ou variables physiques, peuvent caractériser un phénomène physique affectant le système, comme la pression, la température, l'âge du système ou un niveau d'eau.

De nombreux systèmes sont désignés dans la littérature sous le nom de « systèmes dynamiques hybrides », à l'image des travaux de [Zaytoon, 2001] dans le domaine de l'automatique. Cependant, le terme « hybride » désigne souvent un système alliant des composantes discrètes et continues, ou un système alliant des composantes déterministes et stochastiques. Notre définition d'un système dynamique hybride est plus large puisqu'elle inclut des variables discrètes et stochastiques et des variables continues et déterministes. Un système dynamique hybride sera donc caractérisé dans la suite de ce mémoire par le couplage entre des variables aléatoires discrètes et des variables déterministes continues et transitoires.

2.1.2.2 Définition de la fiabilité dynamique

Définition 13. La **fiabilité dynamique** est l'étude de l'évolution des systèmes dynamiques hybrides du point de vue de la fiabilité.

Un modèle de fiabilité dynamique permet de représenter les interactions entre des événements stochastiques discrets et des variables déterministes continues. Par exemple, l'évolution du niveau d'eau dans un réservoir dépend de la configuration des vannes qui la vidangent et donc de leurs défaillances éventuelles. Réciproquement, les ordres d'ouverture de ces vannes dépendent de la position du niveau d'eau par rapport à certains seuils. Cette double interaction est complexe à décrire et à quantifier en fiabilité dynamique.

Le *heated tank*, cas-test de la fiabilité dynamique
Un cas-test, appelé *heated tank* ou simplement « réservoir » est fréquemment utilisé afin de comparer les différentes méthodes en fiabilité dynamique [Broy *et al.*, 2011a], [Lair, 2011], [Perez Castaneda, 2009], [Marseguerra et Zio, 1996]. Il s'agit d'un réservoir

alimenté par deux pompes et vidangé par une vanne. Le liquide contenu dans le réservoir est chauffé. L'objectif est de contrôler ces trois composants pour maintenir le liquide dans un intervalle de niveau donné, et sous un seuil de température. La modélisation de ce *benchmark* fait apparaître une interaction supplémentaire entre variables physiques déterministes et événements discrets aléatoires : le taux de défaillance est une fonction des variables physiques. [Chraibi, 2013a] fait également apparaître cette particularité dans une variante du *benchmark* plus proche de l'objet de la thèse : les vannes y ont un fonctionnement intermédiaire entre les positions « ouverte » et « fermée ».

2.1.2.2.1 Modélisation par les Processus Markoviens Déterministes par Morceaux (PDMP)

Une classe de processus permet généralement de modéliser les systèmes dynamiques hybrides. Il s'agit des processus markoviens déterministes par morceaux ou PDMP, de l'anglais *Piecewise Deterministic Markov Process*. Les PDMP ont été introduits par [Davis, 1984] et leur construction est décrite dans la section 3.1.

Un PDMP est la représentation mathématique d'un système dynamique hybride par un couple où X_t représente un vecteur de variables déterministes continues et I_t la configuration du système à l'instant t.

Une fois le cadre théorique des PDMP choisi, il est nécessaire dans un deuxième temps de décrire et d'implémenter ce processus pour un système réel donné. Pour des systèmes industriels complexes, cette description nécessite souvent l'utilisation d'un formalisme de modélisation (graphique ou non) permettant de spécifier la dynamique d'évolution du système considéré. Les méthodes de description sont donc autant d'approches de codage et de simulation des PDMP. Le PDMP sous-jacent est modélisé au prix d'hypothèses plus ou moins simplificatrices, selon la méthode de description utilisée. Enfin, des analyses quantitatives sont réalisées. La méthode de quantification permettra entre autres d'obtenir des indicateurs sur la fiabilité du système et sur ses défaillances les plus critiques.

2.2 État de l'art en fiabilité dynamique

Cette section passe en revue les différentes méthodes utilisées en fiabilité dynamique. L'objectif est d'identifier une méthode capable d'une part de décrire des systèmes dynamiques hybrides (SDH) de taille industrielle, d'autre part de tirer profit de la variable temporelle en termes d'indicateurs et de métriques. Une telle méthode répondrait aux enjeux exposés dans la section 1.2

Depuis le début des années 2000, la fiabilité dynamique est étudiée par diverses

équipes de recherche [1] dans le cadre de la modélisation et la quantification de SDH. Cet état de l'art n'est pas le premier de la littérature. Beaucoup de travaux au sujet de la fiabilité commencent par se situer par rapport aux méthodes existantes, qu'il s'agisse de travaux de thèse [Lair, 2011], [Perez Castaneda, 2009], d'articles dédiés à la revue de ces méthodes [Siu, 1994], [Labeau *et al.*, 2000], [Aldemir, 2012] ou de projets réunissant plusieurs laboratoires pour comparer l'application de différentes méthodes sur un même cas-test [Kermisch et Labeau, 2000], [Raimond et Durin, 2007], [Aubry *et al.*, 2012], [Adolfsson *et al.*, 2012].

2.2.1 Méthodes de description

Dans cette section sont listées les principales méthodes de description des systèmes dynamiques hybrides. Toutes ces approches ont pour but de représenter au mieux l'évolution d'un système caractérisé à la fois par un phénomène transitoire continu et par des transitions entre les différents états du système ou de ses composants. Le formalisme utilisé doit donc être capable de gérer les équations différentielles spécifiques à la dynamique du système, les événements aléatoires et les distributions de probabilité associées, et les interactions entre ces deux composantes. Les systèmes d'application de chaque méthode doivent présenter explicitement le traitement d'une ou plusieurs variables déterministes continues. Il existe une dizaine de méthodes de description des SDH. Une attention toute particulière est portée à leur représentation graphique et à leur lisibilité, garantes d'une bonne compréhension entre les différents corps de métier de la sûreté de fonctionnement.

Il est possible d'élaborer un classement des méthodes de description en fonction de leur niveau d'abstraction. Une première catégorie réunit les méthodes analytiques et semi-analytiques. Ces méthodes, les plus abstraites, se basent sur la résolution des équations différentielles d'une part, des équations de Chapman-Kolmogorov d'autre part pour exprimer de manière analytique la loi jointe du processus. Une deuxième classe de méthodes repose sur les arbres d'événements dynamiques et leurs variantes. Des codes de calcul sont en charge de la résolution des équations de Chapman-Kolmogorov, pour un type de système donné. Le point commun du troisième groupe de méthodes est l'ajout d'un formalisme graphique à cette résolution. Ces dernières méthodes se veulent plus génériques, dans le but de représenter les systèmes dynamiques hybrides en général. En ce sens, ces méthodes facilitent l'étude de la sûreté d'un SDH dès lors qu'un SDH similaire a déjà été modélisé.

1. Parmi les équipes de recherche ayant étudié la fiabilité dynamique figurent notamment les équipes de Marne la Vallée [Cocozza-Thivent *et al.*, 2006b], de Bordeaux [Zhang *et al.*, 2013], de Nancy [Brinzei *et al.*, 2009], de Pau [Lair *et al.*, 2011] et de Troyes [Brissaud, 2010]. D'un point de vue international, il s'agit des équipes de Bruxelles [Labeau *et al.*, 2000], de Milan [Zio et Maio, 2009], de Munich [Kloos et Peschke, 2006] et de l'Ohio [Aldemir *et al.*, 2007].

2.2.1.1 Les méthodes analytiques et semi-analytiques

Dans le cadre de travaux menés par [Cocozza-Thivent *et al.*, 2006c], [Cocozza-Thivent *et al.*, 2004], [Cocozza-Thivent *et al.*, 2006b], [Cocozza-Thivent, 2012], des petits systèmes sont décrits grâce à une représentation des PDMP : pour chaque configuration du système, l'équation différentielle régissant la (les) variable(s) continue(s) est écrite et résolue analytiquement. Les équations différentielles concernées sont souvent très simples. Les méthodes analytiques se basent sur la résolution des équations de Chapman-Kolmogorov pour estimer la loi jointe du processus. L'**équation de Chapman-Kolmogorov** est une égalité qui met en relation les lois jointes de différents points de la trajectoire d'un processus stochastique. Les méthodes semi-analytiques ont également pour but de résoudre ces équations, mais elles utilisent pour cela des approches numériques. Par exemple, [Cocozza-Thivent *et al.*, 2006a] utilise une méthode de volumes finis, c'est-à-dire une discrétisation et une linéarisation du système d'équations. Ensuite, la loi stationnaire du processus est approchée en inversant numériquement la matrice génératrice du processus.

Ces méthodes sont très efficaces pour calculer les probabilités marginales de petits systèmes aux interactions complexes (dépendance fonctionnelle ou redondance passive entre composants, dépendance à une ou plusieurs variables déterministes continues ...), ainsi que d'autres grandeurs comme la disponibilité asymptotique. Des algorithmes issus des méthodes de volumes finis, comme exposés dans la section 2.2.2.1 sont utilisés pour la quantification du système.

Ces méthodes constituent donc des approches très flexibles pour des systèmes ayant un petit nombre de composants (au plus dix-sept dans la littérature [Lair, 2011], [Lair *et al.*, 2011]). Elles ont l'avantage de modéliser tout type d'interactions et de lois. Malgré cela, ces méthodes ne possèdent pas d'outil d'aide à la modélisation (graphique ou non) et restent très spécifiques aux systèmes étudiés. Les calculs deviendraient vite fastidieux si le nombre de composants augmentait ou si les interactions entre variables étaient modifiées. De plus, le choix de la variable physique est relativement restreint : généralement, la variable continue choisie est le temps, ce qui permet à sa dérivée d'être égale à 1 ou 0. Cela simplifie considérablement les calculs analytiques. L'introduction d'une autre variable comme une température ou un niveau d'eau est possible [Lair, 2011], mais le nombre de configurations possibles du système doit demeurer limité.

2.2.1.2 Les méthodes reposant sur les arbres d'événements dynamiques

Les arbres d'événements sont une méthode reconnue et fréquemment utilisée en fiabilité « classique ». Ils sont brièvement décrits dans la section 2.1.1.5. Dès les débuts de la fiabilité dynamique, les auteurs se sont donc naturellement tournés vers ces méthodes pour les adapter aux systèmes dynamiques hybrides.

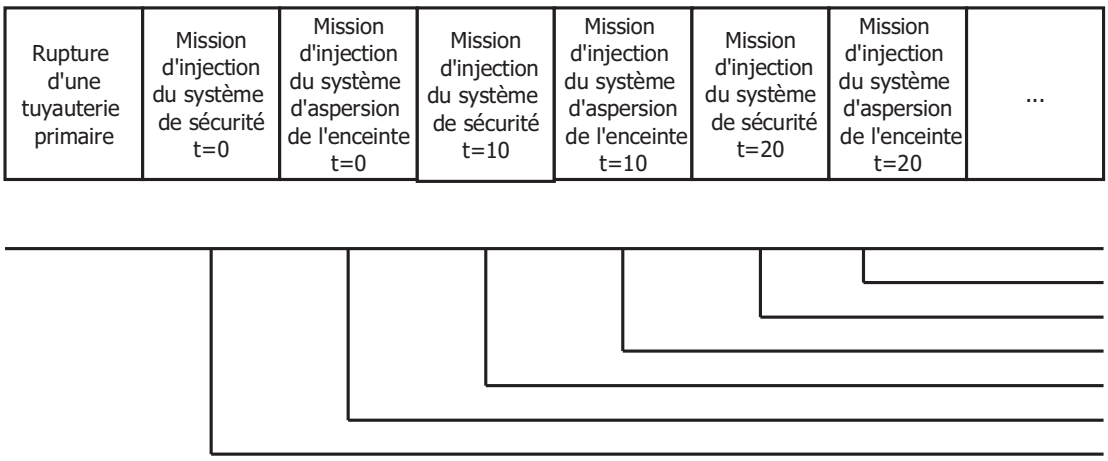

FIGURE 2.5 – Exemple de DDET. Le nombre de branches explose avec le nombre de parades et la discrétisation du temps. Les DDET sont donc peu représentés graphiquement dans la littérature.

1. Les **arbres d'événements continus** (CET, de l'anglais *Continuous Event Tree*) décrivent à la fois les variables aléatoires discrètes et les variables déterministes continues, ainsi que les interactions qui les relient [Smidts, 1994], [Labeau *et al.*, 2000], [Aldemir, 2012]. Cette méthode est proche des méthodes analytiques dans le sens où l'évolution de la densité de probabilité de la configuration du système est estimée à partir des équations de Chapman-Kolmogorov. Les CET sont associés aux méthodes de quantification telle que la simulation de Monte Carlo ou la *Continous Cell-to-Cell Mapping Technique*. A notre connaissance, les CET ne proposent pas de formalisme graphique dédié à la description du système ou du processus le caractérisant. Depuis le début des années 2000, leur version discrète est privilégiée pour répondre aux enjeux industriels.

2. Les **arbres d'événements dynamiques discrets** (DDET, de l'anglais *Discrete Dynamic Event Tree*) sont une variante des CET, où le temps est discrétisé. La branche mère représente le scénario idéal où aucune défaillance ne succède à l'événement initiateur. De nouvelles branches sont reliées à la branche mère à chaque pas de temps, comme l'illustre la figure 2.5. Chacun de ces nouveaux scénarios est déclenché par la défaillance d'un composant élémentaire [Smidts, 1994]. La probabilité de chaque branche est calculée ainsi que l'évolution de la variable déterministe correspondante. Le nombre de scénarios augmente avec la finesse du pas de temps choisi. D'autres branches peuvent être ajoutées, représentant le passage d'un seuil pour la variable physique, et ses répercussions sur le système. Enfin, le nombre de scénarios peut devenir très important si on autorise plusieurs composants élémentaires à être défaillants au cours de la même histoire : le nombre de branches en est multiplié de manière exponentielle.

Des techniques ont été développées afin de mieux gérer les multiples scénarios géné-

rés, et se différencient par leur mise en œuvre, les techniques de branchements, la mémorisation des arbres et la modélisation des interactions homme-machine [Labeau *et al.*, 2000], [Perez Castaneda, 2009]. Le DYLAM (*Dynamical Logical Methodology*) est historiquement le premier DDET. Il se distingue des arbres d'événements classiques par le fait qu'il n'est pas binaire. Le DETAM (*Dynamic Event Tree Analysis Method*) prend mieux en compte les facteurs humains. Enfin, les ADS (*Analyzer of Dynamic Systems*) ont été développés pour étudier des systèmes de grande taille et sont plus flexibles, ce qui a permis leur application en dehors du domaine nucléaire.

3. [Izquierdo et Labeau, 2004] et [Jourdain et Labeau, 2011] proposent une démarche en deux temps nommée ***Stimulus - Driven Theory of Probabilistic Dynamics*** **(SDTPD)**. Une première étape consiste à construire un DDET relativement simple qui décrit l'évolution du système sans tenir compte des événements dont l'instant d'occurrence n'est pas fixe. Pour chaque branche, la dynamique des variables déterministes est évaluée. Les événements qui dépendent d'une loi de probabilité continue sont ensuite injectés sous la forme de stimulus. L'instant d'occurrence du stimulus, si sa loi de distribution est connue, aura été simulé au préalable, en tenant éventuellement compte des paramètres physiques. Cependant, la réponse du système au stimulus n'est pas forcément immédiate : la perturbation occasionnée peut être répercutée sur l'évolution du processus après un certain délai. L'introduction de ces délais qui évitent de confondre le stimulus et le point de branchement qu'il provoque est une des forces de la méthode. Chaque perturbation est associée à sa probabilité d'occurrence au moment de son branchement. La SDTPD est notamment utilisée pour l'étude des événements rares.

[Faghihi, 2012] privilégie également une démarche en deux temps pour modéliser les conséquences d'un *black-out* sur un réseau électrique d'une dizaine de composants et sur l'évolution de l'impédance de ce SDH.

4. [Kloos et Peschke, 2006] et [Kloos, 2011] procèdent également en deux étapes avec la méthode ***Monte Carlo Dynamic Event Tree*** **(MCDET)**. Ils identifient tout d'abord quelles variables aléatoires suivent une loi de distribution continue, ce qui correspond aux temps de défaillance en fonctionnement des composants élémentaires. A N fixé, N réalisations de ces variables sont générées aléatoirement. Dans cette démarche, l'évolution des variables déterministes n'a pas encore été estimée, aussi ces temps de défaillance ne peuvent-ils pas dépendre des paramètres physiques. N DDET sont ensuite construits ; chaque arbre correspond à un jeu de ces instants simulés. Chaque branche d'un arbre représente une modalité des variables aléatoires discrètes, pour la plupart binaires, dont l'instant d'occurrence est constant. Le nombre de séquences des DDET est donc dépendant d'un pas de temps donné et s'en trouve significativement réduit. Les séquences dont la probabilité est inférieure à un certain seuil sont ignorées. Un code de calcul déterministe est ensuite appliqué à chaque séquence pour évaluer l'évolution du processus déterministe. Les résultats montrent que l'évolution de ces variables physiques dépend clairement des temps simulés au préalable. Cette méthode permet de décrire le comportement d'un système d'une vingtaine de composants, soumis à l'évolution de

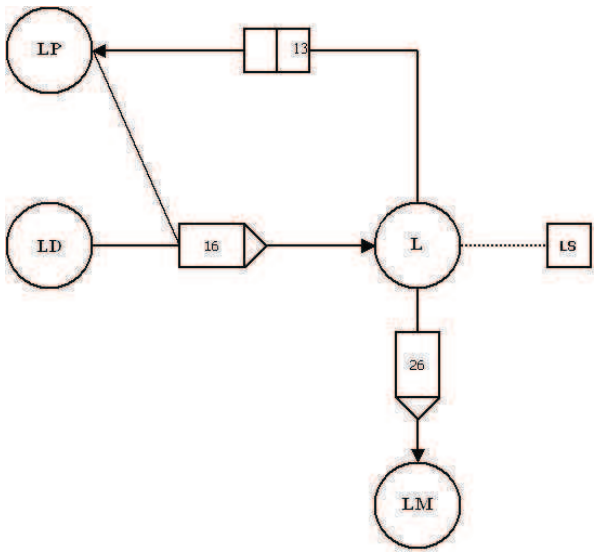

FIGURE 2.6 – Extrait du modèle DFM d'un *benchmark*.

quatre variables physiques, dans le domaine du nucléaire. Les recherches actuelles visent à affiner la description du comportement humain dans ce modèle.

5. La **méthode des graphes de flux dynamiques** (DFM, de l'anglais *Dynamic Flowgraph Methodology*) [Aldemir *et al.*, 2007], [Chaux et Deleuze, 2010] utilise des graphes orientés capables de modéliser un système complexe, de faire des analyses déductives (des effets vers les causes) et inductives (des causes vers les effets) pour quantifier des grandeurs caractéristiques d'événements redoutés. Chaque variable est représentée par un noeud, ce qui est illustré sur la figure 2.6. Les cercles représentent les variables continues (dont l'évolution est déterministe) discrétisées, et les carrés symbolisent l'état de défaillance des composants. Chaque variable est aussi décrite par une table, où chaque état défaillant est couplé à sa probabilité. Les variables sont reliées entre elles par des boîtes de transfert qui représentent les liens de causes à effets entre les variables, et par des boîtes de transition qui introduisent un délai entre l'instant où les variables d'entrée deviennent vraies et celui où les variables de sortie sont modifiées. Chacune de ces boîtes est associée à une table de décision. Le principe d'une table de décision est proche de celui des tables de vérité : à chaque combinaison des états des variables d'entrée est associé un état de la variable de sortie. Ces tables sont déterministes, elles ne contiennent pas de probabilités. Les tables de décision peuvent être construites à partir des avis d'experts, des équations physiques ou du code informatique en cas de système numérique. Les variables physiques continues qui interagissent avec le système sont discrétisées et traitées dans les tables de décision de la même manière que les variables aléatoires discrètes. La figure 2.6 ne représente qu'une partie d'un modèle DFM appliqué au *benchmark* d'un générateur de vapeur décrit dans le rapport [Aldemir *et al.*, 2007], mais on peut y voir l'essentiel des structures existantes.

La figure 2.6 peut s'interpréter ainsi :

- à partir du niveau de vapeur réel L, la boîte de transfert 26 détermine le niveau de vapeur connu LM, conditionnellement à l'état du capteur LS ;
- à partir du niveau de vapeur précédent LP et du changement du niveau de vapeur LD, la boîte de transfert 16 identifie le niveau de vapeur réel actuel L ;
- la boîte de transition indique qu'au bout de 13 unités de temps, le niveau de vapeur réel L est stocké dans le niveau de vapeur précédent LP.

L'analyse peut être inductive ou déductive. Une analyse inductive considère les effets et en recherche les causes, c'est-à-dire les séquences qui y mènent. L'analyse inductive détermine les effets à partir d'un état initial en listant les événements rendus possibles par cet événement initial.

Les DFM sont une approche markovienne, centrée sur les événements. Chaque événement élémentaire (par exemple la panne d'un composant, l'atteinte d'un seuil minimal) est associé à une probabilité. La probabilité de l'événement redouté est calculée à partir des probabilités des événements élémentaires, comme dans un arbre de défaillances. Les lois utilisées sont donc forcément des lois exponentielles. Malgré une représentation graphique très lisible qui permet de modéliser aisément de grands systèmes, le formalisme des DFM est très élémentaire du point de vue théorique pour le calcul des grandeurs de fiabilité. A notre connaissance, les algorithmes de quantification n'ont pas été publiés par les créateurs de la méthode. [Karanta, 2013] présente cependant l'implémentation des DFM sur un système dynamique hybride simple à l'aide d'une programmation logique. Les DFM ont été conçus pour s'adapter aux modèles existants dans le cadre des études probabilistes de sûreté américaines (PRA). Cette adaptation a été réalisée à l'aide de l'outil de programmation SAPHIRE et la méthode de quantification associée est l'injection de défauts. Toutefois, le rapport [Aldemir *et al.*, 2007] se restreint à la modélisation graphique du *benchmark* sans publier de précisions sur la mise en œuvre de la méthode. En 2010 [Chaux et Deleuze, 2010] utilise les DFM pour modéliser un petit système hybride et remarque que cette méthode est sensible à l'explosion combinatoire.

Conclusion sur les arbres d'événements dynamiques

Les méthodes dérivées des arbres d'événements dynamiques présentent plusieurs avantages : un formalisme graphique, l'association avec des codes de calcul déterministes reconnus, des outils de génération des DDET existants tels que l'outil ADAPT [Aldemir *et al.*, 2007], mais aussi un contexte mathématique adapté à la résolution de Chapman-Kolmogorov. Cela permet notamment de traiter les incertitudes liées au modèle. La majorité des études de fiabilité dynamique utilisent ces techniques, surtout dans le domaine de la sûreté nucléaire. Des travaux évoquent la possibilité de les inclure dans les EPS de niveau 2 [Raimond *et al.*, 2007], ce qui représente de potentielles applications sur des systèmes réels issus de l'industrie nucléaire.

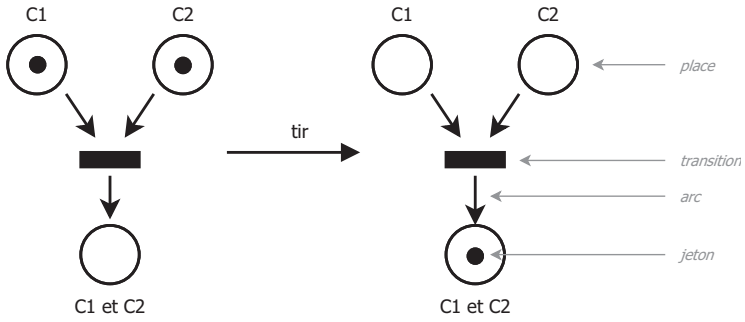

FIGURE 2.7 – Franchissement d'une transition dans un RdP : pour que le sous-système série « C1 et C2 » soit fonctionnel, il faut que les composants C1 et C2 soient tous les deux fonctionnels.

2.2.1.3 Les méthodes basées sur un formalisme graphique

Quelques méthodes se concentrent autour d'un formalisme graphique bien défini afin de décrire de manière intuitive l'évolution du processus. Dans la plupart des cas, le graphe associé illustre les transitions du système d'une configuration à une autre. Nous avons choisi de présenter ici les réseaux de Petri et les réseaux bayésiens dynamiques, car ces méthodes sont présentes de manière récurrente dans la littérature sur la fiabilité dynamique.

1. Les **réseaux de Petri (RdP)** [**Signoret *et al.*, 2013**], [**Dutuit *et al.*, 1997**], [**Medjoudj, 2006**], [**Medjoudj et Yim, 2007**] sont connus pour leur graphisme et leur flexibilité. Les RdP sont des graphes orientés composés de places représentant l'état du système modélisé, de transitions et d'arcs qui relient les places aux transitions et les transitions aux places. Comme l'illustre la figure 2.7, les RdP représentent l'état du système composant par composant. Un marquage du réseau est l'image d'un état du système : une évolution du système correspond à l'évolution du marquage, qui se traduit par le franchissement (tir) de transitions. Ces tirs se produisent lorsque toutes les places en amont de cette transition sont marquées. Ils correspondent, dans le RdP, à l'enlèvement d'une marque de chacune des places situées en amont de la transition et à l'ajout d'une marque dans chacune des places situées en aval de celle-ci. Le nouveau marquage du réseau décrit le nouvel état atteint par le système. Un tir est illustré sur la figure 2.7.

Les RdP peuvent être agrémentés de poids, d'arcs inhibiteurs et de messages afin d'enrichir le modèle. Afin de représenter également des systèmes dynamiques, des extensions des RdP ont été imaginées. Par exemple, les **réseaux de Petri temporisés** permettent de temporiser des places (lorsqu'une place est marquée, la marque est indisponible pendant un temps de séjour donné) ou des transitions (une transition n'est franchie qu'après un certain délai).

Les **réseaux de Petri stochastiques** permettent en outre de temporiser des tran-

sitions par un délai aléatoire. Les RdP stochastiques rendent possible la description d'un système relevant de la fiabilité dynamique et sont eux-même sujets de nombreux travaux qui proposent différents formalismes. Par exemple, les **RdP hybrides** combinent les composantes d'un RdP classique à des places continues dont le marquage est un nombre réel et à des transitions continues associées à la vitesse caractérisant la variable physique [Medjoudj, 2006], [Medjoudj et Yim, 2007]. [Chabot, 1998] et [Chabot *et al.*, 1998] introduisent une communication réciproque entre le RdP stochastique et le code de calcul déterministe. Cette communication, qui porte sur la mise à jour des paramètres, a permis de modéliser la progression d'un incendie.

Les **réseaux de Petri stochastiques Prédicats-Transitions Différentiels** (RdP PTDS) sont présentés par [Medjoudj, 2006], [Medjoudj et Yim, 2007], [Sadou, 2007] et [Sadou et Demmou, 2009]. Ils associent à chaque place un ensemble d'équations algébriques et différentielles : l'arrivée d'un jeton dans une place déclenche l'intégration des équations correspondantes. Ces RdP PTDS ont été utilisés pour la modélisation d'un train d'atterrissage (système dynamique hybride) dans le domaine aéronautique [Medjoudj, 2006].

Les **RdP stochastiques colorés** améliorent la lisibilité de cette méthode de description en associant des couleurs à différents types de composants. Plusieurs composants possédant les mêmes caractéristiques sont ainsi décrits par un seul RdP reconnaissable à sa couleur. [Brissaud, 2010], [Brissaud *et al.*, 2012] résoud ainsi un système de contrôle-commande de centrales nucléaires, composé de huit composants et de treize variables physiques, relevant ainsi de la fiabilité dynamique.

Les RdP stochastiques colorés ont une variante appelée **réseaux d'activités stochastiques** (SAN, de l'anglais *Stochastic Activity Network*) [Ghostine, 2008]. Dans un SAN, une transition est enrichie d'un ensemble d'activités possédant une porte d'entrée et une porte de sortie. Ces activités modélisent par exemple des tâches à planifier ou à exécuter dans le domaine de la robotique.

Les RdP permettent donc de représenter des systèmes aux propriétés très variées. En effet, toutes les interactions entre composants sont possibles ; d'autres lois que la loi exponentielle sont acceptées ; le nombre de composants n'est pas limité, notamment grâce à l'existence de sous-réseaux de Petri. Cependant, leur représentation graphique n'est en général pas facilement interprétable sans explication de la part de l'analyste qui a conçu le RdP. De plus, la complexité de cette représentation peut introduire des erreurs lors de la conception du RdP.

La fréquente utilisation des RdP a permis le développement de langages et de logiciels permettant de les construire tels que le langage Altarica et l'outil KB3. L'outil CPN Tools aide à la construction de RdP stochastiques colorés. Le module Petri du logiciel GRIF [GRIF, 2012] permet de modéliser le comportement de systèmes dynamiques complexes par RdP PTDS. Petri s'appuie sur MOCA-RP, un moteur de simulation de Monte-Carlo. Ces outils génèrent aléatoirement des scénarios afin de quantifier les grandeurs de fiabilité ou de déterminer des séquences d'événements. La méthode

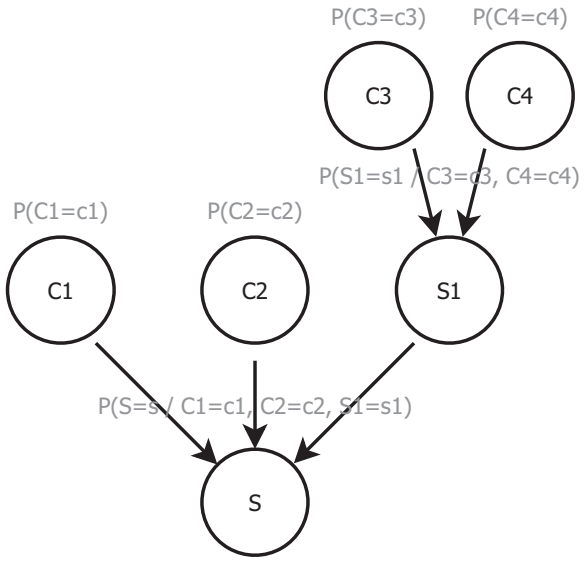

FIGURE 2.8 – Réseau bayésien décrivant le système introduit avec la figure 2.2. Derrière chaque arc, la force de la dépendance d'une variable à ces variables parentes est quantifiée par une distribution de probabilité conditionnelle.

de résolution associée aux RdP est généralement la simulation de Monte Carlo ; des applications existent dans le milieu industriel [Brissaud, 2010], [Brissaud *et al.*, 2012], [Medjoudj, 2006], [Medjoudj et Yim, 2007].

2. Les **réseaux bayésiens dynamiques (RBD)** sont une extension des réseaux bayésiens. Le formalisme des RBD est présenté ici dans son ensemble, afin d'en appréhender la logique dans son intégrité. Le contexte mathématique des RBD est composé d'une méthode de description introduite dans le paragraphe suivant, mais aussi de méthodes d'inférence dédiées à la quantification et spécifiques aux RBD.

Un réseau bayésien, ou modèle graphique probabiliste [Pearl, 1988], [Jensen, 1996], est défini par deux composantes. La composante qualitative est représentée par un graphe orienté sans circuit. Ce graphe est composé de noeuds (représentant les variables aléatoires) et d'arcs (indiquant les dépendances entre ces variables). La composante quantitative ou probabiliste du modèle est constituée par un ensemble de lois de probabilités conditionnelles. Une variable à l'origine d'un arc est dite variable parente de la variable vers laquelle pointe cet arc. La distribution de chaque variable X_i est définie conditionnellement à ses variables parentes dans le graphe, notées $\text{pa}(X_i)$. La figure 2.8 illustre la structure d'un réseau bayésien.

Les réseaux bayésiens sont reconnus dans divers domaines scientifiques et bénéficient d'outils performants pour la modélisation de systèmes complexes [Mechraoui *et al.*, 2010], [Weber et Jouffe, 2006], [Bouissou et Bourreau, 2012].

Un réseau bayésien dynamique (RBD), ou modèle graphique probabiliste markovien [Murphy, 2002] représente la distribution d'une suite de v.a.

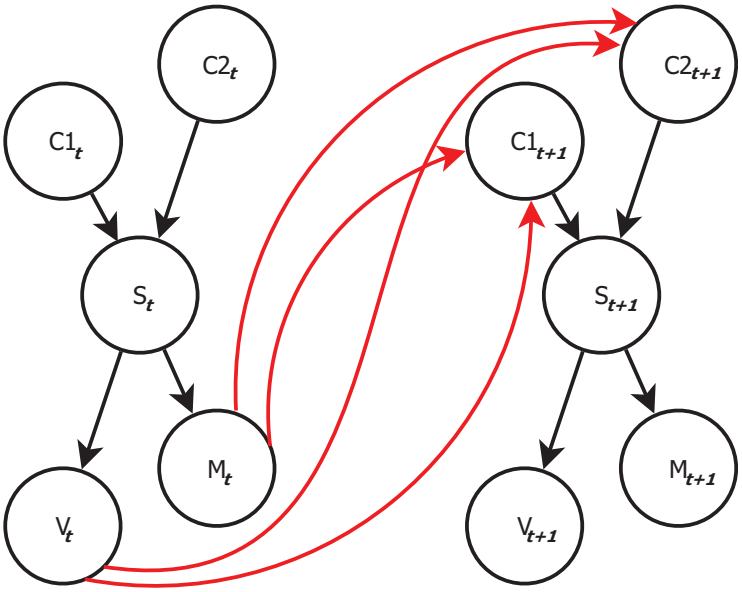

FIGURE 2.9 – Exemple de réseau bayésien dynamique. $C1_t$ et $C2_t$ sont respectivement l'état des composants C1 et C2 à l'instant t. S_t est l'état du système composé de C1 et C2 à l'instant t. De l'état du système S_t dépendent une variable physique V_t et une décision de maintenance M_t. V_t et M_t agissent sur l'état des composants $C1_{t+1}$ et $C2_{t+1}$.

$(\mathbf{X}_t)_{1\leq t\leq T} = (X_{t,1}, ..., X_{t,n})_{1\leq t\leq T}$ à partir d'un réseau bayésien représentant la loi initiale $\mathbb{P}(\mathbf{X}_1)$ et d'un réseau bayésien présentant la loi de transition $\mathbb{P}(\mathbf{X}_t/\mathbf{X}_{t-1})$. On peut alors écrire la loi jointe d'une v.a. conditionnellement à ses parents :

$$\mathbb{P}\left((\mathbf{X}_t)_{1\leq t\leq T}\right) = \prod_{t=1}^{T}\prod_{i=1}^{n}\mathbb{P}\left(X_{t,i}/\mathrm{pa}(X_{t,i})\right). \tag{2.18}$$

Cette modélisation mène à des représentations graphiques telles que celle de la figure 2.9.

[Murphy, 2002] présente le formalisme des RBD comme une généralisation des chaînes de Markov cachées (HMM, de l'anglais *Hidden Markov Models*) [Rabiner, 1989] et des filtres de Kalman [Kalman *et al.*, 1960]. Ce contexte mathématique décrit les interactions entre les variables d'un système dynamique et dispose des méthodes d'inférence qui permettent de quantifier l'évolution de ces systèmes. Toutefois il est possible de décrire un système grâce à un RBD et de le quantifier grâce à une des méthodes présentées dans la section 2.2.2 (méthode de Monte Carlo par exemple).

Il existe des méthodes d'inférence exacte qui consistent à estimer la loi des variables d'intérêt en les éliminant du calcul une à une, en construisant ou non un arbre de jonction [Murphy, 2002]. Optimiser l'ordre d'élimination des variables est un problème NP-difficile (de complexité polynomiale).

Ces méthodes ont montré leur efficacité dans le domaine de la sûreté de fonctionnement, notamment pour les modèles de vieillissement [Donat, 2009], [Donat *et al.*, 2010]. [Bouissou et Bourreau, 2012] répertorie les différentes applications et outils relatifs aux RBD. Leur utilisation dans le cadre de la fiabilité dynamique s'avère délicate en raison de la discrétisation imposée à la variable continue. Il est possible de résoudre des problèmes de fiabilité dynamique lorsque la variable considérée peut être discrétisée en utilisant peu d'intervalles [Varuttamaseni et Lee, 2011], [Tchangani et Noyes, 2005]. En revanche, la méthode d'élimination des variables demande un temps de calcul considérable lorsqu'on essaie d'affiner la discrétisation d'une variable continue afin de rester fidèle à la réalité.

A l'aide des RDB, nous avons décrit avec succès le *benchmark* du *heated tank* évoqué dans la partie 2.1.2.2 [Broy *et al.*, 2011a], [Broy *et al.*, 2011b]. Les résultats du calcul des probabilités des événements indésirables sont satisfaisants dans le sens où ils sont comparables à ceux obtenus avec d'autres méthodes. Cependant, ces calculs sont basés sur la multiplication de matrices dont la taille dépend du nombre de modalités de chaque variable. Cette taille augmente donc si la variable continue est discrétisée avec finesse. Aussi, nous avons jugé que les temps de calcul étaient trop importants pour envisager la modélisation d'un système de taille industrielle. Des pistes ont cependant été identifiées afin d'accélérer les calculs, comme l'exploitation des matrices creuses (essentiellement remplies de zéros à cause des transitions déterministes) ou des matrices définies par blocs (lois de transitions définies par morceaux). Ces pistes n'ont pas été exploitées dans le cadre de ces travaux, au profit des automates stochastiques hybrides. A notre connaissance, elles ne sont pas encore exploitées dans le cadre de la fiabilité dynamique.

3. Les **automates stochastiques hybrides (ASH)** sont encore relativement peu utilisés dans le cadre de la fiabilité dynamique. Cependant cette méthode, décrite en détail dans la thèse [Perez Castaneda, 2009], est représentée dans le cadre du projet Approdyn [Aubry *et al.*, 2012]. Ce projet avait pour but de modéliser le comportement d'un générateur de vapeur.

Un ASH est un automate à états finis temporisé. Chaque mode (état discret) est associé aux équations différentielles qui représentent l'évolution des variables continues. Le caractère hybride de l'automate est assuré par les transitions. Celles-ci peuvent être déterministes (déclenchées par le passage d'un seuil de la variable physique) ou stochastiques (événement aléatoire tel qu'une défaillance).

[Perez Castaneda, 2009] utilise l'environnement open source Scilab / Scicos pour construire l'ASH global décrivant le système étudié dans son intégralité. Cette étape de composition des ASH élémentaires est délicate car elle peut entraîner l'explosion du nombre de modes de l'ASH global. Afin de contourner cette difficulté, [Chraibi, 2013a] utilise des ASH communicant entre eux. Cette méthode est décrite dans la section 3.1.3, ainsi que l'outil associé dans la section 3.3. Dans un cas comme dans l'autre, l'ASH est équipé d'une structure de temps (horloge) et se prête à la simulation de Monte Carlo. Cette horloge est explicite dans les travaux de [Perez Castaneda, 2009] et implicite dans ceux de [Chraibi, 2013a]. Grâce à cette structure temporelle, les temps

d'occurrence des événements aléatoires sont générés, créant ainsi autant d'histoires à analyser pour quantifier le système.

2.2.1.4 Autres méthodes de description

Cet état de l'art ne se veut pas exhaustif. D'autres méthodes existent, mais elles sont peu représentées dans la littérature ou connaissent trop de limites pour être appliquées à un cas-test industriel.

C'est le cas de la méthode GO-FLOW [Siu, 1994], [Labeau *et al.*, 2000], [Aldemir, 2012]. Historiquement, c'est l'une des premières méthodes de fiabilité dynamique. Son formalisme graphique s'inspire des diagrammes de fiabilité, tout en étant enrichi d'opérateurs graphiques spécifiques. La méthode GO-FLOW ne s'applique qu'à des systèmes très simples en raison de l'explosion combinatoire qu'elle provoque.

Les arbres de défaillances dynamiques [Dugan *et al.*, 1992], [Merle, 2010], [Čepin et Mavko, 2002] sont la version dynamique de l'approche par arbre de défaillances décrite dans la section 2.1.1.5. Si le comportement des systèmes décrits grâce à ces méthodes dépend du temps, aucune variable continue déterministe n'est considérée. Ces méthodes ne rentrent donc pas dans le cadre de la fiabilité dynamique.

Une part conséquente de la littérature sur la fiabilité dynamique ne présente pas de nouvelle méthode de description à proprement parler. Ces études [Zio et Maio, 2009], [Zhang *et al.*, 2013], [Aubry *et al.*, 2012] ou [Cabarbaye et Etienne, 2010] décrivent le système grâce à l'association du formalisme des PDMP, des équations différentielles régissant l'évolution des variables physiques, de schémas du système et de schémas fournis par des outils tels que Simulink (Matlab) ou Stateflow (Mathwork). La méthode de description est alors généralement désignée par l'acronyme « PDMP ».

Les chercheurs EDF du département MRI utilisent les bases de connaissances (BdC). Ces dernières sont exploitées dans l'outil de modélisation KB3, basé sur le langage Figaro. KB3 permet de réaliser des représentations graphiques de systèmes à partir des informations contenues dans les BdC (caractéristiques des composants, y compris leur représentation graphique, règles d'occurrence et d'interaction entre les composants). La BdC est munie d'une horloge qui lui permet de suivre dans le temps l'évolution du système et des variables continues qui interagissent avec le système [Bouissou, 2007]. Cette équipe a défini un autre outil, appelé *Boolean logic driven Markov processes* (BDMP) [Bouissou et Bon, 2003]. Si cette méthode est performante pour décrire des systèmes dynamiques complexes, elle n'est pas adaptée au traitement des variables déterministes continues. Les BDMP ne trouvent donc pas leur place dans cet état de l'art.

En revanche, les bases de connaissances permettent, une fois le système et le processus bien ciblés, de décrire l'évolution de ceux-ci de manière automatisée. Cependant, à l'état actuel, le langage Figaro n'est pas adapté à la description d'équations différentielles (ce qui limite la complexité des modèles et nécessite la pose d'hypothèses

simplificatrices). Des travaux en cours, dans lesquels s'inscrit cette thèse, définissent un nouvel outil nommé PyCATSHOO. Cet outil, décrit dans la section 3.3, repose sur le formalisme des automates stochastiques hybrides distribués et sur la programmation orientée objet.

2.2.1.5 Discussion et conclusion

Pour décrire les systèmes dynamiques hybrides, il existe donc une dizaine d'approches qui ne répondent pas forcément aux mêmes objectifs. Le choix d'une méthode suppose un arbitrage sur des critères tels que la taille du système à modéliser, la lisibilité de sa représentation graphique, la complexité des interactions et l'existence ou non d'un outil associé. La plupart de ces méthodes sont associées à une quantification par simulation de Monte Carlo. Certaines sont toutefois compatibles avec des analyses plus proches des équations de Chapman-Kolmogorov, comme un schéma de volumes finis. Ces méthodes de quantification sont décrites dans la section suivante.

Nous avons choisi le formalisme des automates stochastiques hybrides distribués pour décrire les évacuateurs de crues dans le cadre de cette thèse, car ils présentent le double avantage d'un formalisme mathématique riche et d'une représentation graphique intuitive et flexible, ce qui est développé dans la section 3.2. En outre, le développement simultané d'un nouvel outil dédié aux systèmes dynamiques hybrides, appelé PyCATSHOO [Chraibi, 2013a], est également une des raisons de ce choix. PyCATSHOO, élaboré au sein du département MRI, est un outil construit à partir du langage libre Python, dans le but de dépasser les hypothèses liées au langage Figaro pour la modélisation des systèmes dynamiques hybrides. Cet outil, déjà performant pour la modélisation du cas-test du *heated tank*, a besoin d'être validé dans le cadre de l'étude d'un système de taille industrielle.

2.2.2 Méthodes de quantification

Une fois la description qualitative du système réalisée, l'objectif est d'exploiter le modèle afin de réaliser des analyses quantitatives. Plusieurs types d'analyse sont possibles, en fonction du formalisme de modélisation utilisé et des résultats attendus. Les plus répandues sont les méthodes de discrétisation et les méthodes de simulation.

2.2.2.1 Les méthodes de discrétisation

Les méthodes de discrétisation, par opposition aux méthodes de simulation, sont des méthodes de quantification déterministes et exactes au pas de calcul près. Les deux principales méthodes de discrétisation sont les schémas de volumes finis et la *Cell-to-Cell Mapping Technique.*

1. Les **schémas de volumes finis** [Cocozza-Thivent *et al.*, 2006b], [Lair, 2011], [Lair *et al.*, 2011] consistent à discrétiser le temps et l'espace des variables continues en volumes finis. Le pas de discrétisation le mieux adapté au système est déterminé au préalable, pendant la phase de modélisation. Le but est d'estimer les distributions de probabilités marginales $\rho(t)(i, dx)$ que le système soit dans la configuration i et dans le volume dx à l'instant t. Le temps est également discrétisé et des algorithmes calculent ces probabilités de manière récurrente, en évaluant les solutions des équations de Chapman-Kolmogorov. La méthode des volumes finis est donc une méthode de résolution exacte, à la discrétisation près, et permet de prendre en compte n'importe quelle loi de probabilité.

Dans la littérature, quand ils sont testés sur des systèmes de petite taille mais assez complexes (interactions fonctionnelles, redondance passive, ...), ces algorithmes se montrent très performants, apportant des résultats comparables à ceux trouvés avec les simulations de Monte Carlo, mais avec des temps de calcul beaucoup plus courts. Leur convergence est également démontrée. Les méthodes de volumes finis sont aussi plus adaptées que la simulation de Monte carlo pour le calcul d'un indicateur optimum [Lair, 2011], [Lair *et al.*, 2011], et pour les analyses de sensibilité [Mercier et Roussignol, 2008]. Cependant, le temps de calcul devient considérable lorsque la discrétisation des variables continues est affinée dans le but d'être plus fidèle à la réalité.

Afin d'appliquer un schéma de volumes finis à un système ayant plus de cinq composants, [Lair, 2011] expose trois méthodes dont le but est de contourner le manque de place mémoire lié à la dimension de son système. L'une d'elle propose de simplifier la modélisation en réduisant le nombre de variables du processus. Une autre est basée sur une méthode d'approximation.

2. En accord avec la théorie des chaînes de Markov (qui modélise res passages du système d'un état à un autre), la ***Cell-to-Cell Mapping Technique*** **(CCMT)** discrétise l'espace des variables de contrôle continues et définit des probabilités de transition entre ces cellules [Aldemir *et al.*, 2007]. Deux jeux de transition sont alors considérés : les transitions entre deux configurations i et j du système sachant que les variables continues sont dans la cellule $V_{\mathbf{x}}$; les transitions entre deux cellules $V_{\mathbf{x}}$ et $V_{\mathbf{y}}$ sachant que le système est dans la configuration i. Une configuration du système est une combinaison d'états de ses composants.

La CCMT est une méthode de discrétisation permettant de décrire les systèmes, linéaires ou non, de manière discrétisée dans le temps et dans l'espace des états du système. C'est également une méthode basée sur les équations de Chapman-Kolmogorov. La partition de l'espace des variables continues en cellules $V_{\mathbf{x}}$ dépend des événements indésirables. L'évolution du système est décrite par la grandeur $p_{i,\mathbf{x}}(k)$, où $p_{i,\mathbf{x}}(k)$ est la probabilité que les variables continues $\mathbf{x}$ soient dans la cellule $V_{\mathbf{x}}$ au temps $t' = k\,\Delta t$ et que la configuration du système d'être égale à i.

L'application de cette méthode repose sur quelques hypothèses

[Labeau *et al.*, 2000], notamment sur l'indépendance des défaillances des composants. Le principal inconvénient est l'explosion combinatoire du nombre d'états et de cellules, et par conséquent de la taille de la matrice de transition, dès que le nombre de variables augmente ou que les pas de discrétisation sont affinés. Ainsi, [Mandelli *et al.*, 2008] et [Aldemir, 2012] n'utilisent que quatre ou cinq modalités par variable physique afin de limiter la taille des matrices de transition.

La CCMT s'adapte au cadre des EPS américaines grâce à l'outil SAPHIRE. Celui-ci quantifie le système en calculant les probabilités marginales, dans le cadre de petits systèmes, tel qu'un extrait du *benchmark* d'un générateur de vapeur décrit dans le rapport [Aldemir *et al.*, 2007], où seuls trois composants sont étudiés. Les étapes de calcul des probabilités marginales et de la construction de la matrice sont détaillées dans la partie 4 du rapport [Aldemir *et al.*, 2007].

La CCCMT, de l'anglais *Continuous Cell-to-Cell Mapping Technique* [Tombuyses et Aldemir, 1997] est une version continue de la CCMT. La CCCMT consiste à faire tendre le pas de temps Δt vers 0.

2.2.2.2 Les méthodes de simulation de Monte Carlo

La simulation de Monte Carlo [Metropolis et Ulam, 1949] est une méthode très répandue dont le principe est la génération de réalisations de processus stochastiques pour calculer une valeur numérique. Historiquement, la méthode de Monte Carlo a été inventée au milieu du XX$^{\text{ème}}$ siècle pour estimer des intégrales. Cette technique est à présent utilisée dans de nombreux domaines comme la résolution d'équations aux dérivées partielles, la simulation de files d'attente mais aussi dans le management du risque, que ce soit dans les secteurs de la sûreté de fonctionnement ou dans celui de la finance.

Dans le domaine de la sûreté de fonctionnement, cette méthode consiste à simuler un grand nombre d'histoires (trajectoires) du système étudié, puis à approcher la probabilité d'occurrence d'un événement par sa probabilité empirique définie par le ratio

$$\frac{\text{nombre d'apparitions de l'événement}}{\text{nombre d'histoires simulées}}.$$

Cette probabilité d'occurrence caractérise par exemple des indicateurs tels que la fiabilité ou la disponibilité du système. La dispersion autour de la valeur moyenne est également quantifiable. L'avantage de cette méthode est sa grande flexibilité : elle est en effet utilisable quelque soit la complexité ou la taille du système, sous réserve uniquement que ce dernier soit simulable. Toutefois, il est nécessaire de vérifier la convergence des simulations. Par ailleurs, les temps de calcul peuvent être importants, notamment dans le cas d'événements rares, ce qui requiert la génération d'un grand nombre de simulations.

Convergence de la méthode La vitesse de convergence est estimée à partir du théorème-central limite (TCL). La méthode de Monte Carlo a une vitesse de convergence en $1/\sqrt{n}$, où n est le nombre de simulations. Cela signifie que pour pour diviser la largeur de l'intervalle de confiance par M, il faut multiplier le nombre de simulations par M^2. La largeur de l'intervalle de confiance est égale à $2\alpha\sigma/\sqrt{n}$, où σ est l'écart-type et α est un quantile de la loi normale. Pour un intervalle de confiance englobant 95% des données, le quantile d'ordre 2,5% est $\alpha = 1,96$. Il est courant de minorer le nombre de simulations nécessaires par $N = (2\alpha\sigma/l)^2$ où l est la largeur désirée de l'intervalle de confiance.

Dans le cadre de la fiabilité dynamique, deux approches peuvent être utilisées pour générer des simulations de Monte Carlo : un algorithme analogique ou des simulations directes.

1. Un **algorithme analogique** correspond à une simulation numérique des situations physiques réelles [Marseguerra et Zio, 1996], [Kermisch et Labeau, 2000]. Les évolutions possibles du système sont reproduites en alternant les tirages sur les parties aléatoires du problème et les calculs déterministes sur l'évolution du processus continu. La simulation est réalisée en générant un grand nombre d'histoires dont la structure est la suivante pour un calcul de fiabilité :

1. Tirage de l'état initial i et de la valeur initiale x_0 des variables physiques.
2. Echantillonnage de l'instant t de la prochaine transition hors de l'état i partant de x_0.
3. Calcul déterministe de l'évolution dynamique des variables physiques dans l'état i, soit jusqu'à l'instant $\min(t, T)$ où T est l'instant de dépassement de seuil, soit jusqu'à la sortie du domainekermisch2000aD défini dans l'espace des variables. Arrêt si $t > T$.
4. Echantillonnage du nouvel état i' à partir des probabilités de transition hors de l'état courant i. Retour en 2 s'il ne s'agit pas d'un état de panne.

L'estimation de la non-fiabilité du système (ou de la probabilité d'accident dans le cadre des études probabilistes de sûreté) au cours du temps s'obtient en associant à chaque histoire un estimateur binaire, qui vaut 0 tant que le système reste en fonctionnement et 1 dès qu'une défaillance s'est produite, et en moyennant ces résultats sur le nombre d'histoires jouées. On parlera parfois de score pour désigner l'estimation obtenue lors des différentes histoires.

Il est également possible d'estimer d'autres caractéristiques fiabilistes intéressantes, comme le MTTF ou la disponibilité, le MUT et le MDT dans le cas des systèmes réparables.

2. L'algorithme précédent considérait le système dans sa globalité : l'instant de la prochaine transition hors de la configuration i était tirée au sort, avant d'échantillonner la distribution des probabilités de transition vers les configurations du système à partir de i. Une alternative, appelée **simulation directe**, consiste à tirer l'instant de

la prochaine transition de chaque composant. Le minimum de ces différents temps est sélectionné, c'est celui qui détermine l'instant et la nature de la prochaine transition subie par le système global [Kermisch et Labeau, 2000], [Siu, 1994].

La simulation directe conduit souvent à un nombre plus important d'échantillonnages, mais ces derniers concernent en général des distributions plus simples que celle relative au système global. L'équivalence probabiliste des deux approches est démontrée dans [Zio, 1995], [Labeau et Zio, 2002] et [Kermisch et Labeau, 2000].

Limites de la simulation de Monte Carlo
Le principal inconvénient de la simulation de Monte Carlo est le traitementPles événements rares. En effet, si un événement indésirable a une probabilité d'occurrence de l'ordre de 10^{-6}, il ne va être généré en théorie qu'une seule fois lors de la simulation d'un million d'histoires. Il est donc difficile d'obtenir suffisamment d'histoires pour identifier quels événements ont le plus de chances de provoquer cet événement indésirable.

Des solutions existent pour rendre les algorithmes de Monte Carlo plus efficaces. Les plus connues réduisent la variance des estimateurs de Monte Carlo [Glasserman, 2004]. D'autres techniques, plus spécifiques à la sûreté de fonctionnement, favorisent l'apparition des événements rares, comme le biaisage des lois de défaillance, le splitting [Kalos et Whitlock, 1986], [Kermisch et Labeau, 2000] ou le contrôle de poids statistique [Kermisch et Labeau, 2000], [Mandelli *et al.*, 2008].

2.2.2.3 Discussion et conclusion

La quantification d'un système dynamique hybride de taille industrielle réaliste élimine les méthodes de discrétisation, adaptées aux systèmes de petite taille. Si l'événement redouté est défini par l'atteinte d'un seuil précis par la variable déterministe continue, il est impossible de discrétiser cette variable grossièrement. A l'image des systèmes de taille industrielle dans la littérature [Aubry *et al.*, 2012], [Aldemir *et al.*, 2007], [Aldemir, 2012], nous utiliserons les méthodes de Monte Carlo, et plus précisément la simulation directe.

2.2.3 Place de l'information temporelle dans les résultats de fiabilité dynamique

La plupart des études de fiabilité dynamique citées auparavant proposent le même type de format pour leurs résultats. Il s'agit en général de l'évolution de la fiabilité ou de la disponibilité du système dans le temps, de l'évolution de la probabilité d'occurrence de l'événement redouté ou de l'évolution moyenne de la variable déterministe.

Certaines études constatent cependant qu'à chaque composant et chaque instant de défaillance possible pour ce composant correspond une trajectoire différente de la variable déterministe continue. C'est le cas de [Kloos et Peschke, 2006] et

[Peschke et Kloos, 2012] qui modélisent la fiabilité dynamique grâce aux MCDET, mais aucune méthode n'est proposée pour exploiter l'impact des instants de défaillance sur l'issue des simulations.

A partir du même constat, [Zio et Maio, 2009] et [Podofillini *et al.*, 2010] utilisent des techniques de *clustering* pour classer le fruit de leurs simulations. Les *clusters* sont identifiés en fonction des valeurs des variables continues ou en fonction des instants de défaillance, mais toujours sur des systèmes de dimension inférieure à 10 composants.

[Maljovec *et al.*, 2013] utilisent les résultats issus de l'outil ADAPT (basé sur les arbres d'événements dynamiques) en exploitant les propriétés de Morse-Smale. Les données sont projetées sur des cristaux, interfaces graphiques élaborées riches en informations. Chaque cristal est un *cluster* qui est ensuite associé à une distribution des temps de défaillance.

Par ailleurs, fiabilité dynamique et mesures d'importance sont rarement associées. [Cocozza-Thivent, 1997] construit des facteurs d'importance dépendant du temps mais ceux-ci ne sont pas appliqués à des systèmes de grande taille. En revanche, [Tyrväinen, 2013] modélise un système dynamique hybride grâce aux DFM puis calcule l'importance de ses quatre composants sur quelques pas de temps.

A notre connaissance, il n'existe donc pas d'approche ayant pour objectif d'exploiter au mieux l'information temporelle modélisée en fiabilité dynamique. Nous sommes donc amenés à proposer des indicateurs adaptés aux systèmes dynamiques hybrides.

2.3 Conclusion : choix d'une méthodologie et contributions de la thèse

Nous souhaitons proposer une méthodologie qui accompagne l'utilisateur en considérant l'information temporelle tout au long de la modélisation et de l'exploitation des résultats, pour des systèmes dynamiques hybrides de taille industrielle.

L'état de l'art de la section précédente ne propose pas de méthodes répondant à ce critère exigeant. Seules quelques méthodes permettent de décrire un système de taille industrielle. La plus utilisée d'entre elles est basée sur les arbres d'événements dynamiques mais cet outil est généralement associé à des codes de calcul déterministes dédiés aux systèmes de l'industrie nucléaire. En revanche, les Automates Stochastiques Hybrides offrent un formalisme mathématique riche et une représentation graphique intuitive et flexible. Cela permet de décrire des systèmes complexes de dimension importante, sans disposer nécessairement d'un code de calcul déterministe. PyCATSHOO, outil en cours de développement et destiné à être open source, y associe la simulation de Monte Carlo.

Pour répondre à la problématique de la thèse, nous proposons une démarche de modélisation des systèmes dynamiques hybrides de taille industrielle. Cette première

contribution importante de la thèse a permis d'utiliser et de valider le nouvel outil PyCATSHOO. Ce travail repose sur une démarche de modélisation générique grâce aux Automates Stochastiques Hybrides distribués. L'application de cette modélisation à deux évacuateurs de crues a démontré la faisabilité de cette démarche. Cette modélisation s'est ensuite traduite par le développement d'une Base de Connaissances générique, suffisamment générale pour représenter deux évacuateurs différents, mais suffisamment détaillée pour être proche de la réalité.

A partir de la description du système (composants, interactions entre composants, données de fiabilité et données sur la crue), PyCATSHOO génère aléatoirement des histoires. Une histoire est la séquence des états visités par chaque automate le temps d'une crue, associés à la date de chacune de ces transitions.

La deuxième contribution de cette thèse consiste à exploiter de manière adaptée les résultats bruts de simulation, pour une synthèse d'indicateurs de sûreté de fonctionnement de haut niveau. Les histoires sont analysées afin d'extraire l'information temporelle qu'elles comportent, en vue d'obtenir des indications sur la fiabilité des EdC, en particulier des pistes d'amélioration de la sûreté. Cette démarche est rarement associée à la fiabilité dynamique, aussi les indicateurs proposés dans cette thèse seront-ils innovants. Une partie des résultats consiste en l'identification de coupes équivalentesprépondérantes ou du composant le plus important. La classification des histoires en fonction des instants de défaillance des composants, et les possibilités d'analyse en situation qu'elle apporte, constituent l'autre volet des résultats.

Si cette méthodologie a été construite en se basant sur les EdC, il est tout à fait possible de l'adapter à des systèmes relevant d'un autre domaine industriel, conditionnellement à l'élaboration d'une base de connaissances spécifique à la nouvelle catégorie de système étudié. Par exemple dans le contexte nucléaire, et plus particulièrement dans le cadre des EPS de niveau 2.

Chapitre 3

Outils de modélisation pour la sûreté de fonctionnement des évacuateurs de crues

Ce chapitre regroupe les fondements théoriques et techniques de la thèse. La section 3.1 présente les processus dont les propriétés sont à la base de la fiabilité dynamique. Il s'agit des processus markoviens déterministes par morceaux, ou PDMP. Une variante de ces processus, appelée PDMP communicants, facilite la modélisation de systèmes caractérisés par le comportement de nombreux composants. Les ASH décrits dans la section 3.2 permettent d'implémenter ces processus. La version distribuée de ces ASH constituent le socle théorique de l'outil informatique PyCATSHOO, détaillé dans la section 3.3. Notre démarche de modélisation repose donc sur des outils mathématiques dont les propriétés sont utilisées pour justifier l'utilisation d'une méthode de description moins abstraite que le formalisme des PDMP, et le développement d'un outil informatique. Enfin, la section 3.4 expose les principales notations liées aux machines à vecteurs support. Cette technique, appelée couramment SVM (de l'anglais *Support Vector Machine*) n'intervient pas dans le cadre de la modélisation mais sera utilisée dans le chapitre 5, lors de l'exploitation des résultats.

3.1 Les Processus Markoviens Déterministes par Morceaux (PDMP)

La fiabilité dynamique repose sur une modélisation des systèmes dynamiques hybrides par des Processus Markoviens Déterministes par Morceaux. Ces processus, nommés PDMP pour *Piecewise-Deterministic Markov Process*, ont été introduits par Davis en 1984 [Davis, 1984], [Davis, 1993]. Pour introduire le formalisme des PDMP, nous avons choisi d'utiliser les notations de [Lair, 2011], [Lair *et al.*, 2011]. Ces notations sont

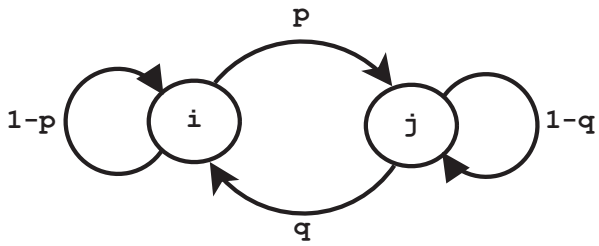

FIGURE 3.1 – Graphe de Markov représentant une chaîne de Markov. La matrice de transition associée est $Q = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}$ où $Q(i,j) = \mathbb{P}(X_n = j / X_{n-1} = i)$.

plus intuitives que celles employées par [Davis, 1984] et présentent l'avantage de définir les PDMP à partir de processus plus classiques employés en fiabilité. En revanche, c'est la définition des PDMP par [Davis, 1984] qui a inspiré [Strubbe et van der Schaft, 2006] pour caractériser les *Communicating Piecewise-Deterministic Markov Process*.

3.1.1 Quelques processus utilisés en fiabilité

Nous reprenons ici les définitions introduites par [Lair, 2011] dans sa thèse. Le lecteur intéressé pourra s'y reporter pour accéder à d'autres définitions ou à des exemples.

3.1.1.1 Chaînes de Markov

Définition 14. Soit E un espace fini et $(X_n)_{n\geq 0}$ une suite de variables aléatoires à valeurs dans E. Soient $n \in \mathbb{N}^*$ et $(i_0, ..., i_n) \in E^n$ tels que $\mathbb{P}(X_{n-1} = i_{n-1}, ..., X_1 = i_1, X_0 = i_0) \neq 0$. Si

$$\mathbb{P}(X_n = i_n / X_{n-1} = i_{n-1}, ..., X_1 = i_1, X_0 = i_0) = \mathbb{P}(X_n = i_n / X_{n-1} = i_{n-1}) \qquad (3.1)$$

alors $(X_n)_{n\geq 0}$ est une **chaîne de Markov**.

Cela signifie que $(X_n)_{n\geq 0}$ vérifie la propriété de Markov faible : toute l'information utile pour la prédiction du futur est contenue dans l'état présent du processus et n'est pas rendue plus précise par les informations provenant du passé. Une chaîne de Markov est caractérisée par la loi initiale de X_0 et par une suite de matrice de transitions $(Q_n)_{n\in\mathbb{N}}$ telle que $Q_{n+1}(i,j) = \mathbb{P}(X_n = j / X_{n-1} = i)$. Si Q_n ne dépend pas de n alors la chaîne de Markov est dite **homogène** et la matrice de transition est notée Q. Les grandeurs fiabilistes comme la disponibilité ou la fiabilité d'un système sont estimées à partir de Q. Souvent utilisées en fiabilité, les chaînes de Markov sont représentées par des graphes de Markov, comme sur la figure 3.1. Les chaînes de Markov ne permettent cependant que la modélisation d'un processus en temps discret. La modélisation d'un processus en temps continu requiert l'usage de processus markoviens de sauts.

3.1.1.2 Processus markoviens de sauts

Définition 15. Soit E un ensemble fini et $(X_t)_{t\geq 0}$ un processus à trajectoires continues à droite avec limites à gauche, à valeur dans E. Soient $n \in \mathbb{N}^*$ et $(t_0, t_1, ..., t_n, t_{n+1})$ tels que $0 \leq t_0 < t_1 < ... < t_n < t_{n+1}$. Soit $(i_0, ..., i_{n+1}) \in E^{n+2}$ tel que $\mathbb{P}(X_{t_0} = i_0, X_{t_1} = i_1, ..., X_{t_n} = i_n) \neq 0$. Si

$$\mathbb{P}(X_{t_{n+1}} = i_{n+1}/X_{t_0} = i_0, X_{t_1} = i_1, ..., X_{t_n} = i_n) = \mathbb{P}(X_{t_{n+1}} = i_{n+1}/X_{t_n} = i_n) \quad (3.2)$$

alors $(X_t)_{t\geq 0}$ est un **processus markovien de sauts** ou PMS homogène.

$(X_t)_{t\geq 0}$ est un processus de sauts (dont les trajectoires sont constantes par morceaux) qui vérifie la propriété de Markov. Soit

$$P_{t_{n+1}-t_n}(i_n, i_{n+1}) = \mathbb{P}(X_{t_{n+1}} = i_{n+1}/X_{t_n} = i_n). \quad (3.3)$$

$(P_t(i,j))_{i,j\in E}$ est appelé **noyau de transition** du PMS au temps t. Le taux de transition, constant, de l'état i vers l'état j est noté $A(i,j)$ et on pose $A(i,i) = -\sum_{j\in E\setminus\{i\}} A(i,j)$. Un PMS est caractérisé par la loi initiale de X_0 et par la matrice génératrice A. La figure 3.2 représente la différence entre chaîne de Markov et PMS.

3.1.1.3 Processus de renouvellement

Définition 16. Soit $(X_n)_{n\in\mathbb{N}}$ une suite de v.a. positives, indépendantes et identiquement distribuées. Soit $S_0 = 0$ et pour $n \geq 1$, $S_n = \sum_{k=1}^{n} X_k$. La suite $S = (S_n)_{n\in\mathbb{N}}$ est appelée **processus de renouvellement**. Les temps S_n sont appelés **temps de renouvellement**. Soit $\begin{array}{ccc} E & \to & \{0,1\} \\ x & \mapsto & \mathbf{1}_A(x) \end{array}$ la fonction indicatrice, qui retourne 1 si $x \in A$ et 0 si $x \notin A$, avec $A \subseteq E$. Le **processus de comptage** associé $(N_t)_{t\in\mathbb{R}^+}$ est défini par $N_t = \sum_{n=1}^{+\infty} \mathbf{1}_{]0,t]}(S_n)$, comptant ainsi le nombre de renouvellements survenus jusqu'au temps t.

3.1.1.4 Processus de renouvellement markovien

Un processus de renouvellement markovien (PRM) est le couplage d'une chaîne de Markov et d'une suite de v.a. représentant les instants de saut de la chaîne.

Définition 17. Soit E un ensemble fini. Soit $(Y_n)_{n\geq 0}$ un processus à valeurs dans E et $(T_n)_{n\geq 0}$ une suite croissante de variables positives. Soient $n \geq 0$, $(i_0, i_1, ..., i_{n-1}, i, j) \in E^{n+2}$ et $(t, t_1, ..., t_n) \in \mathbb{R}_+^{n+1}$. Si

$$\mathbb{P}\left(Y_{n+1} = j, T_{n+1} - T_n \leq t/Y_0 = i_0, Y_1 = i_1, ..., Y_n = i, T_n = t_n\right)$$

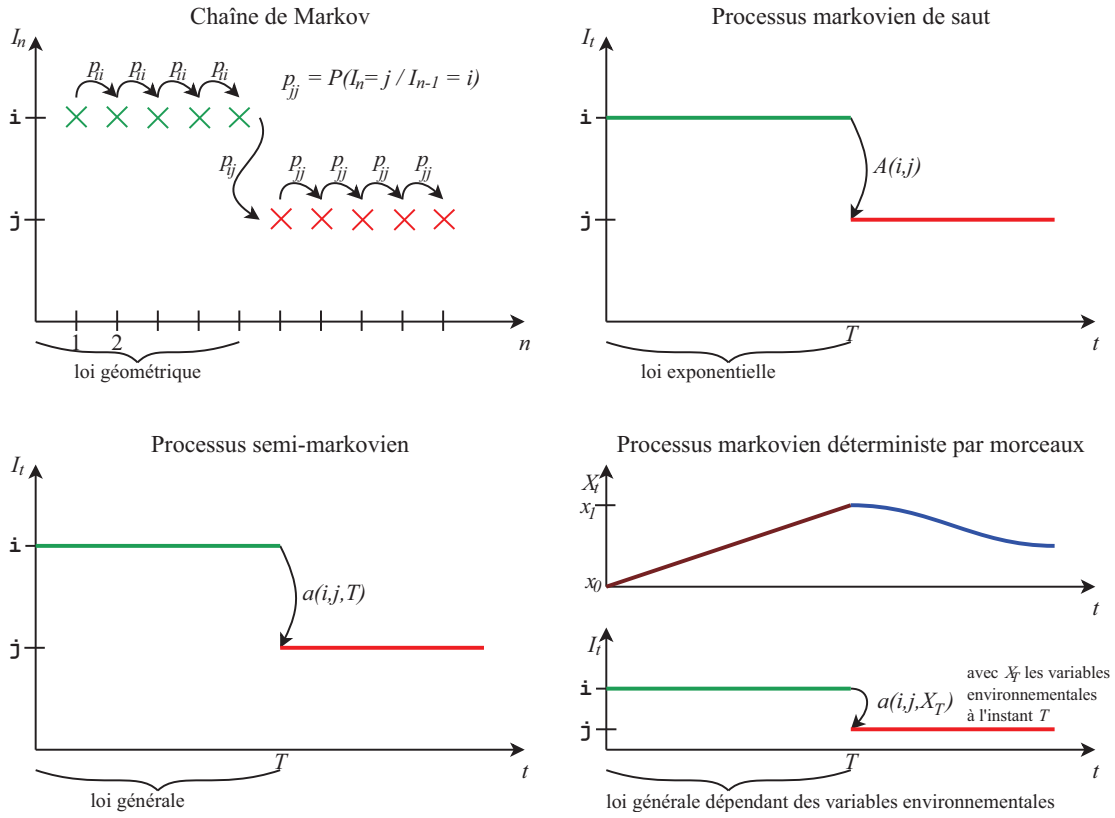

FIGURE 3.2 – Construction d'un PDMP à partir de processus stochastiques classiques

$$= \mathbb{P}\left(Y_{n+1} = j, T_{n+1} - T_n \leq t / Y_n = i\right) \tag{3.4}$$

alors $(Y, T) = (Y_n, T_n)_{n\geq 0}$ est un **processus de renouvellement markovien** homogène. Soit

$$Q(i, j, t) = \mathbb{P}\left(Y_{n+1} = j, T_{n+1} - T_n \leq t / Y_n = i\right) = \int_0^t Q(i, j, du). \tag{3.5}$$

$Q = (Q(i, j, dt))_{i,j\in E}$Pest appelé **noyau semi-markovien** du processus (Y, T).

Les processus de renouvellement markoviens permettent de définir un nouveau type de processus, plus général que les processus markoviens dans le sens où la durée entre deux sauts suit une loi plus générale que la loi exponentielle.

3.1.1.5 Processus semi-markovien

Définition 18. Soit E un ensemble fini. Soit (Y, T) un processus de renouvellement markovien à valeurs dans E tel que $\sup_{n\in\mathbb{N}} T_n = \infty$. Pour tout $t \in [T_n, T_{n+1}[$, on pose $X_t = Y_n$. Le processus $(X_t)_{t\geq 0}$ est le **processus semi-markovien** (PSM) associé à

(Y, T) et la chaîne de Markov Y est la **chaîne de Markov immergée**. La figure 3.2 représente la différence entre PMS et PSM.

3.1.2 Les Processus Markoviens Déterministes par Morceaux (PDMP)

Dans la littérature co-existent deux définitions des PDMP. Historiquement, la première a été introduite par Davis en 1984 [Davis, 1984], [Davis, 1993]. La deuxième résulte des travaux de Cocozza [Cocozza-Thivent, 1997], [Cocozza-Thivent *et al.*, 2006b], [Cocozza-Thivent, 2012], [Lair, 2011], [Lair *et al.*, 2011]. Les PDMP communicants forment le socle théorique des automates stochastiques hybrides. Bien que les PDMP communicants aient été construits à partir de la définition de Davis, nous présentons également les notations introduites par Cocozza *et al.*

3.1.2.1 Définition de Cocozza *et al.*

Dans les travaux de cette équipe, un PDMP est défini comme un couple $(I_t, X_t)_{t\geq 0}$. I_t est la variable discrète aléatoire représentant la configuration du système. X_t est la variable (ou le vecteur de variables) déterministe(s) continue(s), appelées également variables physiques ou variables environnementales. Avec ces notations, la définition du processus est plutôt intuitive, et sa construction découle des propriétés de processus plus classiques. Les propriétés des PDMP ainsi définis sont notamment plus adaptées à la résolution des équations de Chapman-Kolmogorov afin d'estimer leur probabilité de distribution marginale, ou au traitement de problèmes d'arrêt optimal [De Saporta *et al.*, 2010].

Définition 19. Soit le processus $(I_t, X_t)_{t\geq 0}$. I_t représente la configuration du système ; X_t caractérise les d variables déterministes, continues entre deux sauts de I_t. Le processus $(I_t, X_t)_{t\geq 0}$ saute à des instants isolés et les deux composantes interagissent mutuellement l'une sur l'autre. La transition de (i, x) vers (j, y) est régie par le noyau de transition τ avec $\tau(i, x\,; j, dy) = a(i, j, x)\mu_{(i,j,x)}(dy)$ où $a(i, j, X_t)$ est le taux de transition entre les configurations i et $j \in E$ dépendant de la variable environnementale X_t. $\mu_{(i,j,x)}(dy)$ est la loi de distribution de la variable continue, à la suite du saut de la configuration i vers j, dépendant de la valeur de la variable continue X_t^- juste avant le saut. Entre deux sauts, la variable I_t est constante et l'évolution de X_t, déterministe, est solution de l'équation différentielle

$$\frac{dy}{dt} = v(i, y). \tag{3.6}$$

On note $g(i, x, t)$ l'unique solution de 3.6 tel que

$$\forall (i, x) \in E \times \mathbb{R}^d, \quad g(i, x, 0) = x. \tag{3.7}$$

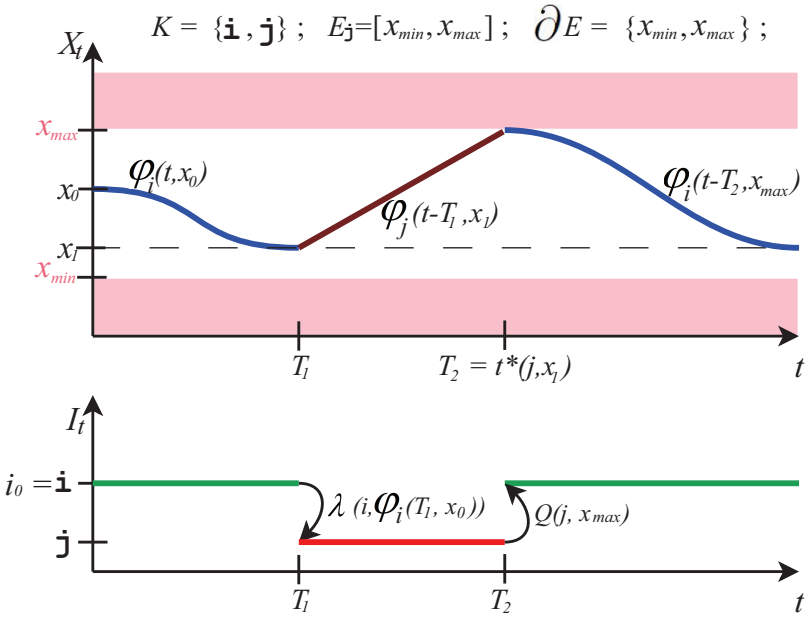

FIGURE 3.3 – Exemple de trajectoire d'un PDMP. Le premier saut est provoqué par un événement aléatoire, le deuxième saut correspond à un passage de seuil par la variable X_t.

Le processus $(T_n, (I_{T_n}, X_{T_n}))_{n\geq 0}$ est un processus de renouvellement markovien défini par son noyau Q. Le **processus markovien déterministe par morceaux** $(I_t, X_t)_{t\geq 0}$ est construit en posant $\forall t \in [T_n, T_{n+1}[$, $I_t = I_{T_n}$ et $X_t = g(I_{T_n}, X_{T_n}, t-T_n)$. La figure 3.2 représente la différence entre PSM et PDMP.

3.1.2.2 Définition de Davis

La définition de Davis est mieux adaptée à la modélisation de sauts causés par les passages de seuil de la variable déterministe.

Définition 20. Soit K un espace d'états fini.

Pour tout $\nu \in K$, $d(\nu) \in \mathbb{N}$ désigne le nombre de variables déterministes continues en interaction avec la configuration ν.

Pour tout $\nu \in K$, E_ν est un sous-ensemble de $\mathbb{R}^{d(\nu)}$ et $g_\nu : \mathbb{R}^{d(\nu)} \to \mathbb{R}^{d(\nu)}$ est une fonction continue localement lipschitzienne sur E_ν.

Le flux $\varphi_\nu(t,x)$ est la solution de l'équation différentielle $\frac{d\zeta}{dt} = g_\nu(x)$ tel que $\varphi_\nu(0,x) = x$.

L'espace d'états hybride du PDMP est défini par

$$E = \{(\nu, x)/\nu \in K,\, x \in E_\nu\}\,. \tag{3.8}$$

Soit ∂E_ν la frontière de E_ν.

Pour tout $(\nu, x) \in E$, on pose

$$t^*(\nu, x) = \begin{cases} \inf \{t > 0 / \varphi_\nu(t, x) \in \partial E_\nu\} \\ \infty \text{ si } \{t > 0 / \varphi_\nu(t, x) \in \partial E_\nu\} = \emptyset \end{cases} . \tag{3.9}$$

$t^*(\nu, x)$ est l'instant d'atteinte de la frontière ∂E_ν sous les conditions initiales (ν, x). Les sauts du PDMP sont déterminés par une fonction de transition λ et une mesure de transition Q. $\lambda : E \to \mathbb{R}^+$ est une fonction mesurable tel que $\forall (\nu, x) \in E$, il existe une valeur $\varepsilon(\nu, x) > 0$ telle que la fonction $s \to \lambda(\nu, \varphi_\nu(s, x))$ soit intégrable sur $[0, \varepsilon(\nu, x)[$.

Soit ∂E la frontière de E et $\mathcal{E}$ l'ensemble des tribus boréliennes de E. $\mathcal{P}(E)$ est l'ensemble des mesures de probabilités sur l'espace borélien $(E, \mathcal{E})$.

$Q : E \cup \partial E \to \mathcal{P}(E)$ est une fonction qui à tout $x \in E \cup \partial E$ associe la distribution de probabilité $A \in \mathcal{E}$.

A partir de l'état initial (ν_0, x_0), deux dynamiques sont engagées. D'une part $t^*(\nu_0, x_0)$ désigne le premier instant à partir duquel le processus va évoluer en dehors de sa frontière ∂E_{ν_0}. Ce passage de seuil va provoquer un changement de la configuration du système à partir de (ν_0, x_0). D'autre part, un instant de saut est déterminé à partir du taux de transition λ. Dans un cas comme dans l'autre, la nouvelle configuration du système ainsi que les nouvelles valeurs de la variable physique vont être déterminées à partir de Q.

3.1.3 Les PDMP communicants (CPDMP)

Définition 21. Un **PDMP communicant**, ou CPDMP (de l'anglais *Communicating PDMP*) [Strubbe et van der Schaft, 2006] est un 10-tuple $(L, Y, \nu, Inv, G, \Sigma, B, P, S, C)$ où

- L est un ensemble fini d'états ;
- Y est un ensemble de variables déterministes continues. Une variable $y \in Y$ peut éventuellement être de dimension $d(y) \geq 1$, et est donc définie sur $\mathbb{R}^{d(y)}$;
- $\nu : L \to 2^Y$ est une fonction qui à chaque configuration $l \in L$ associe un certain nombre de variables physiques, ce qui correspond à une combinaison d'éléments de Y ;
- InvStandard fonction qui à chaque configuration $l \in L$ et à chaque variable $y \in \nu(l)$ associe un sous-ensemble de $\mathbb{R}^{d(y)}$. $Inv(l, y) \subseteq \mathbb{R}^{d(y)}$ représente l'ensemble des valeurs que peut prendre la variable y sans déclencher un saut par dépassement de seuil. $Inv_l = \{Inv(l, y) / y \in \nu(l)\}$. ∂Inv_l est la frontière de Inv_l et $\partial Inv(l, y)$ est la frontière de $Inv(l, y)$;
- G est une fonction qui à chaque couple (l, y) associe une fonction continue localement lipschitzienne appelée **flux** : $t, y_0 \mapsto \varphi_{l,y}(t, y_0)$;

- Σ est un ensemble d'**étiquettes de communication**. $\bar{\Sigma}$Fest appelé **miroir passif** de $\Sigma : \bar{\Sigma} = \{\bar{a}/a \in \Sigma\}$;
- B est un ensemble fini de **transitions de seuil**. $\forall b \in B$, $b = (l, a, l', R)$ où l et l' sont les états de départ et d'arrivée, a est l'étiquette de la transition et R est la fonction de réinitialisation. Après le passage de l vers l', R distribue les valeurs possibles $R^y(x)$ de chaque variable $y \in \nu(l)$ en fonction de la valeur x juste avant le saut ;
- P est l'ensemble des **transitions passives**. $\forall p \in P$, $p = (l, \bar{a}, l', R)$. Lorsqu'une transition $b \in B$ d'étiquette a a lieu, un message est adressé aux transitions de P, ce qui déclenche le passage des transitions p dont l'étiquette est $\bar{a}$;
- S est un ensemble fini de **transitions spontanées**. $\forall s \in S$, $s = (l, \lambda, a, l', R)$. λ est un taux de transition dont les propriétés sont les mêmes que pour un PDMP ;
- C est une **fonction de choix** qui à chaque point-frontière (l, x) associe une distribution de probabilité sur B_l, où $B_{l^*} = \{b = (l, a, l', R) \in B/l = l^*\}$. C permet de déterminer quelle transition sera franchie à partir de l si B_l contient plus d'une transition.

[Strubbe et van der Schaft, 2005] montre l'équivalence des PDMP et des CPDMP. Les CPDMP possèdent donc les propriétés des PDMP. Des opérations de compositions de deux CPDMP ont également été définies par [Strubbe et van der Schaft, 2006].

Les CPDMP sont mieux adaptés que les PDMP à la modélisation de systèmes caractérisés par le comportement de nombreux composants. En effet, le formalisme lié aux étiquettes facilitent la communication et la synchronisation entre les processus sous-jacents. Les Automates Stochastiques Hybrides reposent sur le formalisme mathématique des CPDMP et permettent l'implémentation de ces processus.

3.2 Automates Stochastiques Hybrides (ASH)

3.2.1 De la théorie des automates aux ASH

Les automates [Baptiste et Bournez, 2009] sont des objets mathématiques permettant de modéliser des systèmes informatiques. La théorie des automates se base sur la théorie du langage, mais aussi sur la théorie des graphes. Les automates ont de nombreuses applications telles que la recherche d'occurrrences dans un texte, la génomique ou la modélisation des systèmes à événements discrets en automatique par exemple.

Définition 22. Un **automate fini déterministe** est un quintuplet $(Q, \Sigma, \delta, q_0, F)$ où

- Q est un ensemble fini d'états ;
- Σ est un alphabet fini ; un alphabet est un ensemble de caractères et de symboles ;
- $\delta : Q \times \Sigma \to Q$ est la fonction de transition ;

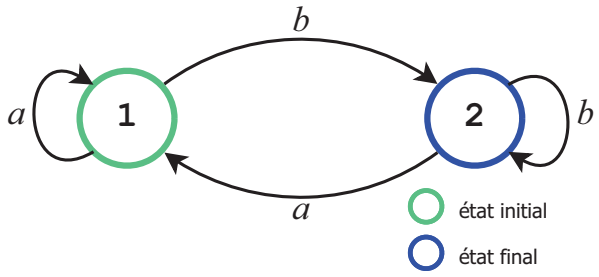

FIGURE 3.4 – Exemple d'automate fini déterministe. Cet automate est conçu pour reconnaitre tous les mots qui contiennent au moins un b.

- q_0 est l'état initial ;
- $F \subseteq Q$ est l'ensemble des états finaux.

Un automate peut être représenté de manière intuitive par un graphe orienté : les éléments de Q sont matérialisés par des sommets et ceux de Σ par des arcs (ou arêtes) étiquetés. L'état initial et les états finaux sont également indiqués par des éléments graphiques. La figure 3.4 représente un automate fini déterministe.

Définition 23. Un **automate hybride** [Henzinger, 2000] est également représenté par un quintuplet $(X, (V, E), \{init, inv, flow\}, jump, \Sigma)$ où

- X est l'ensemble des variables continues étudiées ;
- (V, E) est un graphe orienté composé d'un ensemble de sommets V et d'un ensemble d'arcs E ;
- pour chaque sommet $v \in V$, $init(v)$ est la valeur initiale des variables X à l'entrée dans le sommet v ;
- $inv(v)$ est le sous-ensemble invariant de $\mathbb{R}^n$ dans lequel doivent se trouver les valeurs de X pour ne pas quitter le sommet v ;
- $flow(v)$ représente la dynamique liée au sommet v : c'est l'ensemble des équations différentielles qui régissent l'évolution des variables X pour le sommet v ;
- $jump(e)$ est un sous-ensemble de $\mathbb{R}^n$ dans lequel doivent se trouver les valeurs de X pour déclencher le saut $e \in E$;
- Σ est l'ensemble des événements déclenchant chaque transition $e \in E$.

La figure 3.5 donne un exemple d'automate hybride.

Les transitions d'un automate hybride ne sont donc plus seulement fondées sur un alphabet. En définissant ainsi un automate dont les transitions sont basées sur des événements discrets comme sur l'évolution de variables différentielles (via le passage de seuil de ces variables), [Henzinger, 2000] définit un cadre théorique riche pour l'évolution de systèmes dynamiques hybrides. La théorie des automates hybrides formalise aussi la composition d'automates hybrides. Parmi eux, les automates hybrides linéaires caractérisent des variables dont la dérivée par rapport au temps est constante. Les automates hybrides rectangulaires sont des automates hybrides linéaires si la dérivée de la

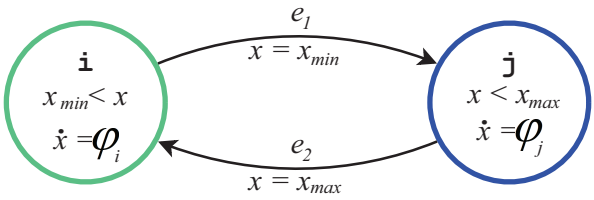

FIGURE 3.5 – Exemple d'automate hybride. Avec les notations précédentes : $X = \{x\}$, $V = \{i, j\}$, $E = \{e_1, e_2\}$, $inv(i) =]x_{min}, \infty]$, $inv(j) =]-\infty, x_{max}]$, $flow(i) = \{\varphi_i\}$, $flow(j) = \{\varphi_j\}$, $jump(e_1) = \{x_{min}\}$, $jump(e_2) = \{x_{max}\}$, $\Sigma = \{\{x = x_{min}\}, \{x = x_{max}\}\}$.

variable continue par rapport au temps est contenue dans un intervalle de forme $[a, b]$. Leurs propriétés dépendent de la forme des équations différentielles qui régissent leur évolution et de la forme des ensembles *init* et *inv*.

Les automates stochastiques hybrides sont des automates hybrides dont les transitions ont également un caractère stochastique. Une transition est donc déclenchée parce qu'une variable continue a atteint un certain seuil ou parce qu'il s'est produit un événement aléatoire occasionnant un changement d'état.

A partir des définitions données par [Henzinger, 2000] et [Julius, 2006], [Chraibi, 2013a] introduit une définition des automates stochastiques hybrides.

Définition 24. Un **automate stochastique hybride** (ASH) est un 6-tuple $A = (M, X, (m_0, x_0), f, inv, T)$ où

- M est un ensemble de **modes** (états discrets) ;
- $X \subseteq \mathbb{R}^n$ est un espace d'états continu ;
- $m_0 \in M$ est le mode initial et $x_0 \in X$ est la valeur initiale de la (les) variable(s) déterministe(s) continue(s) ;
- $f : M \to \mathcal{C}(\mathbb{R}, X)$ est une fonction qui à chaque mode m associe la fonction f_m. $f_m = \frac{d\phi_m}{dt}$ caractérise la dynamique continue pour le mode m ;
- inv est une fonction qui à chaque mode $m \in M$ associe son **sous-ensemble invariant** $Inv(m)$ de X. La réalisation de l'événement $\{\exists t \in \mathbb{R}^+ / \phi_m(t) \notin Inv(m)\}$ déclenche au temps t une transition $\tau \in T$ à partir de m ;
- T est un ensemble de transitions.

Définition 25. Une **transition** $\tau \in T$ est définie par un 6-tuple $(m, m', \lambda_\tau, R_\tau, guard_\tau)$ où

- m est le mode source et m' est le mode destination ;
- $\lambda_\tau : Inv(m) \to \mathbb{R}$ est une fonction de taux de transition qui caractérise la distribution de probabilité de la transition τ. λ_τ est par exemple un taux de défaillance qui peut dépendre de la valeur des variables continues, comme dans le *benchmark* du *heated tank* [Chraibi, 2013a] ;

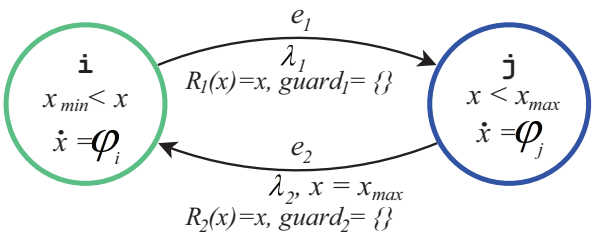

FIGURE 3.6 – Exemple d'automate stochastique hybride. Cet ASH représente le PDMP dont une trajectoire est tracée sur la figure 3.3. La transition e_1 de i vers j est seulement provoquée par un événement aléatoire de taux de transition λ_1. La transition e_2 de j vers i est provoquée, soit par un événement aléatoire de taux de transition λ_2, soit par le passage du seuil x_{max} par la variable x. Avec les notations précédantes : $M = \{i, j\}$, $X = [x_{min}, x_{max}]$, $m_0 = i$, $f_i = \frac{d\varphi_i}{dt}$, $f_j = \frac{d\varphi_j}{dt}$, $inv(i) =]x_{min}, \infty]$, $inv(j) =]-\infty, x_{max}]$, $T = \{e_1, e_2\}$, $e_1 = \{i, j, \lambda_1, R_1, guard_1\}$ avec $R_1(x) = x$ et $guard_1 = \emptyset$; $e_2 = \{j, i, \lambda_2, R_2, guard_2\}$ avec $R_2(x) = x$ et $guard_2 = \emptyset$.

- R_τ est une fonction de réinitialisation des variables continues suite à la transition τ. Cette réinitialisation est continue ou aléatoire ;
- $guard_\tau$ est la **garde** de τ, c'est-à-dire l'ensemble des conditions dont la vérification est nécessaire pour que la transition τ ait lieu.

La figure 3.6 illustre un exemple d'ASH.

3.2.2 Composition et synchronisation des ASH

Des travaux récents [Aubry *et al.*, 2012], [Perez Castaneda, 2009] utilisent les ASH pour résoudre des problèmes de fiabilité dynamique. Le principe consiste à construire un ASH élémentaire pour chaque composant du système étudié, ainsi que l'ASH régissant l'évolution de la variable physique. La principale difficulté réside dans l'étape de synchronisation des ASH élémentaires pour construire l'ASH global décrivant le système dans son intégralité. La synchronisation résulte d'une composition des ASH. Cette composition conserve les propriétés des ASH, mais le nombre de modes de l'ASH global explose rapidement. Pour contourner cette difficulté, nous utiliserons la synchronisation par communication entre automates distribués, grâce à la théorie des CPDMP et à l'outil PyCATSHOO.

3.3 L'outil PyCATSHOO

PyCATSHOO (PythoniC AuTomates Stochastiques Hybrides Orientés Objets) est une plate-forme outils dédiée à l'estimation des grandeurs fiabilistes des systèmes dynamiques hybrides. PyCATSHOO repose sur le formalisme des PDMP, sur le paradigme

multi-agents et sur la programmation fonctionnelle et orientée objet. Le langage scientifique Python, libre et open-source, est un langage intermédiaire [pyt, 2013]. En ce sens, il permet de combiner les avantages des langages compilés et ceux des langages interprétés.

La modélisation des systèmes dynamiques hybrides avec PyCATSHOO repose sur trois niveaux de programmation. Le premier niveau est le « socle » de PyCATSHOO. C'est le logiciel à proprement parlé, constitué de 18 fichiers « source ». Le deuxième niveau est une base de connaissances construite pour détailler les caractéristiques de chaque type de composants d'une classe de systèmes. Le troisième niveau est le modèle décrivant un système particulier. L'architecture de PyCATSHOO s'articule donc en un ensemble de classes Python ayant pour but :

- la modélisation du système global,
- la définition de bases de connaissances qui répertorient les types de composants et leurs interactions possibles,
- le suivi (*monitoring*) de la simulation de Monte Carlo et le traitement des résultats.

3.3.1 Le logiciel PyCATSHOO

Le logiciel PyCATSHOO est constitué de 18 classes. Chacune de ces classes détermine le fonctionnement d'une composante particulière du modèle. Dans certaines classes sont définis des éléments tels qu'un état, une transition, un composant, une boîte à messages ou un lien. La classe *PDMPController* implémente un "contrôleur de PDMP". C'est un automate stochastique hybride dont la création et la paramétrisation se fait à l'aide de méthodes dédiées. Ce contrôleur permet de piloter l'évolution des variables continues en prenant en charge la résolution des équations différentielles qui les gouvernent. Il permet également de piloter l'interaction avec les comportements stochastiques discrets des différents composants du système. D'autres classes précisent les paramètres de simulation, telles que les différentes lois autorisées ou les ressources. Ces simulations reposent sur les générateurs de nombres aléatoires du package d'aide à la simulation Simpy de Python. Finalement, des fichiers coordonnent et synchronisent la simulation dans sa globalité, et enregistrent l'évolution de ces simulations.

3.3.2 Construction d'une base de connaissances

Une **base de connaissances** (BdC) répertorie dans des classes les caractéristiques de chaque type de composant, pour une catégorie de système donnée. Nous avons ainsi construit la BdC « évacuateur de crues ». Toutes les classes PyCATSHOO (PyC) élaborées dans ce contexte héritent de la classe *Component* définie dans le fichier source du même nom.

Une classe PyC est un triplet $(\mathcal{S}, RMB, SMB)$ où

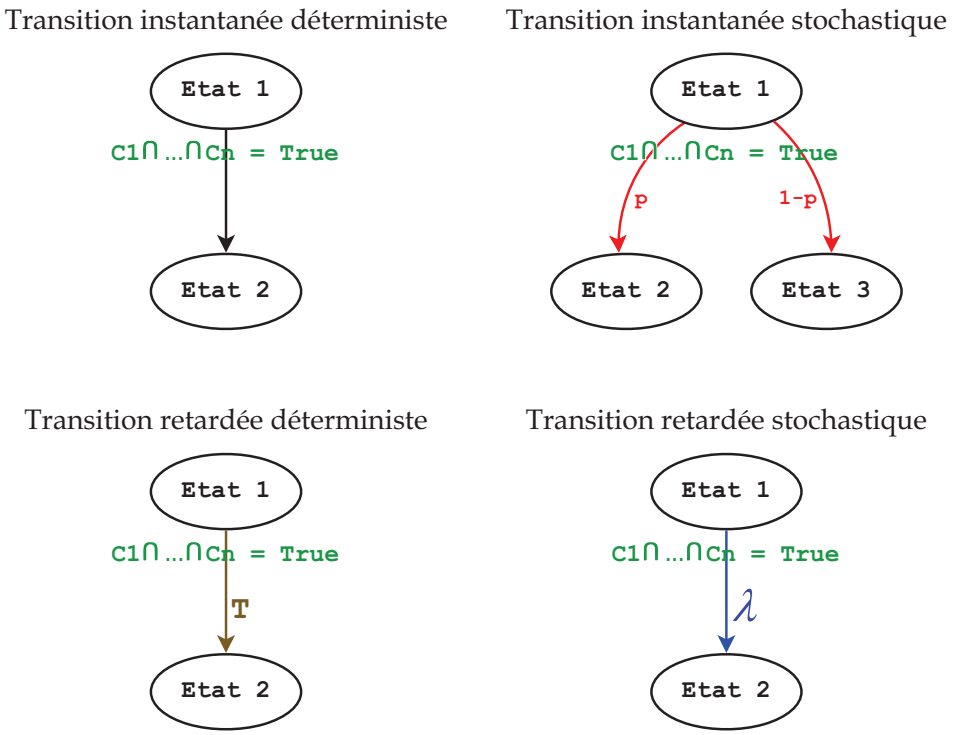

FIGURE 3.7 – Représentation des quatre types de transitions.

– $\mathcal{S}$ est un ensemble d'automates stochastiques discrets ou hybrides, muni d'états, de transitions entre ces états, impliquant des lois et des conditions.
La description d'un objet par plusieurs automates est possible. Dans ce cas, deux états appartiennent à deux automates distincts s'il n'existe pas de chemin composé de transitions pour les relier. Toutefois, deux automates distincts d'un même objet ne sont pas forcément indépendants. En effet, il est possible de conditionner une transition dans un automate par l'activation d'un état dans un autre automate, ce qui est illustré sur la figure 3.9.
Dans un automate hybride, le vecteur des variables déterministes continues est également défini sous la forme d'un sous-automate stochastique caché, avec son évolution déterministe (équation différentielle), ses frontières et ses sauts ;
– *RMB* (*Receiving Message Box*) est un ensemble de boîtes à messages destinataires (`IN`) qui reçoivent les informations provenant des autres classes. Ces boîtes sont paramétrées par une méthode à appeler à la réception d'un message. Cette méthode décode les informations entrantes et certains attributs internes des objets concernés sont mis à jour en fonction des valeurs reçues. Ces modifications d'attributs ont notamment pour conséquence la validation ou l'invalidation des conditions de certaines transitions. Des automates sont ainsi susceptibles de changer d'état.
– *SMB* (*Sending Message Box*) est un ensemble de boîtes à messages expéditrices

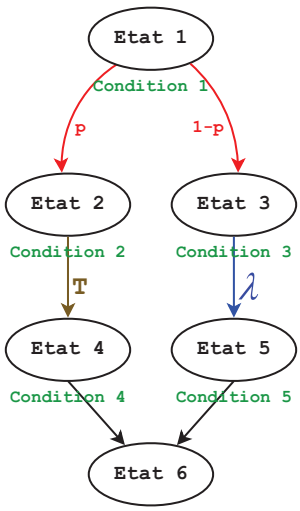

FIGURE 3.8 – Exemple d'automate utilisant les quatre types de transitions.

(`OUT`) qui servent à envoyer des informations aux autres classes. Ces messages sortants sont envoyés lorsqu'un automate entre dans un état particulier.

Les boîtes à messages représentent la seule manière de communiquer entre les objets qui constituent un système. La nature des informations qu'ils reçoivent ou émettent et les conditions de ces échanges doivent être précisément spécifiées (fonctionnement par contrat) pour assurer la clarté et la justesse de la modélisation. Ces informations sont utiles pour l'implantation des capacités de communication des CPDMP.

3.3.2.1 Les différents types de transitions

Les transitions entre états ne se déroulent pas toutes de la même façon. Elles sont instantanées ou retardées, déterministes ou stochastiques. Nous définissons donc quatre types de transitions.

Soit (C_1, ..., C_n) un ensemble de conditions qui doivent être vérifiées afin d'exécuter la transition. L'intersection $C_1 \cap ... \cap C_n$ est donc une expression booléenne.

Transition instantanée déterministe. Dès que l'ensemble de conditions est vérifié, la transition est exécutée et l'automate change d'état. Une transition instantanée est représentée graphiquement par une flèche noire.

Transition instantanée stochastique. Dès que l'ensemble de conditions est vérifié, un tirage est réalisé selon une loi discrète (généralement, la loi binomiale). La transition est exécutée et l'état destination de l'automate est choisi parmi un ensemble d'états, en fonction du résultat du tirage. Une transition instantanée stochastique est représentée graphiquement par au moins deux flèches rouges.

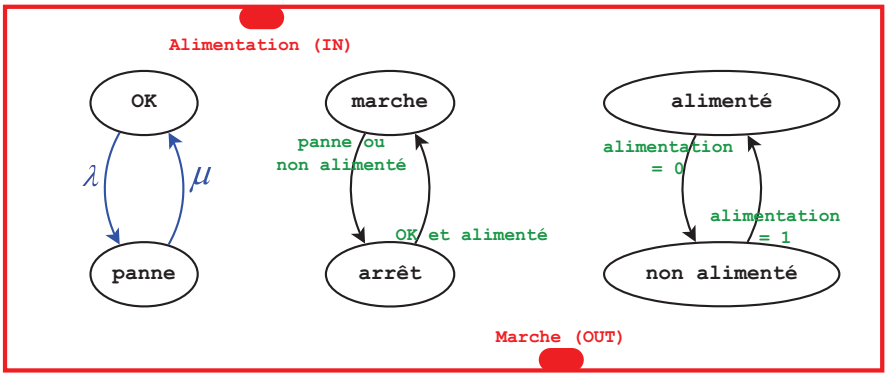

FIGURE 3.9 – Objet PyCATSHOO décrit par trois automates. λ est le taux de défaillance et μ le taux de réparation. Les transitions entre les états **`marche`** et **`arrêt`** sont conditionnées par les états des autres automates. L'information « alimentation » provient d'une boîte à message **`IN`**. L'information « état **`marche`** actif » est transmise par le biais de la boîte à messages **`OUT`**.

Transition retardée déterministe. Dès que l'ensemble de conditions est vérifié, un compte à rebours se déclenche, pour un délai fixé. A la fin de ce délai, la transition est exécutée. Une seconde variante de ce type de transition existe. Dans cette variante, le compte à rebours est interrompu si la condition n'est plus vérifiée. Une transition retardée déterministe est représentée graphiquement par une flèche marron.

Transition retardée stochastique. Dès que l'ensemble de conditions est vérifié, un tirage est réalisé selon une loi continue. La loi exponentielle et la loi de Weibull sont disponibles pour la génération de durées dans PyCATSHOO. Néanmoins il est aisé de programmer d'autres lois dans le fichier source correspondant. Une fois ce tirage effectué, un compte à rebours se déclenche et prend fin au bout de cette durée générée aléatoirement. A la fin de ce délai, si la condition est toujours valide, la transition est exécutée. Une transition retardée stochastique est représentée graphiquement par une flèche bleue.

La figure 3.7 représente les quatre types de transitions et introduit le code couleur qui sera utilisé dans la suite du manuscrit, lors de la modélisation de l'évacuateur de crues.

3.3.2.2 Contrôle de la variable continue

La variable physique doit être l'attribut de l'un des objets. Cet objet va donc « héberger », dans la classe qui lui correspond, le sous-automate stochastique caché et rendre hybride le processus global. Cependant, le contrôle de la variable continue est géré par le fichier source *PDMPController* et la classe du même nom. L'équation différentielle qui régit son évolution y est résolue, en utilisant les packages Numpy et Scipy.

La précision de la résolution de l'équation différentielle dépend de trois paramètres : `minTimeStep`, `maxTimeStep` et `collectingTimeStep`. `minTimeStep`, noté dt, correspond à l'intervalle de temps séparant deux instants pour lesquels la variable physique est calculée. `maxTimeStep` désigne l'intervalle de temps sur lequel est calculée l'évolution du niveau. Soit $T = \min(\texttt{maxTimeStep}, duration - t_0)$, où t_0 est l'instant initial et $duration$ est la durée maximale de l'histoire simulée.

`collectingTimeStep`, noté dt_{coll}, correspond aux instants pour laquelle l'évolution de la variable est stockée. Le vecteur $[\varphi(t_0), \varphi(t_0+dt_{coll}), \varphi(t_0+2dt_{coll}), ..., \varphi(t_0+T)]$ est collecté et mémorisé dans l'optique de calculer ensuite l'évolution moyenne du niveau ou d'autres indicateurs. Entre deux valeurs du vecteur stocké, l'évolution de la variable est obtenue par interpolation. Par exemple, si $t_0 + dt_{coll} = 3$ et $t_0 + 2dt_{coll} = 6$, $\varphi(5)$ est obtenu par interpolation à partir de $\varphi(3)$ et $\varphi(6)$.

Nous avons donc calculé $[\varphi(t_0), \varphi(t_0 + dt), \varphi(t_0 + 2dt), ..., \varphi(t_0 + T)]$ et mémorisé $[\varphi(t_0), \varphi(t_0 + dt_{coll}), \varphi(t_0 + 2dt_{coll}), ..., \varphi(t_0 + T)]$ avec $t_0 = 0$ et sous les conditions initiales.

Suivant le formalisme des PDMP, t_1 peut être fixé de trois façons différentes pour le calcul suivant.

1. Un composant saute d'un état à un autre lors d'une transition retardée stochastique, suivant une loi de probabilité continue. Soit T_1 l'instant de ce saut.
2. Dans le cas de l'approche d'un seuil par la variable déterministe, dt est diminué automatiquement afin de cibler plus précisément l'instant de franchissement T_2 de ce seuil. Une fois le seuil franchi, les conditions d'exécution de certaines transitions sont réunies.
3. Si aucun de ces événements n'arrive, alors $t_1 = t_0 + T$.

Dans l'un des deux premiers cas, notons $T_{next\_trip} = \min(T_1, T_2)$. L'arrivée dans certains états déclenche l'appel de la fonction qui met à jour les paramètres et résout l'équation différentielle entre t_1 et $t_1 + T$, avec $t_1 = T_{next\_trip}$. Aussi, chaque transition entre deux états, pour chaque composant, est suivie d'une mise à jour du PDMP qui vérifie s'il y a lieu de recalculer le vecteur $[\varphi(t_1), ..., \varphi(t_1 + T)]$.

3.3.3 Élaboration du modèle

Une base de connaissances est dédiée à la description d'une catégorie de système donnée. Un modèle regroupe toutes les informations spécifiques à la topologie d'un système en particulier. Dans le script décrivant la modélisation du système, chaque composant est déclaré en tant qu'instance d'une classe PyCATSHOO. Ces objets PyCATSHOO (PyO) sont ensuite reliés entre eux par des liens, qui représentent l'interaction entre deux composants et qui matérialisent les boîtes à messages.

Suite à la description du système, des observateurs dédiés sont créés pour suivre l'évolution des variables d'intérêt lors de la simulation. Les méthodes utiles à l'exploitation des résultats obtenus sont ensuite appelées par le script.

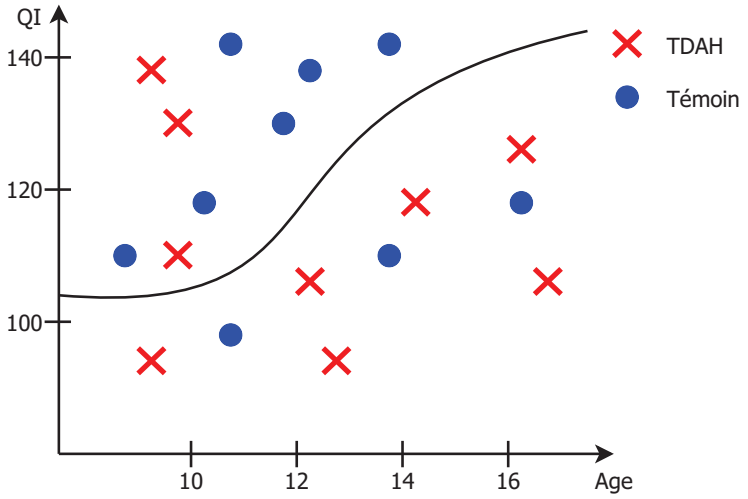

FIGURE 3.10 – SVM construit en vue du diagnostic du trouble du déficit de l'attention avec hyperactivité (TDAH), en fonction de l'âge et du QI de jeunes patients, inspiré par [Colby *et al.*, 2012].

Le chapitre 4.2 est consacré à la modélisation de deux systèmes hydrauliques, par la construction d'une BdC PyCATSHOO dédiée aux évacuateurs de crues. Les chapitres 5 et 6 détaillent l'exploitation des résultats obtenus suite aux simulations.

3.4 Machines à vecteurs support (SVM)

Les SVM (de l'anglais *Support Vector Machine*, aussi nommés « séparateurs à vaste marge » dans la littérature) sont une technique d'apprentissage automatique (*machine learning* en anglais) pour la classification binaire des données. Cette section est une brève introduction à une théorie très riche, décrite plus en détail dans les ouvrages [Hasan et Boris, 2006], [Cristianini et Shawe-Taylor, 2000], [Wang, 2005].

3.4.1 Problématique et notations

Les SVM concernent de nombreux domaines d'application, comme le traitement d'images, l'interprétation textuelle ou l'aide au diagnostic biologique. Ils sont mis en œuvre lorsque les données disponibles sont définies d'une part par des variables continues, d'autre part par une variable binaire, appelée **étiquette**. Par exemple, le diagnostic d'un nouveau cas d'une maladie repose naturellement sur la mesure de grandeurs caractéristiques du patient et sur la comparaison avec des cas déjà identifiés, ce qui est présenté par la figure 3.10.

Le but des SVM est de caractériser la frontière séparant les deux modalités de la variable binaire, à partir des données disponibles, afin de prédire l'étiquette d'un nouveau vecteur de données et donc de le classifier correctement.

Soit S l'ensemble des N données disponibles : $S = \{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq N}$. Pour tout i, $\mathbf{x}_i$ est le vecteur des variables continues et y_i est l'étiquette, avec $y_i \in \{-1, 1\}$.

Soient S^+ le sous-ensemble de S d'étiquette positive, et S^- son complémentaire : $S^+ = \{(\mathbf{x}, y) \in S / y = 1\}$ et $S^- = \{(\mathbf{x}, y) \in S / y = -1\}$.

3.4.1.1 Cas linéairement séparable

Dans cette section les données sont supposées être linéairement séparables, c'est-à-dire qu'il existe un hyperplan affine tel que tous les vecteurs de chaque classe se trouvent de part et d'autre de ce séparateur.

Définition 26. Le **séparateur linéaire** est défini par l'équation

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b \tag{3.10}$$

où $\mathbf{w} \cdot \mathbf{x}$ désigne le produit scalaire des vecteurs $\mathbf{w}$ et $\mathbf{x}$. La **frontière de séparation** ou **hyperplan séparateur** est définie par l'équation

$$f(x) = \mathbf{w} \cdot \mathbf{x} + b = 0. \tag{3.11}$$

Ce n'est pas la valeur de $f(\mathbf{x})$ qui est importante : $f(\mathbf{x})$ n'est pas nécessairement égal à 1 ou à -1, mais il doit être du même signe que y. Autrement dit, un point $(\mathbf{x}, y)$ est bien classé si $yf(\mathbf{x}) > 0$.

S est linéairement séparable si il existe $\mathbf{w}$ et b tels que $\begin{cases} \forall (\mathbf{x}, y) \in S^+ & f(\mathbf{x}) > 0 \\ \forall (\mathbf{x}, y) \in S^- & f(\mathbf{x}) < 0 \end{cases}$.

Il existe alors une infinité de séparateurs, ce qui est illustré par la figure 3.11. Comment déterminer le « meilleur » séparateur ?

La distance d'un point $\mathbf{x}$ à l'hyperplan est donnée par $\frac{|\mathbf{w} \cdot \mathbf{x} + b|}{\|\mathbf{w}\|}$. La distance du point le plus proche de l'hyperplan est appelée **marge**. L'hyperplan optimal est celui pour lequel la marge est maximale. Les coordonnées des vecteurs les plus éloignés de la frontière n'ont pas d'incidence sur ce problème d'optimisation ; seuls les vecteurs les plus proches des séparateurs sont pris en compte. Ces points sont nommés **vecteurs support** et donnent leur nom à la méthode.

Maximiser la marge revient à minimiser $\|\mathbf{w}\|$ sous certaines contraintes. Afin d'anticiper le calcul de dérivées, le problème d'optimisation s'écrit

$$\begin{cases} \min & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.c.} & \forall i,\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1. \end{cases} \tag{3.12}$$

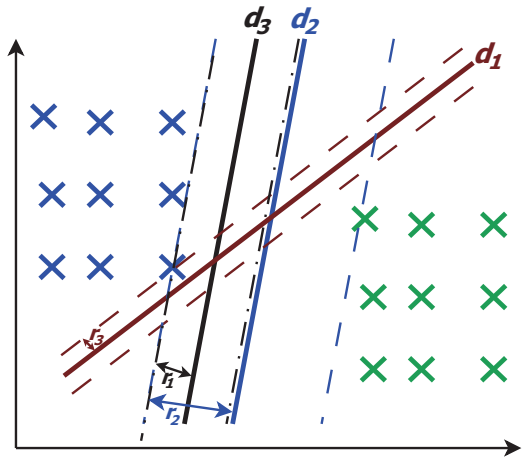

FIGURE 3.11 – La droite d_2 sépare « mieux » ces points que les droites d_1 et d_3, car c'est le séparateur qui a la plus grande marge r.

Cette optimisation sous contrainte devient un problème dual avec l'introduction de multiplicateurs de Lagrange. Le problème s'écrit alors

$$\begin{cases} \max & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{s.c.} & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \forall i,\ \alpha_i \geq 0. \end{cases} \tag{3.13}$$

Si les α_i^* sont solutions de ce problème alors $\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i$. Seuls les α_i^* correspondent aux vecteurs supports non nuls. La fonction de décision associée est $f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i \cdot \mathbf{x} + b$.

3.4.1.2 Cas non séparable

Le cas non linéairement séparable se résoud selon deux axes.

– Certains points peuvent avoir une marge négative ; ces « écarts » sont pénalisés. Le problème d'optimisation devient

$$\begin{cases} \min & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \\ \text{s.c.} & \forall i,\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0. \end{cases} \tag{3.14}$$

Le problème dual a la même forme que dans le cas séparable, seule la contrainte $\forall i,\ 0 \leq \alpha_i \leq C$ est différente.

– Les données qui ne sont pas linéairement séparables peuvent l'être dans un autre espace $\mathcal{F}$. Une solution pour simplifier la classification est de projeter les vecteurs dans cet autre espace puis d'y réaliser une séparation linéaire, ce qui est illustré par la figure 3.12. Cette projection est notée $\Phi : \mathbb{R}^N \to \mathcal{F}$ avec $card(\mathcal{F}) < N$.

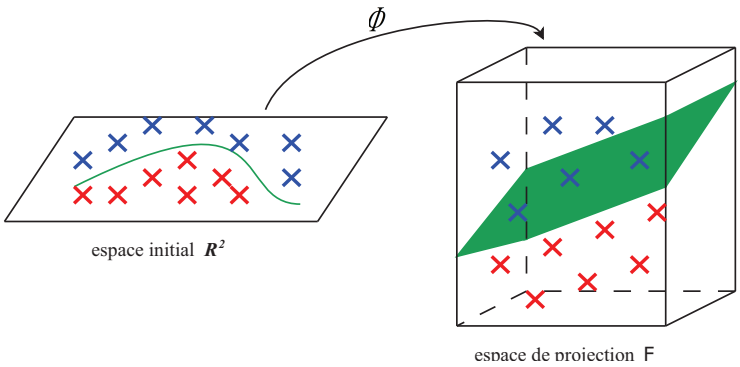

FIGURE 3.12 – Projection des données dans un espace où elles sont linéairement séparables.

Le problème d'optimisation dans $\mathcal{F}$ s'écrit

$$\begin{cases} \max & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ \text{s.c.} & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \forall i,\ 0 \leq \alpha_i \leq C. \end{cases} \tag{3.15}$$

La solution est de la forme

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^* y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b. \tag{3.16}$$

Le problème et sa solution dépendent de la fonction noyau k où k est le produit scalaire tel que $k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$. Les noyaux de référence sont les noyaux linéaires, polynomiaux, gaussiens et laplaciens.

Finalement, l'hyperplan séparateur est défini par

1. les α_i^* solution du problème d'optimisation. L'algorithme le plus couramment utilisé est l'algorithme SMO (*Sequential Minimal Optimization*) ;
2. le choix de b, déterminé grâce à un algorithme avec critère d'arrêt ;
3. le choix du bon noyau k.

3.4.2 La librairie libsvm

La librairie libsvm [Chang et Lin, 2011] est une bibliothèque d'algorithmes de SVM pour la classification mais également pour la régression. Cette bibliothèque *open source* est reconnue par la communauté scientifique et propose des librairies compatibles avec les langages de programmation les plus courants.

Deuxième partie

Prise en compte de l'information temporelle de la modélisation à la synthèse d'indicateurs fiabilistes

Chapitre 4

Description et modélisation des évacuateurs de crues

Afin d'évaluer et d'étudier la sûreté de fonctionnement des Systèmes Dynamiques Hybrides (SDH), nous proposons d'associer la description par les Automates Stochastiques Hybrides (ASH) distribués à la simulation de Monte Carlo. La seconde partie, à travers ses trois chapitres, apporte la preuve de la faisabilité de cette démarche. Si les évacuateurs de crues (EdC) illustrent cette démonstration, la méthodologie proposée n'est pas restreinte à ce type de système. En effet, les EdC, par leur dynamique et par leurs propriétés, sont représentatifs des SDH en général. Par exemple, il est possible de modéliser et de quantifier un système issu de l'industrie nucléaire et relevant de la fiabilité dynamique, grâce à la méthodologie proposée.

Ce chapitre présente la classe de systèmes étudiés et la modélisation qui en découle. Les évacuateurs de crues (EdC) sont décrits dans la section 4.1. Leur fonctionnement dépend du temps et la modélisation doit prendre en compte cette dynamique. Les EdC sont des systèmes dynamiques hybrides complexes. Cette complexité est due à

- la dimension des EdC. Chaque EdC est constitué d'environ 75 composants ;
- les interactions entre les composants, mais aussi entre les composants et l'environnement. L'ouverture des vannes est en effet commandée après l'atteinte d'un seuil par le niveau de la retenue ;
- l'évaluation simultanée de l'état du système et du niveau d'eau dans la retenue, par résolution de l'équation différentielle qui la gouverne, en fonction de l'état des composants.

La modélisation par Automates Stochastiques Hybrides distribués est décrite dans la section 4.2. En utilisant les attributs des classes PyCATSHOO, il est possible d'appréhender les systèmes dynamiques hybrides dans toute leur complexité. La méthodologie proposée est d'abord exposée sur un réservoir simple. Cet exemple illustratif est progressivement étoffé. Finalement, chaque composant est modélisé avec réalisme et le fonctionnement du système étudié est représentatif de celui des EdC.

4.1 Fonctionnement des évacuateurs de crues

Cette section présente le système industriel qui sera l'objet de notre étude. Ces travaux s'inscrivent dans un projet qui, à long terme, a l'ambition de modéliser non seulement un système, mais une catégorie de systèmes : celle des évacuateurs de crues vannés. La compréhension du fonctionnement des évacuateurs est une démarche qui a été menée parallèlement à celle de l'assimilation de la méthode GASPART. Aussi, la description des évacuateurs est parfois intimement mêlée à celles d'hypothèses de modélisation utilisées par GASPART.

Cette section débute par une introduction sur la prise en compte du temps dans le déroulement d'une crue. Après une présentation des caractéristiques d'une crue, la section 4.1.3 expose le fonctionnement des deux évacuateurs, en détaillant la marche des sous-systèmes hydromécaniques, d'alimentation électrique et de contrôle-commande. Ce niveau de détail a un double objectif : d'une part, saisir les différences entre les deux barrages ; d'autre part, introduire les verrous que la modélisation devra lever. Ensuite, la section 4.1.4 résume le rôle de l'opérateur pendant la crue. Finalement, une synthèse sur l'élaboration des données de fiabilité précède un récapitulatif des hypothèses de modélisation utilisées.

4.1.1 Prise en compte du temps dans le déroulement d'une crue

L'objectif est la modélisation du processus d'évacuation d'une crue. L'intervalle de temps concerné par ce processus commence à l'instant où les prémices d'une crue deviennent théoriquement perceptibles et se prolonge jusque la fin de cette crue. Cet intervalle de temps est divisé en deux phases : la phase de veille et la phase de crue, ce qui est illustré sur la figure 4.1.

L'instant $t_0 = 0$ correspond à l'apparition des prémices d'une crue susceptibles d'être détectées par le système de prévision centralisé, s'il y en a un, que cette détection soit un succès ou non. En cas de non détection immédiate de la crue, l'instant t_0 ne correspond donc pas au début de la phase de veille effective.

La phase de veille effective débute lorsque les prémices d'une crue deviennent perceptibles (l'alerte est donnée par un moyen d'alerte centralisé ou par un automate barrage). Une fois cette alerte reçue par un ou plusieurs opérateurs, ces derniers se rendent sur site s'ils n'y sont pas déjà.

Une fois que les paramètres de l'aménagement (débit entrant, niveau du plan d'eau, débits déversés, etc.) atteignent certains seuils, le système entre en phase de crue. La phase de crue s'étale de l'établissement de la crue à sa fin.

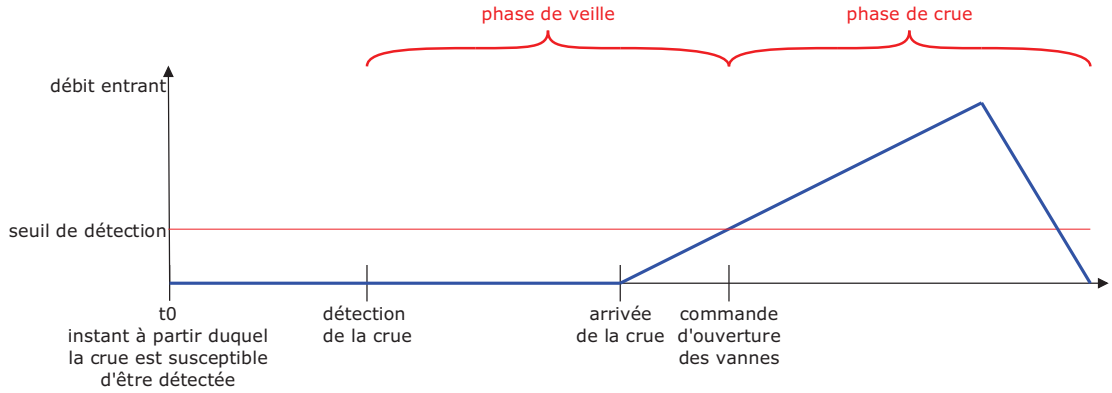

FIGURE 4.1 – Chronologie d'une crue

4.1.2 Caractérisation d'une crue

4.1.2.1 Fréquence d'une crue

Les crues considérées sont des crues décennales (10^{-1}/ an), des crues centennales (10^{-2}/ an) et des crues millenales (10^{-3}/ an).

4.1.2.2 Forme et débit d'une crue

Dans le cas le plus simple, une crue est seulement caractérisée par le débit maximal déversé depuis l'amont du barrage. Cela implique la pose d'une hypothèse pessimiste : ce débit maximal est considéré comme le débit entrant pendant toute la durée de la crue.

Dans le cas le plus général, une crue est définie par un hydrogramme, qui caractérise heure par heure le débit entrant en $m^3 \cdot s^{-1}$.

4.1.2.3 Durée de la crue et délais de détection et d'établissement

Chaque type de crue est caractérisé par une durée, qui est une donnée d'entrée du modèle. Les crues les plus fortes (crues millenales) sont aussi les plus brusques, dans le sens où elles s'établissent vite et s'écoulent vite. Une crue dure entre 16 et 90 heures. La durée moyenne d'une crue est d'environ 48 heures [Chraibi, 2013b]. Une analyse est réalisée pour un type de crue, caractérisé par une durée et un hydrogramme donnés.

Dans le cas d'une défaillance du dispositif d'alerte centralisé, la détection de la crue est effectuée par le dispositif de contrôle-commande local ou par l'opérateur. Ce délai de détection est une autre caractéristique de la crue.

Enfin, le délai d'établissement de la crue correspond au temps écoulé entre l'instant t_0 et le début de la phase de crue. Ce délai varie de 12 à 40 heures. La période

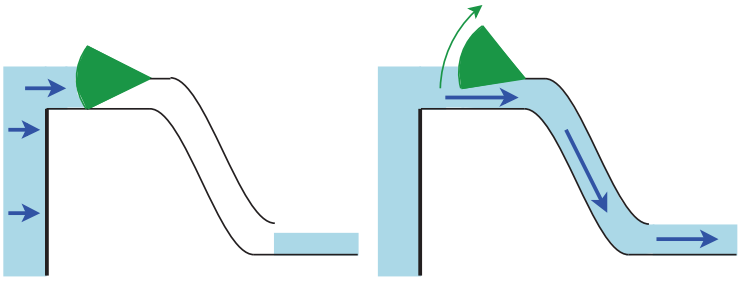

FIGURE 4.2 – Représentation schématique d'une vanne de surface

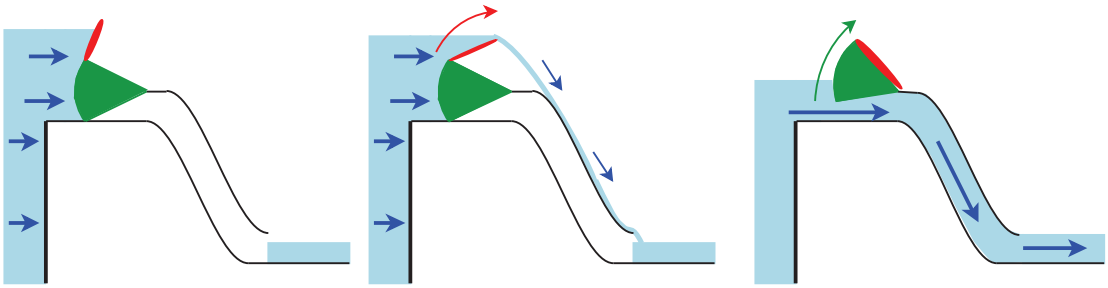

FIGURE 4.3 – Représentation schématique d'une vanne de surface associée à un clapet

étudiée pour le processus d'évacuation d'une crue dure donc entre 28 et 130 heures [Chraibi, 2013b].

4.1.3 Fonctionnement de deux évacuateurs de crues

Cette sous-section décrit le fonctionnement de deux installations réelles dont la structure est confidentielle. Des représentations schématiques des sous-systèmes hydro-mécaniques, d'alimentation électrique et de contrôle-commande sont proposées pour les deux EdC. Cependant ces représentations ne peuvent être publiées que dans une version confidentielle, de diffusion restreinte, de ce manuscrit.

La figure 4.2 représente schématiquement le fonctionnement d'une vanne de surface. Une vanne de demi-fond fonctionne selon le même principe, mais plus bas par rapport à la structure du réservoir.

La figure 4.3 représente schématiquement le fonctionnement d'une vanne de surface si celle-ci est surmontée d'un clapet. Dans un premier temps, le clapet s'efface, puis la vanne est manoeuvrée.

4.1.4 Rôle de l'opérateur

Durant la phase de veille, les opérateurs préparent l'ouverture des évacuateurs par des opérations de vérification du matériel. Lors de la phase de crue, les opérateurs sont

responsables des manœuvres d'ouverture des vannes. Pendant ces deux périodes, ils assurent aussi les réparations des composants défaillants si nécessaire et si possible.

L'opérateur n'est pas forcément sur le site au début de la phase de veille. Dans ce cas, l'opérateur d'astreinte arrive sur le barrage après un délai de déplacement. Certains aléas peuvent survenir, comme une non-réponse de l'opérateur d'astreinte à l'alarme ou des difficultés de déplacement. Une fois sur place, l'opérateur est susceptible d'échouer dans certaines manœuvres de vérification et de réparation.

4.1.5 Données de fiabilité

L'élaboration des données de fiabilité requiert trois étapes [Chraibi, 2013b]. Dans un premier temps, les données de fiabilité moyennes sont issues de l'analyse du retour d'expérience sur l'ensemble des évacuateurs de crues vannés du parc EDF. Dans un deuxième temps, ces données moyennes sont ajustées pour chaque site afin de les adapter aux spécificités de l'évacuateur étudié. Une troisième étape consiste à estimer les données de fiabilité des actions des opérateurs.

L'évaluation des données de fiabilité humaine n'a pas fait l'objet d'une méthodologie spécifique. La différence entre un opérateur parfait ou fiable et un opérateur faillible repose donc sur des probabilités estimées de façon purement forfaitaire, dans l'objectif de montrer la sensibilité à l'efficacité des actions des opérateurs. Si l'on considère un opérateur parfait, toutes ces probabilités sont égales à 0, et dans ce cas seule la fiabilité intrinsèque du système d'évacuation matériel est évaluée.

Enfin, si les défaillances de cause commune (DCC) fonctionnelles sont bien prises en compte, ce n'est pas le cas des autres DCC. Les DCC fonctionnelles (par exemple, défaut de l'alimentation électrique) sont distinguées des défaillances intrinsèques au système et sont comptabilisées dans l'élaboration des données de fiabilité. En revanche, les DCC liées à une cause externe (par exemple un orage) ou un défaut de conception ou de maintenance sur plusieurs composants identiques, sont agrégées aux défaillances intrinsèques au système.

Un composant est considéré comme réparable si la durée de sa réparation est courte relativement à la durée de la crue. Dans le cas contraire, le composant est considéré comme non réparable. Chaque type de panne est associée à un taux de défaillance. En fonction de la réparabilité des composants, les données de fiabilité disponibles sont le taux de défaillance non réparable, le taux de défaillance réparable et la durée de réparation moyenne. Les composants actionnés sont aussi caractérisés par une probabilité de défaillance à la sollicitation, réparable et/ou non réparable.

4.1.6 Hypothèses de modélisation de la méthode GASPART et des travaux de thèse

Le travail de modélisation proposé repose sur plusieurs hypothèses. La plupart ont été formulées dans l'élaboration de la méthode GASPART, d'autres l'ont été dans le cadre de ces travaux de thèse.

1. Les capteurs ne donnent pas d'informations contradictoires ni de fausses indications. Un capteur défaillant ne donne aucune information ; aucune prise de décision n'est possible s'il n'y a pas d'autres capteurs ou mesure visuelle.
2. La pression et le niveau d'eau dans la retenue peuvent jouer sensiblement sur la puissance requise pour actionner les vannes, et par conséquent sur le nombre de transmissions nécessaires à l'ouverture des vannes. La prise en compte de cette dépendance complète les propriétés des EdC en tant que représentants des SDH. Cette sensibilité n'est pas étudiée ici mais pourrait l'être lors de l'enrichissement de la base de connaissances.
3. Certains composants ont été agrégés et ne sont représentés que par un seul composant, de probabilité de défaillance équivalente. Cela concerne essentiellement le système d'alimentation électrique de l'un des deux évacuateurs étudiés.
4. Le premier objectif d'un évacuateur de crues est d'évacuer un volume d'eau suffisant grâce à ses vannes pour éviter qu'un niveau critique ne soit submergé. Un deuxième objectif, non étudié ici, est de laminer cette même crue. Ainsi, le débit en aval du barrage ne doit pas être supérieur au débit en amont. Dans la réalité, cela se traduit par une ouverture partielle des vannes, voire la fermeture éventuelle de certaines vannes. Afin de simplifier la modélisation, les vannes sont considérées totalement ouvertes. La réduction du débit sortant se retrouve dans le calcul, en empêchant le débit sortant d'être supérieur au débit entrant (la différence entre débit entrant et débit sortant est minorée par un débit nul). L'optimisation du calcul de commande d'ouverture et la considération du processus de fermeture des vannes pourraient également être l'objet d'un enrichissement de la base de connaissances.
5. Avant l'arrivée de la crue, un abaissement du plan d'eau est parfois envisagé par les opérateurs, dans le but d'évacuer la crue plus progressivement. Cette possibilité ne sera pas modélisée.
6. Certains évacuateurs sont équipés de vannes double-corps : le haut et le bas de la vanne doivent s'accrocher avant le début de l'ouverture de la vanne. Cette manœuvre peut échouer, empêchant ainsi l'ouverture de la vanne. Cela ne sera pas modélisé ici car cela ne concerne pas les deux EdC étudiés.

Les hypothèses 7 et 8 ont été ajoutées à celles de la méthode GASPART, pour la réalisation des travaux de thèse. Ces hypothèses concernent la période précédant l'arrivée de la crue. Elles pourraient être levées grâce à la modélisation d'un quatrième sous-système représentant le dispositif d'alarme.

7. Les manœuvres effectuées pendant la phase de veille ne seront pas modélisées. Il s'agit de vérifications, voire de réparations, dédiées à diminuer les risques de défaillances au moment le plus intense de la crue.
8. Le système d'alarme (télécommunications) ne sera pas étudié. L'automate qui détecte la crue est considéré comme capable d'alarmer directement l'opérateur d'astreinte. La modélisation est ainsi simplifiée, notamment en ce qui concerne des retards possibles dans l'avertissement de l'opérateur.

La levée de chacune de ces hypothèses est théoriquement possible en utilisant le formalisme des ASH, sans explosion du nombre d'automates. Les hypothèses 4 et 5 concernent les paramètres de l'équation différentielle qui régit l'évolution du niveau dans la retenue. Leur prise en compte grâce à un algorithme d'optimisation représenterait plus finement le comportement anticipatif des opérateurs avant l'arrivée de la crue. De même, les hypothèses 7 et 8 sont relatives à la période précédant l'arrivée de la crue. La modélisation de cette phase de veille demanderait un effort de modélisation du dispositif d'alarme en définissant les automates associés, et en distinguant les paramètres de fiabilité en fonction de la phase étudiée. Enfin les hypothèses 1, 3 et 6 sont des hypothèses relatives aux composants ; l'enrichissement des automates correspondant par quelques états permettrait de les lever. Finalement, l'hypothèse 2 demanderait éventuellement la définition d'une nouvelle variable continue, ou de nouvelles boîtes à messages vers les actionneurs.

4.2 Modélisation des évacuateurs de crues

Cette section détaille le principe de modélisation des évacuateurs de crues. Cette modélisation consiste à construire une base de connaissances dédiée aux évacuateurs de crues. Cette construction repose sur plusieurs étapes :

- Décomposer les systèmes, composant par composant. Rassembler les composants similaires par classes.
- Pour chaque classe de composant, identifier les états qui décrivent les différentes phases de son fonctionnement.
- Caractériser les transitions déterministes entre ces états par un ensemble de conditions.
- Étudier les interactions entre composants. Un composant c_1 interagit sur un autre composant c_2 si un changement d'état de c_1 provoque le changement d'un attribut de c_2. Ces liens sont modélisés par des boîtes à messages.
- Doter une des classes de composants du dispositif de calcul de la variable continue.

La dernière partie du travail de modélisation est la description précise de chaque évacuateur. Cette description liste chaque composant du système comme une instance de la classe qui lui correspond. Les liens entre composants sont également énumérés. Les données physiques et les données de fiabilité y sont renseignées.

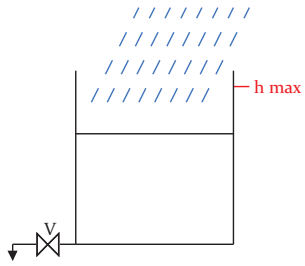

FIGURE 4.4 – Illustration du cas-test simple

La base de connaissances PyCATSHOO « évacuateurs de crues » a débuté par la modélisation d'un réservoir simple, puis les composants des sous-systèmes décrit dans la section 4.1.3 ont été ajoutés progressivement. Il s'est avéré que la plupart des composants ont un fonctionnement similaire, à quelques nuances près. Ce constat a mené à une refonte de la base de connaissances, dont la nouvelle version est plus générique.

La modélisation d'un cas-test simple est décrite précisément dans la section 4.2.1. Les classes principales de la base de connaissances finale sont ensuite présentées dans la section 4.2.2.

4.2.1 Modélisation d'un cas-test simple

La construction d'une base de connaissances est développée dans cette section. Pour cela, l'évacuateur de crues est réduit à sa plus simple expression. Il s'agit d'un réservoir à ciel ouvert, muni d'une ou plusieurs vannes de vidange, comme celui représenté schématiquement sur la figure 4.4. L'évolution du niveau du réservoir est soumise à l'arrivée d'une crue en amont de celui-ci.

Le temps est pris en compte conformément à la section 4.1.1. L'intervalle de temps considéré démarre à l'instant où les prémices d'une crue sont susceptibles d'être détectées. L'ouverture des vannes est commandée quand la crue est établie. La prise en compte du temps s'arrête à la fin de la crue. Ainsi, le déroulement du temps est constitué de deux phases. La première phase, qui correspond à la phase de veille, est caractérisée par des débits entrant et sortant nuls. La deuxième phase est la phase de crue.

Lorsque l'ouverture de la vanne est commandée, celle-ci peut refuser de s'ouvrir. Il s'agit d'une défaillance à la sollicitation. Au cours de son ouverture, la vanne est susceptible de se bloquer : c'est une défaillance en fonctionnement.

4.2.1.1 Évolution du niveau dans le réservoir

La dynamique du niveau dans le réservoir dépend des débits entrant et sortant dans celui-ci.

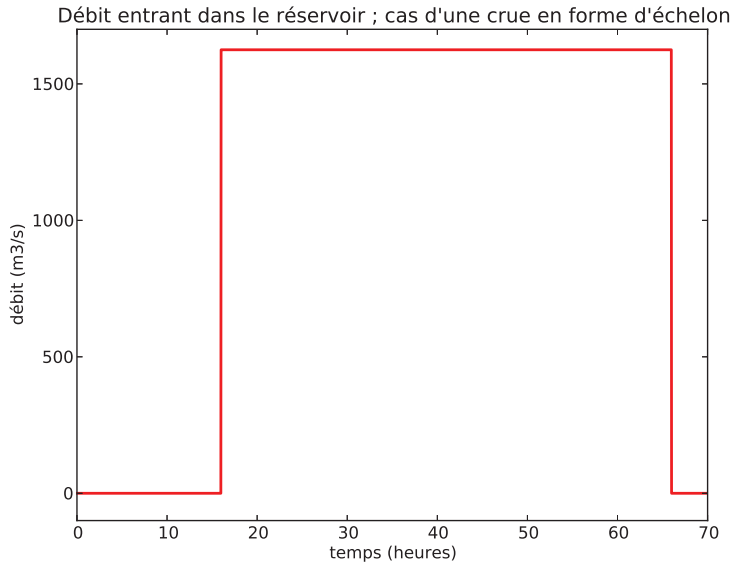

FIGURE 4.5 – Crue en forme d'échelon : évolution du débit entrant dans le réservoir en fonction du temps

Les données disponibles pour l'évacuateur de M. permettent de modéliser l'hydrogramme de crue (débit entrant) et la débitance des vannes (débit sortant) en accord avec la réalité. En revanche, ces données n'étaient pas accessibles pour l'un des évacuateurs au moment de l'analyse de cette installation. La méthode GASPART propose dans ce cas une modélisation de la crue sous la forme d'un échelon : l'hydrogramme est remplacé par la valeur du débit maximal de la crue, pendant toute sa durée. De même la débitance est constante, de valeur égale à la débitance correspondant au niveau de sûreté.

Dans un premier temps, nous utiliserons ces hypothèses simplificatrices pour modéliser l'évolution du niveau dans la retenue. La comparaison possible avec des calculs analytiques en est le principal bénéfice. L'introduction d'un hydrogramme réaliste et d'une débitance dépendant du niveau font partie des étapes suivantes.

4.2.1.1.1 Évolution simple

4.2.1.1.1.1 Débit entrant

Le débit entrant correspondant à une crue en forme d'échelon s'écrit

$$q_{ent}(t) = I_c \times \mathbf{1}_{[t_{c_0}; t_{c_f}]}(t) \tag{4.1}$$

où I_c est une constante représentant l'intensité de la crue, $\mathbf{1}$ la fonction indicatrice et t_{c_0} et t_{c_f} les instants respectifs de début et de fin de la crue. Le débit entrant est illustré par la figure 4.5.

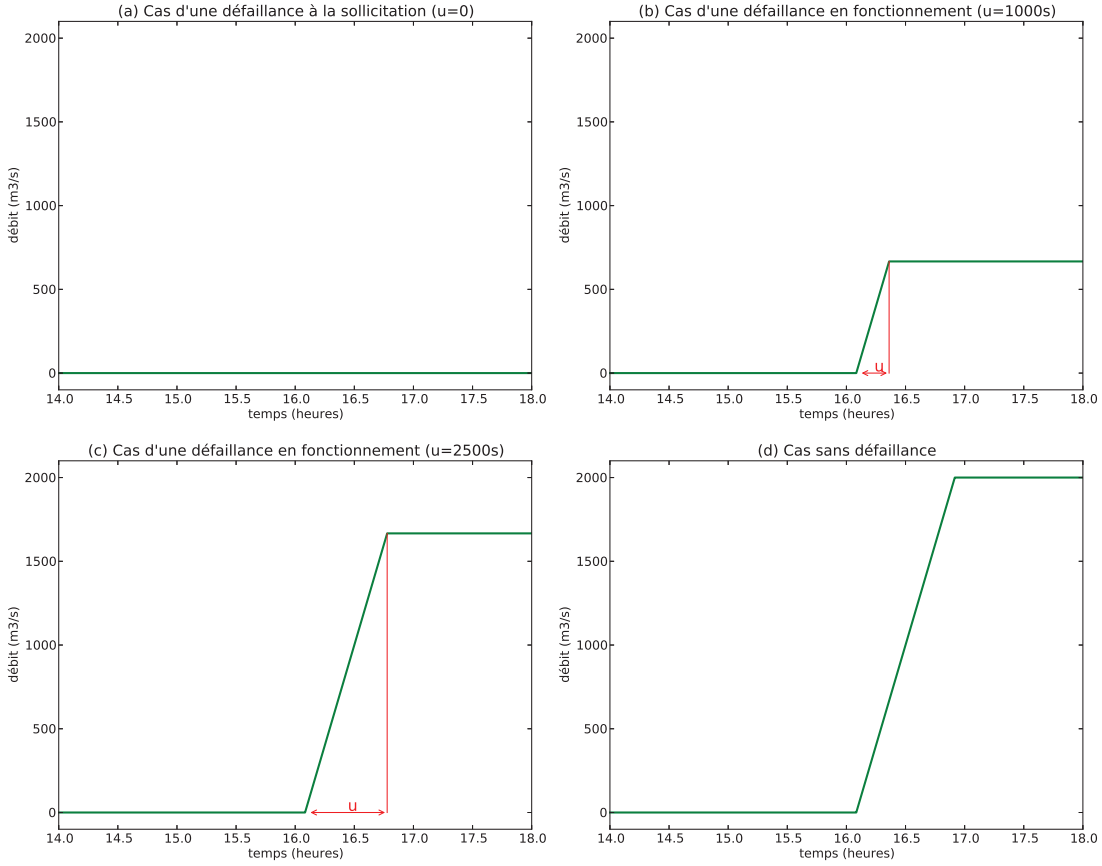

FIGURE 4.6 – Débitance constante : évolution du débit sortant du réservoir en fonction du temps, en fonction de l'instant de panne u

4.2.1.1.1.2 Débit sortant

Soient t_{v_0} l'instant du début de l'ouverture de la vanne v et $d_{ouv}^{(v)}$ la durée du processus d'ouverture.

La débitance de la vanne v est nulle avant le début de l'ouverture de la vanne, constante après l'atteinte de l'ouverture maximale, et linéaire entre ces deux instants :

$$q_{sor}^{(v)}(t) = \begin{cases} 0 & \text{si } t \leq t_{v_0} \\ \frac{(t-t_{v_0}) \times q_{max}^{(v)}}{d_{ouv}^{(v)}} & \text{si } t_{v_0} \leq t \leq t_{v_0} + \min\left(u,\, d_{ouv}^{(v)}\right) \\ \min\left(q_u^{(v)},\, q_{max}^{(v)}\right) & \text{sinon} \end{cases} \tag{4.2}$$

où $q_{max}^{(v)}$ est le débit sortant maximal théorique de la vanne v, correspondant à son ouverture entière. Si la vanne tombe en panne au bout d'une durée u lors de son ouverture ($u \leq d_{ouv}^{(v)}$), alors $q_u^{(v)} = q_{sor}^{(v)}(t_{v_0} + u)$ est le débit sortant associé à cet instant.

Cela traduit la distinction entre deux cas :

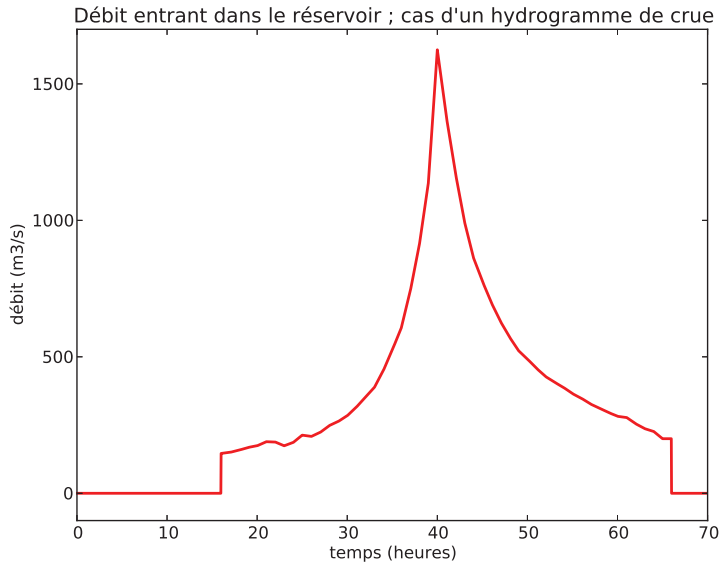

FIGURE 4.7 – Hydrogramme de crue disponible : évolution du débit entrant du réservoir en fonction du temps

- Dans le premier cas sans panne, le débit sortant associé à la vanne v augmente de manière linéaire jusqu'à son débit maximal théorique $q_{max}^{(v)}$, pendant la durée $d_{ouv}^{(v)}$. Après la fin de l'ouverture à l'instant $t_{v_0} + d_{ouv}^{(v)}$, la débitance est constante et sa valeur est $q_{max}^{(v)}$.
- Dans le deuxième cas avec panne au bout d'une durée u, le débit sortant augmente de manière linéaire jusqu'au débit $q_u^{(v)}$ correspondant à l'instant $t_{v_0} + u$. Après la défaillance à l'instant $t_{v_0} + u$, la débitance est constante et sa valeur est $q_u^{(v)}$.

La figure 4.6 présente l'évolution de la débitance d'une vanne pour quatre situations différentes. Ces situations sont la panne à sollicitation, la défaillance en fonctionnement au bout de $u = 1000$ secondes, la défaillance en fonctionnement au bout de $u = 2500$ secondes et l'ouverture totale de la vanne sans défaillance.

4.2.1.1.1.3 Évolution du niveau

L'évolution du niveau dans le réservoir en fonction du temps est donnée par l'équation différentielle

$$\frac{dh}{dt}(t) = q_{ent}(t) - \sum_v q_{sor}^{(v)}(t). \tag{4.3}$$

4.2.1.1.2 Évolution réaliste

4.2.1.1.2.1 Débit entrant

La crue est caractérisée par le débit amont entrant dans le réservoir. Ce débit est

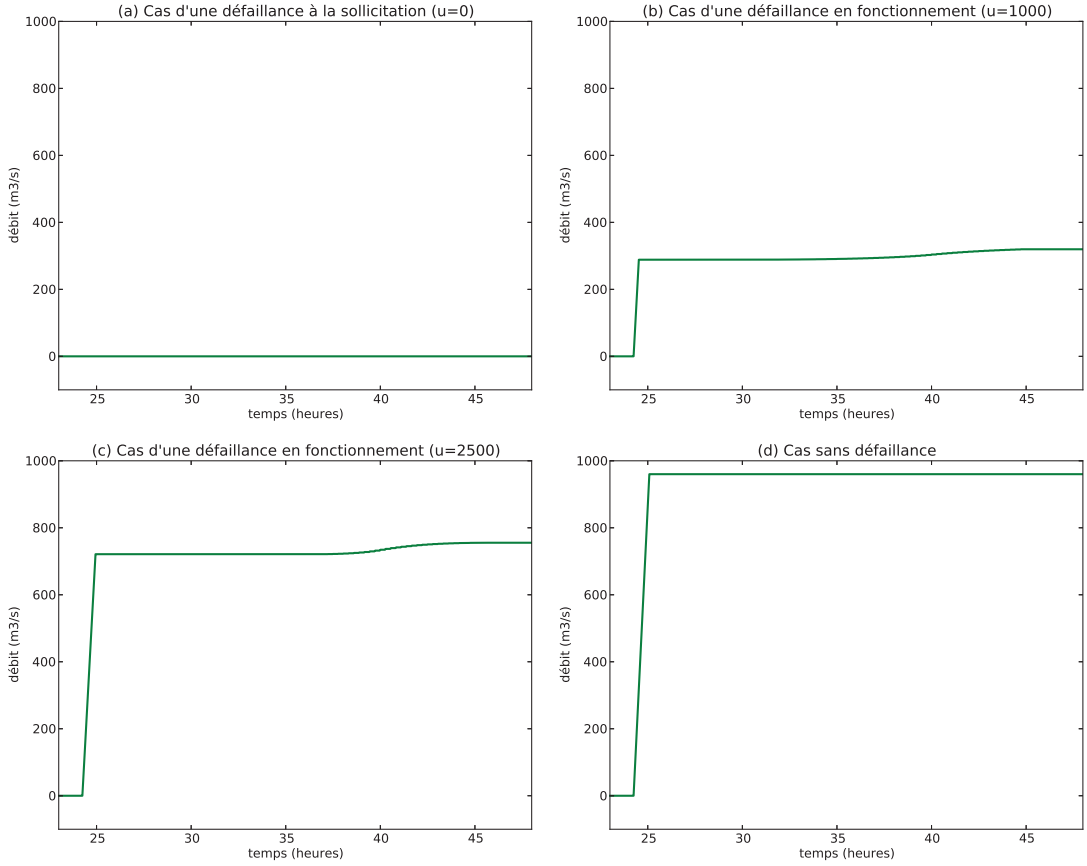

FIGURE 4.8 – Débitance réaliste : évolution du débit sortant du réservoir en fonction du temps

disponible à intervalles de temps réguliers. Ces données forment un nuage de points appelé hydrogramme. La figure 4.7 est représentative de l'hydrogramme d'une crue millenale.

Entre deux points $H[T]$ et $H[T+dt]$ de l'hydrogramme, le débit entrant à l'instant $t \in [T,\, T+dt[$ dans le réservoir est estimé grâce à l'interpolation linéaire

$$q_{ent}(t) = H[T] + (H[T+dt] - H[T]) \times \frac{t-T}{dt}. \tag{4.4}$$

4.2.1.1.2.2 Débit sortant

La débitance d'une vanne de surface s'exprime sous la forme

$$q_{sor}^{(v)}(t) = mL \times ouv(t) \times \sqrt{2g \times \max(0,\, h(t) - sb)} \tag{4.5}$$

où m est un coefficient propre à chaque aménagement hydraulique, L est la largeur

de la vanne et sb est la hauteur du seuil bas de la vanne. La constante $g = 9,81\ \mathrm{m} \cdot \mathrm{s}^{-2}$ est l'accélération de la pesanteur terrestre [Chraibi, 2013b].

$h(t)$ est le niveau de la retenue à l'instant t. Si ce niveau est inférieur au seuil bas de la vanne, alors celle-ci ne débite pas.

Enfin, $ouv(t)$ représente la hauteur de l'ouverture de la vanne à l'instant t. La fonction ouv s'écrit

$$ouv(t) = \begin{cases} 0 & \text{si } t < t_{v_0} \\ (t - t_{v_0}) \frac{H}{d_{ouv}^{(v)}} & \text{si } t \in \left[t_{v_0},\, t_{v_0} + d_{ouv}^{(v)}\right[\text{ et } u > d_{ouv}^{(v)} \\ H & \text{si } t \geq t_{v_0} + d_{ouv}^{(v)} \text{ et } u > d_{ouv}^{(v)} \\ \frac{H}{d_{ouv}^{(v)}} u & \text{si } t \geq t_{v_0} + d_{ouv}^{(v)} \text{ et } u \leq d_{ouv}^{(v)} \end{cases} \tag{4.6}$$

où t_{v_0} est l'instant du début de l'ouverture de la vanne, $d_{ouv}^{(v)}$ est la durée du processus d'ouverture, H est la hauteur d'ouverture totale et u la date d'une éventuelle défaillance de la vanne.

La hauteur du seuil bas du clapet est également un paramètre dans le calcul de la débitance de celui-ci. Or le seuil bas d'un clapet est posé au dessus de la vanne qui lui est associée. Aussi la variable sb_{clapet} dépend-elle de la hauteur de l'ouverture de la vanne associée au clapet à l'instant t [Chraibi, 2013b]. Finalement, la débitance d'un clapet est donnée par

$$q_{sor}^{(c)}(t) = mL \left(\max\left(0,\, y(t) - sb_{clapet}(t) - ouv(t)\right)\right)^{3/2}. \tag{4.7}$$

La figure 4.8 présente l'évolution de la débitance d'une vanne pour les quatre situations évoquées ci-dessus.

4.2.1.1.2.3 Évolution du niveau

L'évolution du niveau dans le réservoir en fonction du temps est théoriquement donnée par l'équation différentielle

$$\frac{dh}{dt}(t) = q_{ent}(t) - \sum_v q_{sor}^{(v)}(t).$$

Or le rôle d'un évacuateur de crues est de laminer la crue. Aussi, le débit sortant ne doit pas être supérieur au débit entrant. C'est pourquoi la forme de l'équation différentielle est en réalité

$$\frac{dh}{dt}(t) = \max\left(0,\ q_{ent}(t) - \sum_v q_{sor}^{(v)}(t)\right). \tag{4.8}$$

Il est toujours possible d'estimer l'évolution du niveau dans la retenue grâce à la simulation de Monte Carlo et la résolution numérique de cette équation différentielle. Cependant, la complexité de l'équation différentielle ne permettra pas de la résoudre analytiquement et de comparer les résultats. C'est pourquoi une approche par modélisation puis simulation de Monte Carlo est privilégiée.

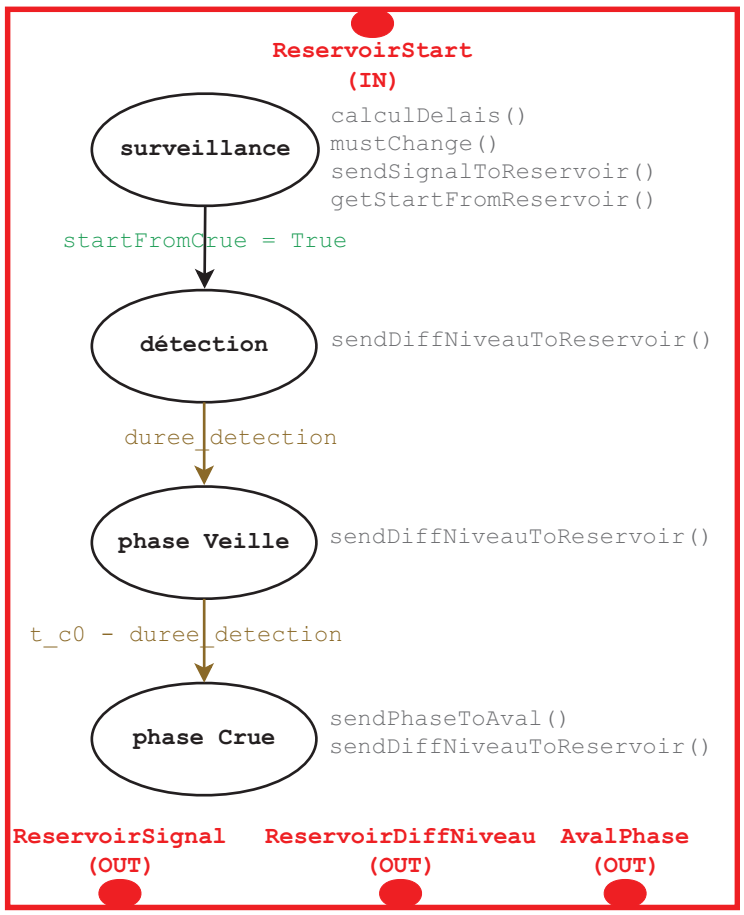

FIGURE 4.9 – Automate de la crue

4.2.1.2 Modélisation par les Automates Stochastiques Hybrides

Comme détaillé dans la section 3.2, un automate stochastique est défini par un ensemble d'états et de transitions. Chaque transition est définie par un ensemble de conditions, et éventuellement par une loi et un délai. Si cet automate est également muni de la modélisation d'une variable déterministe continue et de la description des interactions de cette variable avec les événements stochastiques, alors l'automate est Automate Stochastique Hybride (ASH).

PyCATSHOO est basé sur la programmation orientée objet (POO). La POO définit des briques logicielles appelées objets et permet l'interaction entre elles. Un objet définit une structure de données rassemblées dans une classe. Chaque classe est caractérisée par des attributs et des méthodes. La POO permet de définir des héritages d'une classe vers une autre et des échanges de messages entre objets.

La structure de PyCATSHOO est évoquée dans la section 3.3. Les attributs d'une classe PyCATSHOO sont une liste d'états et une liste de transitions. Ces états et ces

transitions sont ceux des ASH correspondant. L'arrivée dans un état est susceptible de provoquer l'exécution de méthodes associées à cet état. La liste de boîtes à messages entrant et sortant est également un attribut d'une classe PyCASHOO.

Notre système simple est caractérisé par le comportement de trois objets : une crue, une vanne et un réservoir. Ces trois objets interagissent entre eux. Le modèle global que nous proposons est illustré par la figure 4.12, qui représente la structure de la base de connaissances associée. Les sections 4.2.1.2.1, 4.2.1.2.2 et 4.2.1.2.3 décrivent la modélisation de ces trois objets. Cette modélisation a été également présentée dans [Broy *et al.*, 2013].

4.2.1.2.1 Modélisation de la crue

L'automate de la crue et la classe PyCATSHOO associée sont représentés sur la figure 4.9.

4.2.1.2.1.1 États

Le déroulement d'une crue se décompose en deux phases : une phase de veille et une phase de crue, d'où la création des deux états **`phase Veille`** et **`phase Crue`**. La phase de veille ne démarre pas toujours à l'instant initial, il est donc nécessaire de créer un état **`surveillance`**.

Le calcul de la dynamique du niveau d'eau est enclenché dès l'initialisation d'une histoire. Ce calcul débute correctement s'il est guidé par des méthodes qui assurent l'échange d'informations entre la crue et le réservoir. Les méthodes sont appelées à l'occasion d'une transition, aussi l'état **`détection`** est-il créé en vue d'utiliser la transition **`surveillance`**→**`détection`**.

4.2.1.2.1.2 Transitions et méthodes

L'activation de l'état **`surveillance`** appelle la méthode `calculDelais()` qui est chargée de déterminer les dates de début de phase de veille et de phase de crue.

La transition **`surveillance`**→**`détection`** est conditionnée par l'égalité `mustChange()=True`. Cette méthode `mustChange()` envoie un signal au réservoir via la méthode `sendSignalToReservoir()`. Le réservoir accuse réception via la méthode `getStartFromReservoir()`. A la réception de ce « start », la méthode `mustChange()` renvoie `True` et la transition **`surveillance`**→**`détection`** est possible.

Les transitions **`détection`**→**`phase Veille`** et **`phase Veille`**→**`phase Crue`** sont des transitions déterministes stochatiques. L'arrivée dans les états **`après`**, **`phase Veille`** et **`phase Crue`** appelle la méthode `sendDiffNiveauToReservoir()`. Cette méthode envoie des informations au réservoir sur les caractéristiques de la crue susceptibles d'influer sur le niveau d'eau.

Enfin, l'arrivée dans l'état **`phase Crue`** appelle la méthode `sendPhaseToAval()`. Cette méthode informe d'éventuels composants, appelés « aval », de l'arrivée effective

de la crue. Ces composants peuvent être l'ABM, l'opérateur ou des capteurs. Dans ce système simple, c'est la vanne qui sera directement informée de l'arrivée de la crue.

4.2.1.2.1.3 Boîtes à messages

La classe Crue dispose de quatre boîtes à messages.

1. « ReservoirSignal » est une boîte à messages `OUT` chargée d'envoyer le signal sortant du début de l'histoire au réservoir via la méthode `sendSignalToReservoir()`. Le réservoir réceptionne l'information entrante via sa boîte à messages `IN` « CrueSignal » et la méthode `getSignalFromCrue()`.
2. « ReservoirStart » est une boîte à messages `IN` chargée de réceptionner le signal « Start » via la méthode `getStartFromReservoir()`. Le réservoir avait envoyé ce signal via sa boîte à messages `OUT` « CrueStart » et la méthode `sendStartToCrue()`.
3. « ReservoirDiffNiveau » est une boîte à messages `OUT` chargée d'envoyer les caractéristiques de la crue susceptibles d'influer sur le niveau d'eau au réservoir qui calcule son évolution. Ces informations transitent via la méthode `sendDiffNiveauToReservoir()`. Le réservoir réceptionne l'information via sa boîte à messages `IN` « AmontDiffNiveau » et la méthode `getDiffNiveauFromAmont()`. Cette boîte à messages est synchronisée avec son équivalent de la classe Vanne. Cela signifie que le passage par la méthode `sendDiffNiveauToReservoir()` de la classe Crue appelle automatiquement la méthode `sendDiffNiveauToReservoir()` de la classe Vanne.
4. « AvalPhase » est une boîte à messages `OUT` chargée d'envoyer le signal d'établissement de la crue via la méthode `sendPhaseToAval()`. Le composant aval concerné réceptionne l'information via sa boîte à messages `IN` « CruePhase » et la méthode `getPhaseFromCrue()`.

4.2.1.2.2 Modélisation d'une vanne

L'automate de la vanne et la classe PyCATSHOO associée sont représentés sur la figure 4.10.

4.2.1.2.2.1 États

Avant la crue, une vanne est fermée (**`closed`**). Lorsque son ouverture est commandée, la vanne entame son processus d'ouverture (**`opening`**). Si la vanne refuse de d'ouvrir à la sollicitation ou si elle se bloque en cours d'ouverture, elle est stoppée (**`stopped`**). Dans le cas contraire, son ouverture est totale (**`open`**). Les états sont donc naturellement **`closed`**, **`opening`**, **`open`** et **`stopped`**.

4.2.1.2.2.2 Transitions et méthodes

A partir de l'état **`closed`**, la vanne commence à s'ouvrir avec la probabilité $1-p_{soll}$ et refuse de s'ouvrir avec la probabilité p_{soll}. La transition **`closed`**→**`opening [stopped]`**

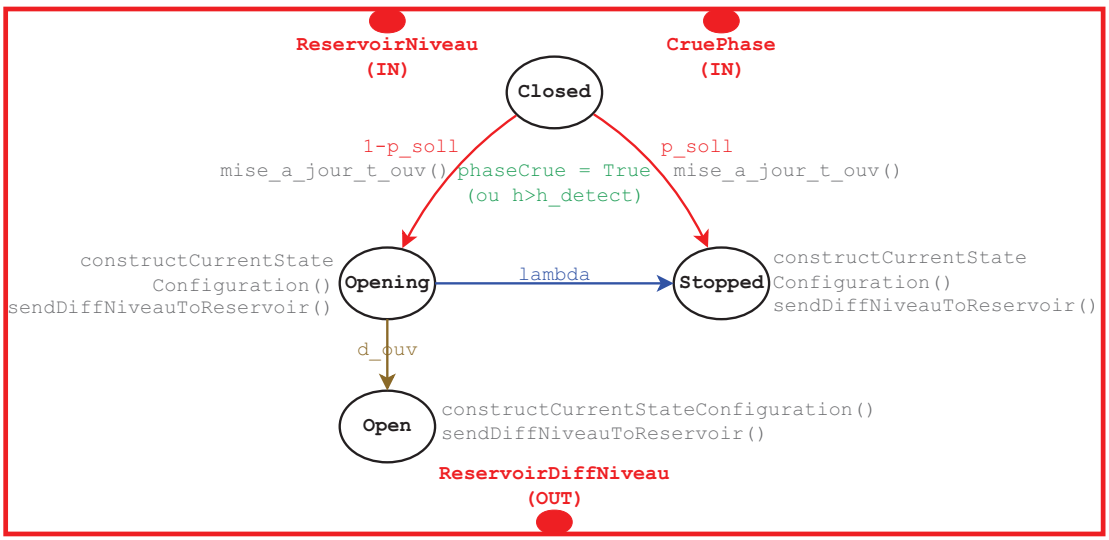

FIGURE 4.10 – Automate d'une vanne

est donc une transition instantanée stochastique. Cette transition est conditionnée par la réception d'un message tel que « niveau du réservoir supérieur à un seuil donné », « Crue établie » ou « Manœuvre de l'opérateur ». La réalisation de cette transition appelle la méthode `mise_a_jour_t_ouv()` qui mémorise l'instant de la transition dans la variable t_{v_0}.

La transition **`opening`**→**`stopped`** est une transition retardée stochastique qui dépend d'une loi exponentielle.

La transition **`opening`**→**`open`** est une transition retardée déterministe qui a lieu au bout de la durée $d_{ouv}^{(v)}$ s'il n'y a pas eu de transition **`opening`**→**`stopped`** auparavant.

L'arrivée dans les états **`opening`**, **`stopped`** et **`open`** appellent les méthodes `constructCurrentStateConfiguration()` et `sendDiffNiveauToReservoir()`. `constructCurrentStateConfiguration()` met à jour les caractéristiques de la vanne qui vont influer sur le niveau. `sendDiffNiveauToReservoir()` est similaire à la méthode de la classe Crue.

4.2.1.2.2.3 Boîtes à messages

La classe Vanne dispose de trois boîtes à messages.

1. « ReservoirNiveau » est une boîte à messages **`IN`** chargée de réceptionner la valeur du niveau du réservoir via la méthode `getNiveauFromReservoir()`. Cette boîte à messages est surtout utile en présence d'un asservissement muni d'un capteur. Le réservoir envoie cette information via sa boîte à messages **`OUT`** « CapteurNiveau » et la méthode `sendNiveauToCapteur()`.

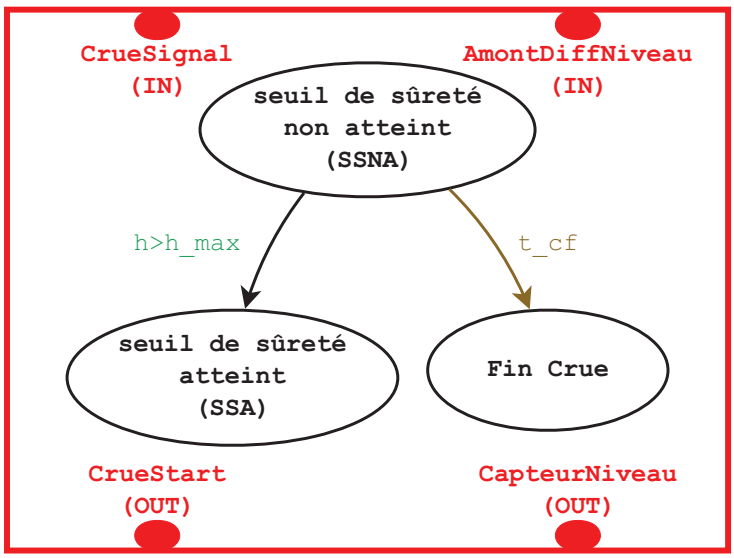

FIGURE 4.11 – Automate du réservoir

2. « CruePhase » est une boîte à messages **IN** chargée de réceptionner le signal d'établissement de la crue via la méthode `getPhaseFromCrue()`. La crue avait envoyé ce signal via sa boîte à messages **OUT** « AvalPhase » et la méthode `sendPhaseToAval()`.
3. « ReservoirDiffNiveau » est une boîte à messages **OUT** chargée d'envoyer les caractéristiques de la vanne susceptibles d'influer sur le niveau d'eau au réservoir qui calcule son évolution. Ces informations transitent via la méthode `sendDiffNiveauToReservoir()`. Le réservoir réceptionne l'information via sa boîte à messages **IN** « AmontDiffNiveau » et la méthode `getDiffNiveauFromAmont()`. Cette boîte à messages est synchronisée avec son équivalent de la classe Crue.

4.2.1.2.3 Modélisation du réservoir

La classe Reservoir héberge le contrôleur de PDMP (*PDMPControler*) qui implémente l'automate stochastique hybride chargé de piloter l'évolution du niveau à partir de l'équation différentielle qui la régit. L'ASH du réservoir et la classe PyCATSHOO associée sont représentés sur la figure 4.11.

4.2.1.2.3.1 États

L'état du réservoir dépend de l'événement « atteinte du seuil de sûreté » ou non. Les deux états **seuil de sûreté atteint (SSA)** et **seuil de sûreté non atteint (SSNA)** sont donc naturellement créés. Dans la suite de ce chapitre et dans les exemples du chapitre suivant, ce sont les notations **SSA** et **SSNA** qui seront utilisées pour désigner ces états. L'état **Fin Crue** est également créé pour signifier la fin de la crue.

4.2.1.2.3.2 Transitions et méthodes

La transition **SSNA→SSA** a lieu lorsque la condition $\{h > h_{max}\}$ est vérifiée, où h est le niveau dans la retenue et h_{max} le niveau maximal. Si la crue est évacuée correctement, c'est la transition **SSNA→Fin crue** qui termine l'histoire. La transition **SSNA→Fin crue** est une transition retardée déterministe.

4.2.1.2.3.3 Boîtes à messages

La classe Reservoir dispose de quatre boîtes à messages.

1. « CrueSignal » est une boîte à messages **IN** chargée de réceptionner le signal du début de l'histoire via la méthode `getSignalFromCrue()`. La crue avait envoyé cette information via sa boîte à messages **OUT** « ReservoirSignal » et la méthode `sendSignalToReservoir()`. La méthode `getSignalFromCrue()` appelle la méthode `sendStartToCrue()`.
2. « CrueStart » est une boîte à messages **OUT** chargée d'envoyer le signal « Start » via la méthode `sendStartToCrue()`. La crue réceptionne ce signal via sa boîte à messages **IN** « ReservoirStart » et la méthode `getStartFromReservoir()`.
3. « AmontDiffNiveau » est une boîte à messages **IN** chargée de réceptionner les caractéristiques de la vanne et de la crue susceptibles d'influer sur le niveau d'eau du réservoir qui calcule son évolution. Ces informations transitent via la méthode `getDiffNiveauFromAmont()`. La crue et la vanne ont envoyé ces informations via leur boîte à messages **OUT** « ReservoirDiffNiveau » et la méthode `sendDiffNiveauToReservoir()`.
4. « CapteurNiveau » est une boîte à messages **OUT** chargée d'envoyer la valeur du niveau du réservoir via la méthode `sendNiveauToCapteur()`. Le composant concerné, souvent le capteur d'un asservissement, réceptionne cette information via sa boîte à messages **IN** « ReservoirNiveau » et la méthode `getNiveauFromReservoir()`.

4.2.1.3 Modèle global du système simple

La dernière partie du travail de modélisation est la description précise de chaque évacuateur. Cette description liste chaque composant du système comme une instance de la classe qui lui correspond. Les liens entre composants sont également énumérés. Les données physiques et les données de fiabilité y sont renseignées.

Par exemple, la vanne V1 est une instance de la classe Vanne. Les paramètres sur les dimensions physiques de la vanne y sont précisés (hauteur, largeur, coefficient m, durée d'ouverture $d_{ouv}^{(v)}$), ainsi que les données de fiabilité (type de loi utilisé, taux de défaillance, probabilité de défaillance à la sollicitation).

Les boîtes à messages sont matérialisées par des liens. Ces liens peuvent relier un à plusieurs objets expéditeurs à un ou plusieurs objets destinataires. Par exemple le

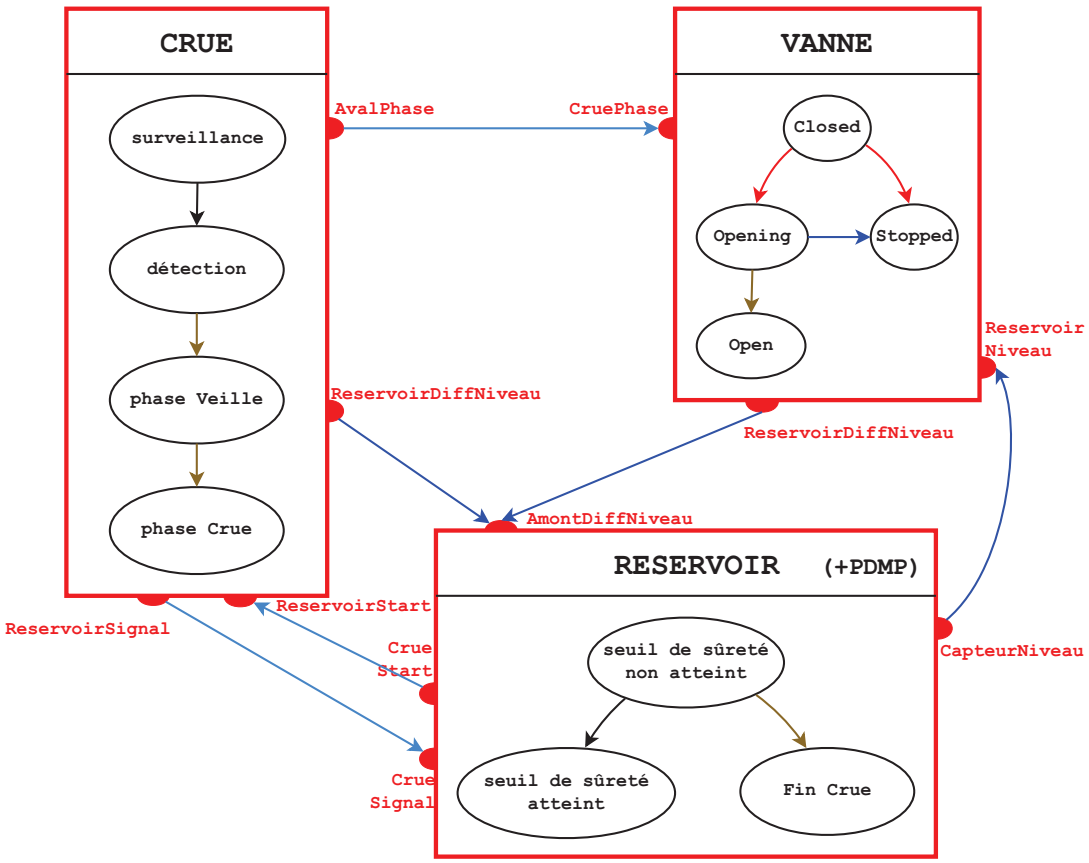

FIGURE 4.12 – Modèle global du système simple

lien [(V1, « ReservoirDiffNiveau »), (C, « ReservoirDiffNiveau »), (Res, « AmontDiff-Niveau »)] matérialise l'échange sur le différentiel de niveau dans le réservoir.

Le modèle global est représenté sur la figure 4.12.

4.2.1.4 Chronologie d'une histoire

L'enchainement de l'algorithme déroulé par PyCATSHOO est détaillé dans l'annexe A. PyCASTHOO stocke l'enchaînement des transitions de cette simulation et initialise une nouvelle simulation. Cet enchaînement de transitions est appelé histoire. L'histoire retournée pour cette simulation est de la forme
[(0, Reservoir, **SSNA**),
(0, Vanne, **closed**),
(0, Crue, **avant**),
(0, Crue, **après**),
(3600, Crue, **phase Veille**),
(57600, Crue, **phase Crue**),

(57900, Vanne, `opening`),
(60900, Vanne, `open`),
(237600, Reservoir, `Fin Crue`)].

4.2.2 Modélisation du problème industriel

A partir de cette base de connaissances dédiée à la modélisation du réservoir simple, la base de connaissances PyCATSHOO du problème industriel « évacuateurs de crues » est construite en ajoutant progressivement les composants des sous-systèmes décrits dans la section 4.1.3. Il s'est avéré que la plupart des composants ont un fonctionnement similaire, à quelques nuances près. Ce constat a mené à une refonte de la base de connaissances, dont la nouvelle version est plus générique.

Les composants sont des objets manoeuvrés par l'opérateur, alimentés par une source d'énergie et/ou réparables. Trois classes ObjetManoeuvré, ObjetAlimenté et ObjetRéparable sont donc définies génériquement pour décrire les automates associés. Cela évite de répéter la définition de motifs semblables pour chaque type de composants. Les classes Crue et Reservoir sont semblables à celles décrites pour le système simple. Le passage à l'échelle industrielle n'a pas apporté de modifications majeures à leur modélisation. Enfin, la classe Opérateur modélise le plus finement possible les actions de celui-ci.

Les trois premières sections décrivent la structure des classes ObjetManoeuvré, ObjetAlimenté et ObjetRéparable. La construction du modèle de l'opérateur est ensuite présentée, puis la section 4.2.2.5 expose la construction du modèle d'une vanne. Cela permet de constater le bénéfice du principe d'héritage de classes et le progrès de la modélisation depuis le système simple. Enfin ce chapitre se termine par la représentation de la structure des évacuateurs modélisés liée à la modélisation.

4.2.2.1 Modélisation d'un objet manoeuvré

4.2.2.1.1 Structure de l'automate

Les objets manoeuvrés sont les objets manipulés directement par l'opérateur. La plupart des objets manoeuvrés sont des composants du contrôle-commande (coffrets de commande locaux). C'est aussi le cas de composants de secours (diesels ou moteurs thermiques) et des bascules. Ils ont pour point commun un possible échec de leur démarrage, provoqué soit par une défaillance intrinsèque au matériel, soit par une mauvaise manipulation de l'opérateur. Ces deux événements ont pour probabilités respectives p_{soll} et p_{echec}. Certaines manœuvres sont plus urgentes que d'autres. De même, elles vont demander plus ou moins de temps à l'opérateur, et impliquent un délai de déplacement variable en fonction de la localisation du matériel. L'ASH de la figure 4.13 modélise ces critères.

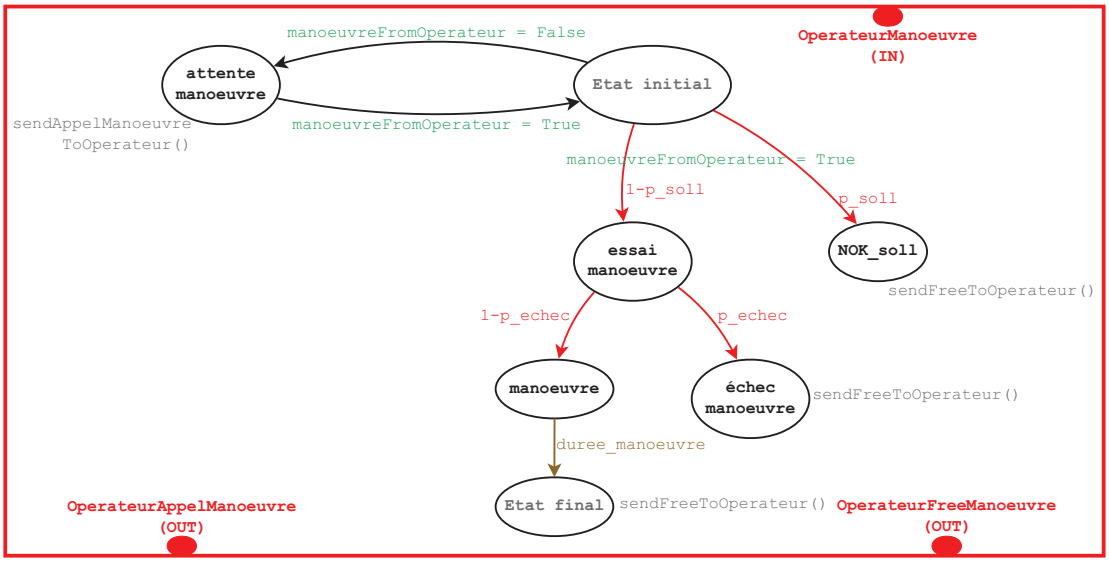

FIGURE 4.13 – Automate d'un objet manoeuvré

Les états **Etat initial** et **Etat final** ne sont pas fixés et leur intitulé est un paramètre précisé dans la description de la classe propre à l'objet considéré.

4.2.2.1.2 Boîtes à messages

La classe ObjetManoeuvré est dôtée de trois boîtes à messages. La boîte à messages **IN** « OperateurManœuvre » importe des informations sur la disponibilité de l'opérateur et sur sa probabilité p_{echec} d'échec de la manœuvre. Les boîtes à messages **OUT** « OperateurAppelManœuvre » et « OperateurFreeManœuvre » envoient respectivement le signal d'appel et le signal de fin de la manœuvre à l'opérateur, ainsi que des informations sur la priorité de la manœuvre et sur la localisation du composant à manœuvrer.

4.2.2.2 Modélisation d'un objet alimenté

4.2.2.2.1 Structure de l'automate

Exception faite des composants qui produisent de l'énergie ou qui l'acheminent vers le barrage (Réseau, Diesel ou Batterie), tous les composants ont besoin d'être alimentés pour fonctionner. Un tel objet n'est pas seulement « alimenté » ou « non alimenté » car des coupures d'alimentation temporaires sont possibles.

L'état **NonAlim** signifie une rupture définitive de l'alimentation et donc une mise hors service du composant concerné jusqu'à la fin de l'histoire. Inversement, l'état **NonAlimTemp1** indique qu'aucun des composants supposés l'alimenter n'est allumé. L'état **NonAlimTemp2** mentionne la même chose, après avoir testé le bon fonctionnement du composant. Enfin, l'état **AttAlim** est activé après vérification qu'une autre

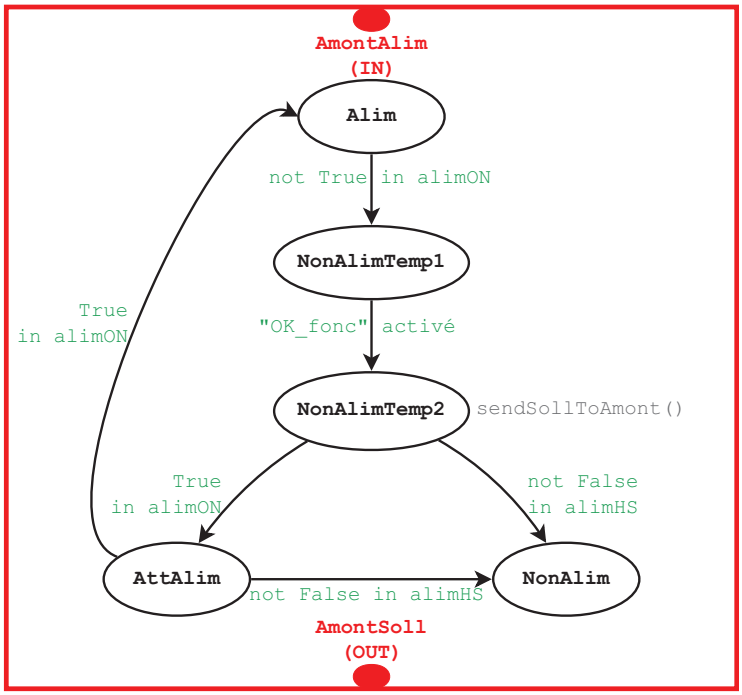

FIGURE 4.14 – Automate d'un objet alimenté

source d'alimentation peut prendre le relais.

Parmi les attributs de la classe ObjetAlimenté, les listes $alimON$, $alimATT$ et $alimHS$ caractérisent l'état des différentes sources d'alimentation par des booléens. Par exemple, un composant c bénéficie de trois sources A_1, A_2 et A_3. A_1 est hors service, A_2 est allumé et A_3 est en attente d'une réparation ou d'une manœuvre de l'opérateur. Alors $c.alimON = [0, 1, 0]$ indique que seul A_2 est ON ; $c.alimATT = [0, 0, 1]$ mentionne l'attente de A_3 et $c.alimHS = [1, 0, 0]$ précise que A_1 est définitivement HS.

La figure 4.14 illustre ces propos.

4.2.2.2.2 Boîtes à messages

La classe ObjetAlimenté est munie de deux boîtes à messages. La boîte à messages **`IN`** « AmontAlim » reçoit les informations sur l'état des sources d'alimentation et les stocke dans des attributs $alimON$, $alimATT$ et $alimHS$. La boîte à messages **`OUT`** « AmontSoll » fait remonter le signal de sollicitation à ces sources en cas d'activation de l'état **`NonAlimTemp2`**.

4.2.2.2.3 Cas de la classe « Bascule »

La classe Bascule hérite des deux classes ObjetManoeuvré et ObjetAlimenté. Cela permet d'insérer les étapes d'une manœuvre entre les états **`NonAlimTemp1`** et

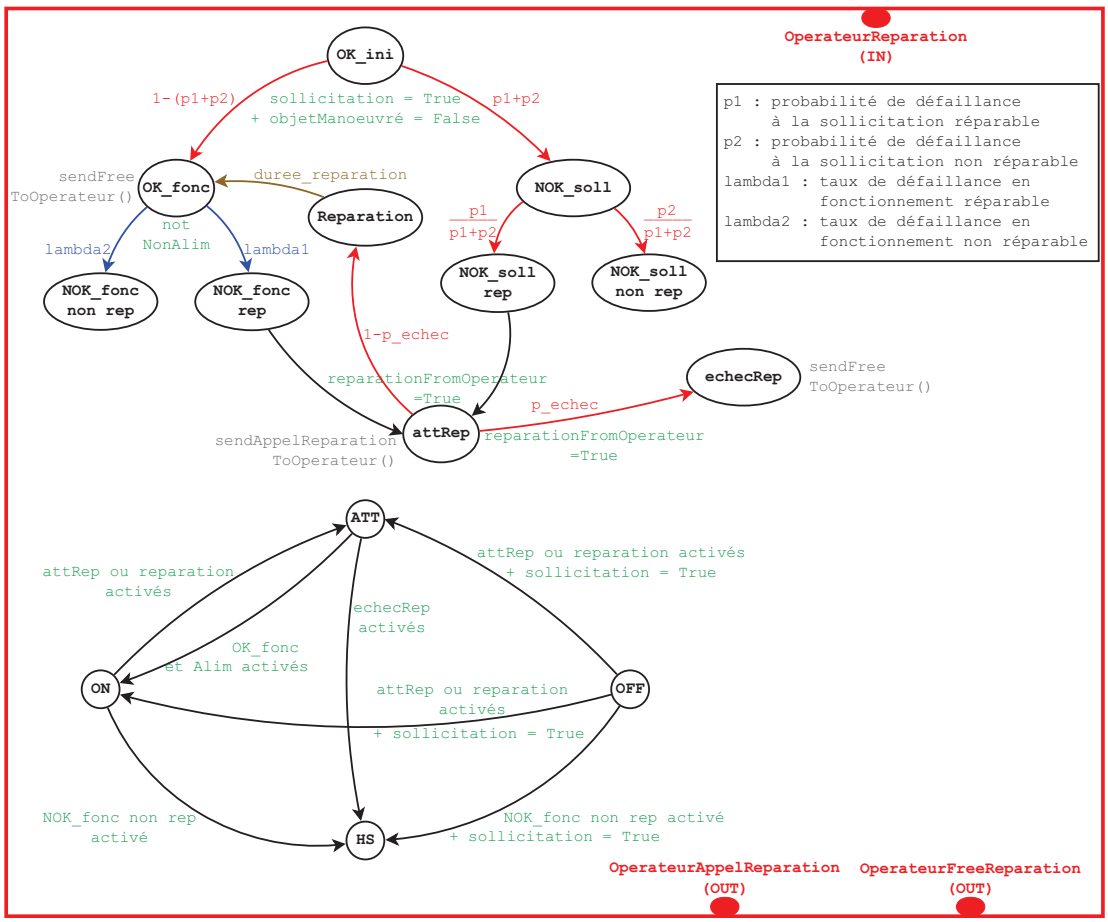

FIGURE 4.15 – Automate d'un objet réparable

NonAlimTemp2. Ainsi, tous les cas de figure détaillés dans la section **??** sont modélisés.

4.2.2.3 Modélisation d'un objet réparable

4.2.2.3.1 Structure de l'automate

La plupart des objets sont susceptibles d'être défaillants, puis réparés. Cette caractéristique commune est présente dans la structure de l'automate décrivant ce mécanisme. Chaque classe d'objet hérite ensuite de la classe ObjetRéparable. En réalité, un objet réparable est décrit par deux ASH distincts. Le premier décrits les modes de défaillance, le second les modes de fonctionnement, comme l'illustre la figure 4.15.

Avant toute sollicitation, l'automate est initialisé en activant l'état **OK_ini**. La transition **OK_ini→OK_fonc [NOK_soll]** est une transition instantanée stochastique qui se réalise à partir du moment où deux conditions sont réunies. La première condition est la réception d'un signal de sollicitation. En deuxième lieu, le composant concerné ne doit

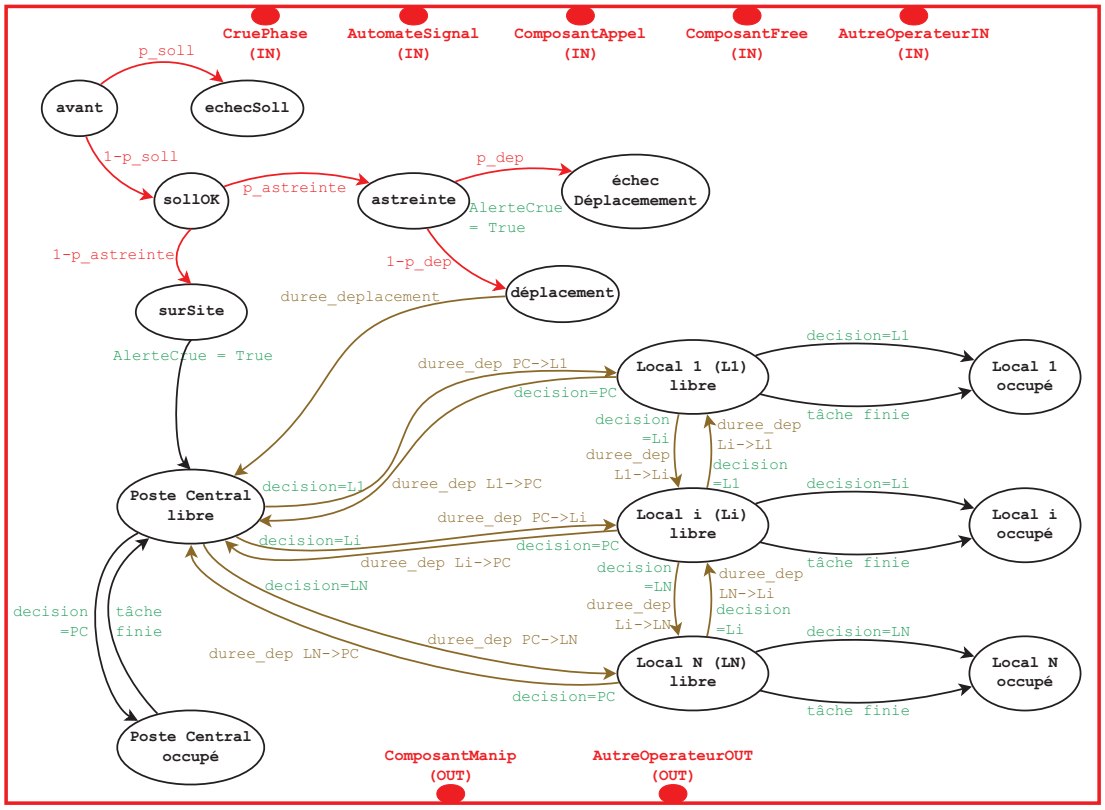

FIGURE 4.16 – Automate d'un opérateur

pas hériter de la classe ObjetManoeuvré. Si c'est le cas, la transition **OK_ini→OK_fonc [NOK_soll]** est remplacée par la séquence de transitions caractéristiques d'un objet manoeuvré, où **état initial** = **OK_ini** et **état final** = **OK_fonc** (figure 4.13). Cette injection d'un automate dans un autre permet d'introduire la durée de manœuvre nécessaire à l'opérateur.

Si une défaillance a lieu à la sollicitation (arrivée dans l'état **NOK_soll**), ce peut être une défaillance réparable (**NOK_soll rep**) ou non réparable **(NOK_soll non rep**). De même, une défaillance en fonctionnement peut être réparable (**NOK_fonc rep**) ou non réparable (**NOK_fonc non rep**). Une défaillance réparable provoque l'activation de l'état **attRep** (Attente de réparation) dont le mécanisme est similaire à l'état **attente manœuvre** de l'objet manoeuvré. La probabilité d'échec de la réparation et la durée de la réparation sont ainsi prises en compte avant un éventuel retour dans l'état **OK_fonc**.

Les états **ON**, **ATT** (attente de réparation), **OFF** (pas encore sollicité) et **HS** (panne définitive) sont créés. Les transitions entre ces états dépendent des états activés dans l'ASH décrit précédemment. Certaines transitions sont également conditionnées par l'activation de l'état **Alim** si le composant est également un objet alimenté.

4.2.2.3.2 Boîtes à messages

Les boîtes à messages « OperateurReparation », « OperateurAppelReparation » et « OperateurFreeReparation » ont un rôle similaire aux boîtes à messages « OperateurManœuvre », « OperateurAppelManœuvre » et « OperateurFreeManœuvre » de la classe ObjetManœuvré.

4.2.2.4 Modélisation d'un opérateur

L'opérateur est mobile et ne peut être qu'à un endroit à la fois. De même, il ne peut être occupé qu'à une tâche à la fois. La classe Opérateur décrit les allers et venues en considérant les durées de déplacement et les possibilités d'échec des actions de l'opérateur, ce qui est illustré par la figure 4.16.

4.2.2.4.1 Structure de l'automate

A l'instant initial, l'opérateur est caractérisé par une probabilité p_{soll} d'échec de sa sollicitation. Si la sollicitation réussit et que l'attention de l'opérateur est suscitée, les événements suivants dépendent de sa localisation, qui est aléatoire. Dans le cas où l'opérateur est d'astreinte, son déplacement peut échouer avec la probabilité p_{dep}. Une fois sur site, les actions de l'opérateur dépendent d'une *todo list*. Cette liste répertorie les composants qui appellent l'attention de l'opérateur, soit pour une manœuvre, soit pour une réparation, ainsi que leur localisation et la priorité de la manipulation. Un algorithme désigne la prochaine action de l'opérateur en fonction de ces critères et la stocke dans la variable *decision*.

4.2.2.4.2 Boîtes à messages

Les boîtes à messages `IN` « CruePhase » et « AutomateSignal » sont chargées de réceptionner le signal d'arrivée de la crue dans la variable *AlerteCrue*. Les boîtes à messages `IN` « ComposantAppel » et « ComposantFree » reçoivent les sollicitations pour manœuvrer ou réparer tel ou tel composant. La boîte à messages `OUT` « ComposantManip » envoie au composant le signal de début de manipulation ainsi que la probabilité d'échec associée. Enfin, les boîtes à messages « AutreOpérateurIN » et « AutreOpérateurOUT » sont destinées à un éventuel échange entre deux opérateurs qui se partagent les tâches.

4.2.2.5 Modélisation d'une vanne

4.2.2.5.1 Structure de l'automate

La classe Vanne reçoit les caractéristiques des classes ObjetAlimenté et ObjetRéparable. Les automates dont la structure est héritée de la classe ObjetAlimenté sont représentés sur la figure 4.17 par la couleur gris-vert ; ceux dont la structure provient de la classe ObjetRéparable sont en gris-rose. Les vannes de surface et les vannes de demi-fond ne diffèrent que par les paramètres de l'équation de débit sortant. L'automate

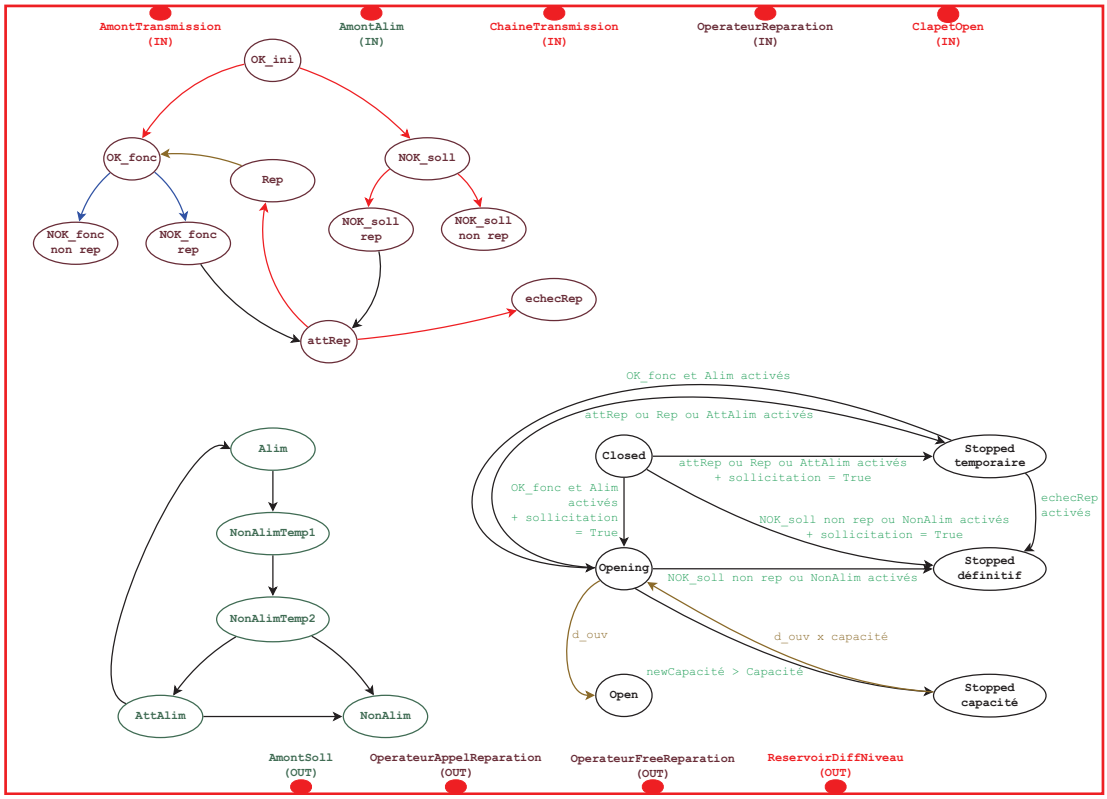

FIGURE 4.17 – Automate Stochastique Hybride d'une vanne

d'un clapet est légèrement différent dans la définition des conditions de la transition **Closed→Opening**. En effet, la vanne ne commence à s'ouvrir qu'après avoir reçu l'information d'ouverture complète du clapet, via la boîte à messages **IN** « ClapetOpen ». Le clapet possède une boîte à messages **OUT** « VanneOpen ».

Trois particularités propres aux vannes empêchent l'utilisation de l'automate **ON/ATT/OFF/HS** pour modéliser le processus d'ouverture.

1. L'état **ON** ne permet pas de décrire la différence entre « en cours d'ouverture » et « complètement ouvert ». C'est pourquoi les états **opening** et **open** ont été retenus.
2. Dans le cas de certains barrages, l'ouverture d'une vanne est conditionnée par l'ouverture totale du clapet qui lui est associé. Le signal de sollicitation est donc défini au cas par cas, en fonction de la nature de la vanne. Ce peut être la réception du signal de démarrage de la transmission ou celui de fin d'ouverture du clapet.
3. Le processus d'ouverture de la vanne peut être stoppé pour plusieurs raisons. Une défaillance non réparable, l'attente d'une manœuvre ou une capacité moindre de l'actionneur seront occasionnées et résolues par des transitions différentes. Ceci justifie la création des états **Stopped définitif**, **Stopped temporaire** et **Stopped capacité**.

Pour ne pas utiliser l'automate **ON/ATT/OFF/HS** hérité de la classe ObjetRéparable, il suffit de n'activer aucun de ces états lors de l'initialisation des histoires.

4.2.2.5.2 Boîtes à messages

En héritant des classes ObjetAlimenté et ObjetManoeuvré, l'objet Vanne dispose déjà des boîtes à messages « AmontAlim », « AmontSoll », « OperateurReparation », « OperateurAppelReparation » et « OperateurFreeReparation ».

Les boîtes à messages **IN** « AmontTransmission » et « ChaineTransmission » reprennent le même principe que « AmontAlim », mais en y ajoutant les spécificités des transmissions. Il s'agit d'informations sur la capacité de l'actionneur ou le défaut de synchronisation des chaînes.

Comme pour le système simple, la boîte à messages **OUT** « ReservoirDiffNiveau » est chargée d'envoyer les caractéristiques de la vanne et de la crue susceptibles d'influer sur le niveau d'eau au réservoir qui calcule son évolution.

Enfin, la boîte à messages **IN** « ClapetOpen » réceptionne le signal de fin d'ouverture du clapet, dans le cas où la vanne est surmontée d'un clapet.

4.2.2.6 Représentation des deux évacuateurs de crues

Qu'ils appartiennent au sous-système hydromécanique, d'alimentation ou au contrôle-commande, chaque type de composant est modélisé par un objet PyCATSHOO. Cet objet est un ensemble de données rassemblées dans des classes. A l'image de la classe Vanne, les classes héritent systématiquement d'une, deux ou trois classes parmi les classe ObjetManoeuvré, ObjetRéparable et ObjetAlimenté de la BdC générique.

La structure des évacuateurs étudiés est confidentielle, aussi leur représentation n'est-elle disponible que dans une version confidentielle, de diffusion restreinte, de ce manuscrit. Etant donnée la dimension d'un système industriel, il n'est plus possible de représenter ensemble les automates des composants ainsi que les interactions entre ceux-ci, comme sur la figure 4.12 qui décrit le fonctionnement du système simple. A moins de disposer d'un support interactif, il est recommandé de représenter les automates de chaque classe de composant, puis la structure des évacuateurs étudiés. Un support interactif permettrait, à partir de l'architecture d'un EdC, de cliquer sur le composant d'intérêt pour afficher son automate.

4.2.3 Conclusion

Les évacuateurs de crues sont des systèmes dont la modélisation est un enjeu industriel pour l'évaluation de la sûreté. La compréhension du processus de crue et de son évacuation, la définition des conditions à l'origine d'une transition entre deux états, la

description de l'impact d'une transition sur les autres composants sont autant d'étapes de cette modélisation. Les automates stochastiques hybrides fournissent un formalisme capable de représenter ces étapes. Ce formalisme inclut le calcul de l'évolution d'une variable déterministe continue, dont les paramètres sont réajustés à chaque événement aléatoire. Toute classe de composant est associée à un ASH capable de détailler avec souplesse et lisibilité tous les cas de figure existant dans la réalité. Les boîtes à messages participent à la synchronisation des automates. La représentation de ces flux d'informations entre composants suffit à avoir une image du fonctionnement d'un système particulier, complété par la visualisation des automates concernés. Ainsi, tout système, quelles que soient sa taille et sa complexité, peut être modélisé avec les automates stochastiques hybrides.

La complexité de la modélisation dépend de la complexité du système étudié, mais pas de la dimension du système. En effet, un système dynamique hybride est rendu complexe par sa taille, par le nombre et la nature des interactions entre les composants et par la considération de la dynamique d'une variable déterministe continue. La variété des interactions augmente le nombre de boîtes à messages à définir dans la base de connaissances. Le nombre de classes dans cette base dépend de la variété des composants, mais pas de leur quantité. La croissance du système, en nombre de composants, implique donc l'ajout de nouveaux automates, mais pas l'explosion des automates déjà présents. Les interactions entre les composants sont matérialisées par des liens entre les boîtes à messages. Ces liens sont au nombre de 90 pour un évacuateur réel de 75 composants. Si le nombre de liens n'est pas ici soumis à l'explosion combinatoire, l'analyste doit toutefois se montrer attentif pour ne pas en oublier lors de la phase d'instanciation du système.

L'outil PyCATSHOO est une aide à la construction de cette modélisation. Après l'analyse fonctionnelle d'une classe de système, la création d'une base de connaissances repose sur la définition d'états, de transitions et de boîtes à messages. Ces éléments doivent être communs à tous les systèmes représentatifs de la classe étudiée. Cela requiert l'identification de caractéristiques invariantes et la factorisation de ces caractéristiques au sein d'une seule base de connaissances. La construction d'une nouvelle base de connaissances en suivant cette méthodologie suivra toujours ce principe. A partir des automates contenus dans la base de connaissances, PyCATSHOO génère aléatoirement autant de simulations que demandé.

L'évolution des variables continues est une fonction de l'état des composants ou d'événements élémentaires. Par exemple, seule l'ouverture ou non des vannes influe sur le niveau. L'état des autres composants n'a pas d'impact direct sur l'évolution du niveau, il modifie seulement celui des vannes. Cette modélisation dispense de la construction d'un espace d'états de dimension importante, étape nécessaire dans le produit d'automates stochastiques hybrides non distribués.

Les résultats de ces simulations sont appelés « histoires ». Une histoire est la séquence de tous les états activés, ainsi que la date de cette activation, lors du passage de l'algorithme lors d'une simulation. Le fruit de plusieurs milliers de simulations est donc

une liste de plusieurs milliers d'histoires. Cette liste suppose un traitement ultérieur afin d'en déduire des résultats lisibles. La complexité du système apparaît en effet dans les résultats par le nombre et la longueur des histoires.

Même si l'illustration de la faisabilité de la modélisation de systèmes complexes est apportée ici, il faut donc une méthode pour extraire, synthétiser et utiliser l'information issue de la simulation du modèle. C'est l'objet des deux chapitres qui suivent.

Chapitre 5

Analyse des histoires et quantification probabiliste de la fiabilité

Ce chapitre présente les différents indicateurs de fiabilité obtenus à partir des histoires générées par l'outil PyCATSHOO. La probabilité d'occurrence de l'événement redouté et son évolution dans le temps est un indicateur usuel introduit dans la section 5.2. La section 5.3 explique comment rassembler les histoires en coupes afin de repérer quels groupes d'événements ont le plus de poids dans la réalisation de l'événement redouté.

L'événement redouté (ER) est le dépassement d'un seuil h_{max} par le niveau de la retenue. La dynamique de l'évolution du niveau dans la retenue est telle que cet événement redouté ne survient qu'au bout d'un certain délai après la ou les défaillance(s) des composants. Est-il possible d'anticiper l'occurrence de cet événement à partir des instants de défaillance ? A partir des durées de fonctionnement sans défaillance de chaque composant, un modèle est proposé pour classifier les histoires en fonction de leur issue ; cette démarche est exposée dans la section 5.4.

5.1 Introduction

5.1.1 Objectifs de la quantification

A travers une démarche de synthèse et d'abstraction, l'objectif et de synthétiser, à partir des histoires, des indicateurs ou des « métriques » de haut niveau qui caractérisent la fiabilité et la sûreté du système. Ces indicateurs peuvent être utiles dans l'évaluation de la sûreté et de la criticité, ainsi que dans le pronostic de l'issue de certaines situations. Cette aide à l'identification des composants les plus critiques et à la priorisation des

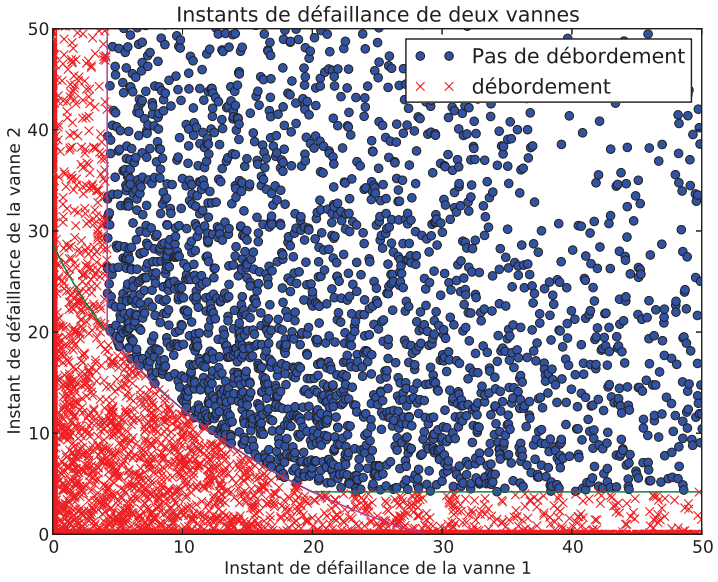

FIGURE 5.1 – Occurrence de l'événement redouté en fonction des instants de défaillance de deux vannes

actions de maintenance fait partie des objectifs généraux de l'analyse des histoires et de la quantification probabiliste de la sûreté développées dans ce chapitre.

La figure 5.1 illustre la frontière entre les histoires à succès et les histoires à défaillance.

Cette classification des histoires est à l'origine de la caractérisation d'une fonction f. Cette fonction dépend des dates de défaillance en fonctionnement $(T_1, ..., T_n)$ des n composants et retourne l'occurrence ou non de l'événement redouté, ici un débordement du barrage. La probabilité d'occurrence de l'événement redouté s'obtient en intégrant cette fonction, associée aux fonctions de densité f_i de probabilité des temps de défaillance de chaque composant i :

$$P(ER) = \int_0^\infty ... \int_0^\infty f(T_1, ..., T_n) \prod_{i=1}^{n} f_i(T_i) dT_1 ... dT_n. \tag{5.1}$$

5.1.2 Démarche : de KB3 à PyCATSHOO

L'outil GASPART, présenté dans le chapitre 1, repose sur la plate-forme logicielle multi-domaines KB3. Parmi d'autres modules, KB3 possède deux modules de quantification nommés FIGSEQ et YAMS [Chraibi, 2013b]. Ce sont deux outils développés par EDF. FIGSEQ permet de construire de manière quasi exhaustive tous les scénarios à partir d'un initiateur donné, jusqu'à l'événement redouté. Chaque scénario, ou

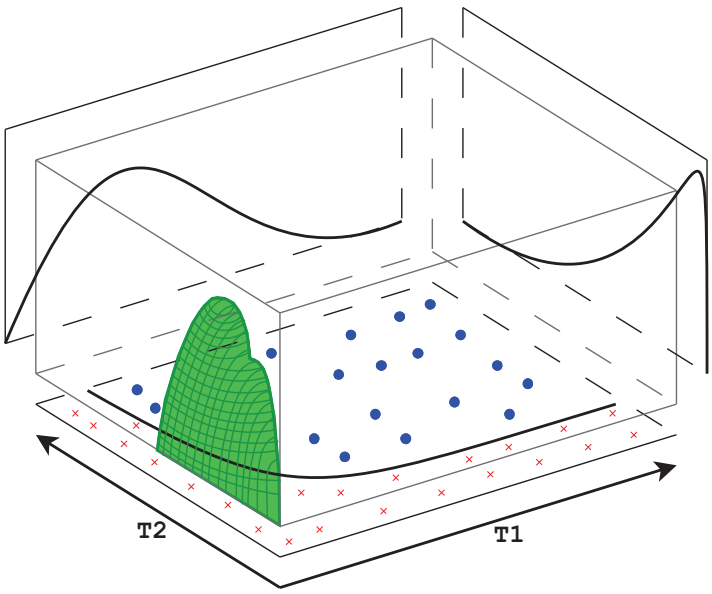

FIGURE 5.2 – Probabilité de l'événement redouté en fonction de la frontière et des densités de probabilités des instants de défaillance de deux composants

séquence, est associé à sa probabilité d'occurrence et les séquences prépondérantes sont ainsi identifiées. YAMS procède par tirage aléatoire des instants d'occurrence de l'ensemble des événements pouvant se produire dans une situation donnée. Ainsi, YAMS, à la différence de FIGSEQ, permet de considérer des défaillances en fonctionnement, même lorsque les processus ne sont pas markoviens. YAMS évalue la probabilité d'occurrence de l'événement redouté mais ne construit pas de manière directe et déterministe les séquences prépondérantes.

Deux projets ont été initiés pour pallier à ce manque d'un unique outil adapté aux systèmes dynamiques hybrides. Il s'agit de ces travaux de thèse et de la création de l'outil PyCATSHOO par les chercheurs d'EDF ; ces deux projets n'étaient pas liés au départ. La première étape des travaux de thèse fut l'ébauche d'un petit outil de modélisation d'un réservoir simple, dans le but de coupler les méthodes de simulation de Monte Carlo et d'exploration de séquences. Cet outil reposait déjà sur une programmation en Python orientée objet. Puis ces travaux de thèse se sont orientés vers l'utilisation et la validation de l'outil PyCATSHOO. Après la construction de la base de connaissances PyCATSHOO dédiée aux évacuateurs de crues, la formalisation des résultats et l'exploitation des données temporelles contenues dans les histoires ont constitué le dernier volet de la thèse. Cette démarche est présentée dans le cadre des évacuateurs de crues, mais la méthodologie proposée est applicable aux systèmes dynamiques hybrides en général.

5.1.3 Formalisation des résultats : séquences, histoires et vecteurs de durées

A partir des objets définis dans la base de connaissances et de leur structure, l'algorithme de PyCATSHOO génère aléatoirement des séquences d'événements. Ces séquences sont la liste des états activés pendant le cheminement de l'algorithme, guidé par l'architecture des automates stochastiques hybrides. Chaque information sur l'activation d'un état est agrémentée du nom du composant concerné et de la date de l'activation. A la fin d'une simulation, PyCASTHOO mémorise la séquence générée avant d'initialiser la simulation suivante.

5.1.3.1 Définition d'une séquence

Soit n_o le nombre d'objets différents définis dans une base de connaissances. Le comportement d'un objet est représenté par un ou plusieurs automates distincts. Soit $o_i \in [o_1, ..., o_{n_o}]$ un de ces objets et $n_A^{o_i}$ le nombre d'automates chargés de décrire l'objet o_i.

Un automate est composé de plusieurs états ou modes. Soit $A_j^{o_i} \in [A_1^{o_i}, ..., A_{n_A^{o_i}}^{o_i}]$ un automate retraçant une partie du fonctionnement de l'objet o_i et $n_e^{A_j^{o_i}}$ le nombre d'états composant cet automate. Notons $[e_1^{A_j^{o_i}}, ..., e_{n_e^{A_j^{o_i}}}^{A_j^{o_i}}]$ ces états.

Un objet peut être instancié une ou plusieurs fois. C'est le cas par exemple lorsque le système est composé de trois vannes de même type. Notons $n_c^{o_i}$ le nombre d'instances de l'objet o_i et $c_l^{o_i} \in [c_1^{o_i}, ..., c_{n_c^{o_i}}^{o_i}]$ une instance de l'objet o_i.

Définition 27. Une séquence est la suite chronologique des états activés lors d'une simulation. Une séquence s'écrit donc

$$\left(\left[t_{c_l}\left(e_k^{A_j^{o_i}}\right), c_l^{o_i}, e_k^{A_j^{o_i}}\right]\right)_{1\leq i\leq n_o,\, 1\leq j\leq n_A^{o_i},\, 1\leq k\leq n_e^{A_j^{o_i}},\, 1\leq l\leq n_c^{o_i}}$$

où $t_{c_l}\left(e_k^{A_j^{o_i}}\right)$ désigne la date de l'activation de l'état $e_k^{A_j^{o_i}}$ de l'automate $A_j^{o_i}$ pour l'instance $c_l^{o_i}$.

Exemple 1
Considérons le système simple de la section 4.2.1, composé d'un réservoir soumis à une crue et vidangé par une vanne. Les objets présents sont Réservoir, Crue et Vanne. Chaque objet n'est décrit que par un seul automate. Ces trois automates sont respectivement A^R, A^C et A^V. Les états de A^R sont $\texttt{SSNA}^{A^R}$, $\texttt{SSA}^{A^R}$ et $\texttt{Fin Crue}^{A^R}$. Les états de A^C sont $\texttt{surveillance}^{A^C}$, $\texttt{détection}^{A^C}$, $\texttt{phase Veille}^{A^C}$ et $\texttt{phase Crue}^{A^C}$.

Les états de A^V sont $\texttt{closed}^{A^V}$, $\texttt{opening}^{A^V}$, $\texttt{open}^{A^V}$ et $\texttt{stopped}^{A^V}$. Soit R l'instance du Reservoir, C une instance de la Crue et V une instance de la Vanne.

La séquence retournée par PyCATSHOO après une simulation du fonctionnement sans défaillance du système simple est ([0, R, **SSNA**], [0, V, **closed**], [0, C, **surveillance**], [0, C, **détection**], [3600, C, **phase Veille**], [57600, C, **phase Crue**], [57900, V, **opening**], [60900, V, **open**], [237600, R, **Fin Crue**]).

Cette séquence est bien de la forme $\Big(\Big[t_R(\texttt{SSNA}^{A^R}),\ \text{R},\ \texttt{SSNA}^{A^R}\Big],$ $\Big[t_V(\texttt{closed}^{A^V}),\ \text{V},\ \texttt{closed}^{A^V}\Big],$ $\Big[t_C(\texttt{surveillance}^{A^C}),\ \text{C},\ \texttt{surveillance}^{A^C}\Big],$ $\Big[t_C(\texttt{détection}^{A^C}),\ \text{C},\ \texttt{détection}^{A^C}\Big],$ $\Big[t_C(\texttt{phase Veille}^{A^C}),\ \text{C},\ \texttt{phase Veille}^{A^C}\Big],$ $\Big[t_V(\texttt{closed}^{A^V}),\ \text{V},\ \texttt{closed}^{A^V}\Big],$ $\Big[t_V(\texttt{opening}^{A^V}),\ \text{V},\ \texttt{opening}^{A^V}\Big],$ $\Big[t_R(\texttt{End Crue}^{A^R}),\ \text{R},\ \texttt{Fin Crue}^{A^R}\Big]\Big)$.

Remarque sémantique
Le terme « séquence » prête à confusion et son utilisation sera limitée dans cette thèse. En effet, une séquence d'événements désigne en général dans la littérature une suite ordonnée d'événements. Si la notion d'ordre est importante dans la définition d'une séquence, celle de la date des événements n'est pas courante. L'information temporelle est centrale dans l'analyse des résultats, c'est pourquoi l'emploi du terme « histoire » sera privilégié pour désigner les sorties de PyCATSHOO. Les histoires offrent l'avantage de conserver l'information sur les dates des événements (et pas seulement leur ordre).

5.1.3.2 Définition d'une histoire

PyCATSHOO est initialement programmé pour stocker l'historique de toutes les activations d'états. Or l'activation des états initiaux est systématique et ne constitue pas une information intéressante. Il en est de même pour l'activation d'états intermédiaires ou « virtuels » créés uniquement pour le bon déroulement du processus de simulation.

Soit $A_j^{o_i}$ l'automate d'un objet o_i et $[e_1^{A_j^{o_i}}, ..., e_{n_e^{A_j^{o_i}}}^{A_j^{o_i}}]$ les états de cet automate. On notera $e_{k*}^{A_j^{o_i}}$ un état dont il sera intéressant de suivre l'activation et $E_*^{A_j^{o_i}}$ la liste de ces états intéressants.

Soit $s = [e_1, ..., e_N]$ une séquence composée de N triplets $\left(\left[t_{c_l}\left(e_k^{A_j^{o_i}}\right), c_l^{o_i}, e_k^{A_j^{o_i}}\right]\right)_{1\leq i\leq n_o,\ 1\leq j\leq n_A^{o_i},\ 1\leq k\leq n_e^{A_j^{o_i}},\ 1\leq l\leq n_c^{o_i}}$.

Définition 28. Une histoire $h = [e_1^*, ..., e_{N^*}^*]$ est composée de N^* triplets $\left(\left[t_{c_l}\left(e_{k*}^{A_j^{o_i}}\right), c_l^{o_i}, e_{k*}^{A_j^{o_i}}\right]\right)_{1\leq i\leq n_o,\ 1\leq j\leq n_A^{o_i},\ 1\leq k\leq n_e^{A_j^{o_i}},\ 1\leq l\leq n_c^{o_i}}$ tels que $e_{k*}^{A_j^{o_i}} \in \{s \cap E_*^{A_j^{o_i}}\}$.

Une histoire est une séquence dont on ne garde que les événements intéressants. Les états intéressants sont précisés au moment de leur définition dans la base de connaissances. Le fait de stocker et de retourner des histoires à la place des séquences permet un gain de temps et de mémoire.

Exemple 2
Soient $E_*^{A^R} = \left\{\texttt{SSNA}^{A^R},\ \texttt{Fin Crue}^{A^R}\right\}$, $E_*^{A^C} = \emptyset$ et $E_*^{A^V} = \left\{\texttt{opening}^{A^V},\ \texttt{open}^{A^V},\ \texttt{stopped}^{A^V}\right\}$. L'histoire correspondant à la séquence de l'exemple précédent est donc ([57900, V, **opening**], [60900, V, **open**], [237600, R, **Fin Crue**]).

5.1.3.3 Définition d'un vecteur de durées de fonctionnement sans défaillance

Certaines analyses quantitatives ont besoin de connaître la position de chaque date d'activation par rapport à la chronologie de la crue. Dans ce cas, toutes les informations répertoriées dans les histoires sont utilisées. Ce sera par exemple le cas de l'estimation de l'importance dynamique d'un composant en fonction du déroulement de la crue.

Pour d'autres analyses, seule la durée de fonctionnement avant défaillance est nécessaire. C'est le cas lorsque le modèle est suffisamment puissant pour reconstituer toute la chronologie à partir des instants de ces défaillances, comme notre modèle de classification des histoires en fonction de leur issue. Ceci justifie la définition d'un vecteur de durées de fonctionnement à partir d'une histoire.

Remarque 1. Calculer une durée de fonctionnement sans défaillance implique de connaître la date de début et la date de fin de ce fonctionnement. La fin est provoquée par une défaillance ou par l'achèvement du processus étudié (par exemple, l'aboutissement du processus d'ouverture de la vanne). Il est donc nécessaire de placer dans les états intéressants ces états correspondant au début et aux deux fins possibles du fonctionnement d'un objet.

Notations Ces états ne sont présents que dans un seul automate, même si l'objet est défini par plusieurs ASH. Désigner l'état $e_{k*}^{A_j^{o_i}}$ par $e_{k*}^{o_i}$ permet d'alléger les notations.

Soit $h = [e_1^*, ..., e_{N^*}^*]$ une histoire composée de N^* triplets $([t_{c_l}(e_{k*}^{o_i}), c_l^{o_i}, e_{k*}^{o_i}])$.

Pour un objet o_i, $e_{0*}^{o_i}$ est l'état de fonctionnement, $e_{s*}^{o_i}$ est l'état après succès du processus et $e_{f*}^{o_i}$ est l'état après défaillance du processus.

Une histoire va donc contenir les triplets $([t_{c_l}(e_{0*}^{o_i}), c_l^{o_i}, e_{o*}^{o_i}], [t_{c_l}(e_{s*}^{o_i}), c_l^{o_i}, e_{s*}^{o_i}])$ en cas de fonctionnement sans défaillance du composant $c_l^{o_i}$ ou les triplets $\left([t_{c_l}(e_{0*}^{o_i}), c_l^{o_i}, e_{o*}^{o_i}], \left[t_{c_l}\left(e_{f*}^{o_i}\right), c_l^{o_i}, e_{f*}^{o_i}\right]\right)$ en cas de défaillance du composant $c_l^{o_i}$.

Définition 29. Soit T_{c_l} la durée de fonctionnement sans défaillance (TTF, de l'anglais *Time To Failure*) du composant $c_l^{o_i}$. Alors

$$T_{c_l} = \min\left(t_{c_l}(e_{s*}^{o_i}), t_{c_l}(e_{f*}^{o_i})\right) - t_{c_l}(e_{0*}^{o_i}) \tag{5.2}$$

où $\min(t_1, t_2)$ désigne la date du premier événement apparu dans une histoire, le second ne pouvant ainsi plus être réalisé.

Soit $O^{\#} = \left[o_1^{\#}, \ldots o_{N^{\#}}^{\#}\right]$ la liste des objets dont le TTF est intéressant. Par exemple, les TTF de la crue ou du réservoir ne seront pas définis.

Définition 30. Le vecteur des durées de fonctionnement avant défaillance (VTTF, de l'anglais *Vector of Times To Failure*) associé à une simulation s'écrit

$$\left\{\left[\min\left(t_{c_l}(e_{s*}^{o^{\#}}), t_{c_l}(e_{f*}^{o^{\#}})\right) - t_{c_l}(e_{0*}^{o^{\#}})\right]_{o^{\#}\in O^{\#}, l\leq n_c^{o^{\#}}}, \; issue\right\} \tag{5.3}$$

où $issue \in \{-1; 1\}$ désigne l'issue de la simulation, c'est-à-dire la réalisation de l'événement redouté ($issue = 1$) ou non ($issue = -1$) avant la fin de la simulation.

Remarques pratiques

- L'information contenue dans une histoire, c'est-à-dire dans la suite de $n_o \times n_c^o \times e_{N*}$ triplets, est maintenant condensée dans un vecteur composé de $n_{o^{\#}} \times n_c^{o^{\#}}$ durées avec $n_{o^{\#}} \leq n_o$. La manipulation de ce vecteur nécessite d'avoir défini et mémorisé l'indice de chaque composant $c_l^{o^{\#}}$ afin de lui restituer son TTF sans erreur en temps voulu.
- Si la simulation s'achève avant le début du fonctionnement du composant $c_l^{o^{\#}}$, alors son TTF est remplacé par sa durée normale de fonctionnement $t_{c_l}(e_{s*}^{o^{\#}}) - t_{c_l}(e_{0*}^{o^{\#}})$. Il en est de même si la simulation s'achève après le début du fonctionnement du composant $c_l^{o^{\#}}$, mais avant la fin de celui-ci et sans défaillance.

Exemple 3. Soit ([57900, V, **opening**], [60900, V, **open**], [237600, R, **Fin Crue**]) l'histoire retournée par PyCATSHOO pour une seule vanne qui ne connait pas de défaillance. Le VTTF associé est $\{[3000], -1\}$.

Exemple 4. Imaginons un système construit à partir de la même base de connaissances, mais où l'objet Vanne est instanciée trois fois. Toutes les vannes sont supposées s'ouvrir au même instant t_{v_0}.

L'histoire ([t_{v_0}, V1, **opening**], [t_{v_0}, V2, **opening**], [t_{v_0}, V3, **stopped**], [t_{v_1}(stopped), V1, **stopped**], [t_{v_2}(open), V2, **open**], [t_{c_f}, R, **Fin Crue**]) raconte que

- la vanne V1 a commencé à s'ouvrir à l'instant t_{v_0} ;
- la vanne V2 a commencé à s'ouvrir à l'instant t_{v_0} ;
- la vanne V3 a refusé de s'ouvrir à l'instant t_{v_0} : c'est une défaillance à la sollicitation ;
- V1 a connu une défaillance à $t_{v_1}(\texttt{stopped})$: le processus d'ouverture est stoppé ;
- V2 achève son processus d'ouverture sans défaillance à $t_{v_2}(\texttt{open})$;
- la crue s'achève à t_{c_f} sans que l'événement redouté n'ait eu lieu.

Ces informations peuvent être résumées dans le vecteur $\{[T_1, T_2, T_3], -1\}$ où T_i désigne le TTF de la vanne i . Ici, $T_1 = t_{v_1}(\texttt{stopped}) - t_{v_0}$, $T_2 = t_{v_2}(\texttt{open}) - t_{v_0}$ et $T_3 = 0$. L'issue **`Fin Crue`** est représentée par la valeur -1 qui désigne la non réalisation de l'événement redouté.

Exemple 5. Soit l'histoire ([t_1, CCl1, **`ok`**], [t_2, CCl2, **`ok`**], [t_3, CCl2, **`nok`**], [t_4, V1, **`opening`**], [t_5, V1, **`open`**], [t_6, R, **`SSA`**]). Cette histoire raconte que

- le contrôle-commande local CCl1 est actionné à l'instant t_1,
- le contrôle-commande local CCl2 est actionné à l'instant t_2,
- le contrôle-commande local CCl2 tombe en panne à l'instant t_3, avant le début de l'ouverture de la vanne V2 associée ;
- la vanne V1 commence à s'ouvrir à l'instant t_4,
- la vanne V1 achève son ouverture à l'instant t_5,
- l'événement redouté arrive à l'instant t_6, ce qui interrompt la simulation, alors que le CCl1 est encore dans son état de fonctionnement **`ok`**.

Soit $O^{\#} = \{CCl1, CCl2, V1, V2\}$ les composants dont on veut mémoriser le TTF. Le VTTF associé à l'histoire de cet exemple est $\{[T_1, T_2, T_3, T_4], 1\}$ avec $T_1 = d_{op}^{(CCl)}$, $T_2 = t_3 - t_2$, $T_3 = t_5 - t_4$ et $T_4 = d_{ouv}^{(v)}$. $d_{op}^{(CCl)}$ est la durée d'opération d'un contrôle-commande local et $d_{ouv}^{(v)}$ est la durée d'ouverture d'une vanne. L'issue **`SSA`** est représentée par la valeur 1 qui désigne la réalisation de l'événement redouté.

5.1.4 Description des systèmes étudiés

Le but de ce chapitre est l'analyse des histoires et la quantification probabiliste de la fiabilité. De nouveaux indicateurs de fiabilité sont proposés et validés en comparant certains résultats issus de la simulation avec le produit d'un calcul analytique. En vue de ce double objectif compréhension / validation, il s'agit d'expliquer et d'interpréter la méthodologie concernée. Cette démarche est plus aisée si son application se limite dans un premier temps à un système de petite taille.

L'estimation de chaque indicateur sera en général appliquée à un système très simple, adapté à la définition de cet indicateur. Cette application concernera ensuite deux autres systèmes, toujours les mêmes, afin de former deux cas-tests « Fil Rouge » dans ce chapitre. Ces systèmes « Fil Rouge » seront suivis jusque la fin du manuscrit.

Le premier système, nommé FR1, est constitué de 3 composants. Le second, caractérisé par 7 composants, est appelé FR2.

Le système que nous avons étudié et modélisé jusqu'ici est un évacuateur de crues constitué d'environ 75 composants. Cette dimension est trop importante pour servir de support didactique à la présentation de la formalisation et l'exploitation des résultats. Cependant nous avons souhaité garder le concept d'alimentation et de vannes. Aussi, nous n'avons pas modélisé de nouveaux systèmes, mais simplement modifié les paramètres de fiabilité du système réel.

- Les composants du sous-système de contrôle-commande, les composants du sous-système hydromécanique (hormis les vannes), ainsi que l'opérateur, sont supposés parfaits. Cela signifie que leur probabilité de défaillance à la sollicitation est égale à zéro et que la notion de taux de défaillance n'a pas de sens. Le taux de défaillance est remplacé par un instant de défaillance fixe, déterministe : pour chaque histoire simulée, la défaillance du composant concerné est forcée à cet instant (et ne peut avoir lieu avant). Cet instant de défaillance est choisi postérieur à la fin de la crue.
- De même, les paramètres du sous-système d'alimentation sont modifiés pour rendre les composants parfaits, à l'exception de quelques-uns. En effet, seule l'alimentation principale garde une probabilité non nulle de tomber en panne. Le processus de redondance est neutralisé en rendant inopérants les composants de secours. Le système d'alimentation est ainsi agrégé en un unique composant de taux de défaillance $\lambda_{alim} = 7,2 \times 10^{-4}$.
- La structure du sous-système hydromécanique est composée de 4 vannes de surface et 2 vannes de demi-fond. Les paramètres de fiabilité des six vannes de l'évacuateur de crues sont modifiés au cas par cas. En effet, le volume d'eau arrivant dans le réservoir est relativement important, caractéristique d'une crue millénale. Cinq vannes opérationnelles sont nécessaires et suffisantes pour évacuer ce type de crue, à condition que la sixième vanne ne soit pas défaillante avant d'avoir atteint le tiers de son ouverture. Autrement dit, deux vannes défaillantes suffisent pour mener au dépassement du seuil maximal par le niveau de la retenue. Cet événement redouté sera par la suite noté ER. Ce nombre de défaillances sera toutefois à nuancer en fonction des dates de ces défaillances (cette nuance est expliquée dans la section 5.4). Le rôle des vannes se résume ainsi.

1. Système FR1 à 3 composants : alimentation + 2 vannes.
   Seules les vannes 1 et 2 peuvent défaillir de manière aléatoire. Les 4 autres vannes sont considérées comme parfaites ; il est impossible que leur processus d'ouverture soit stoppé avant l'ouverture totale, même si l'alimentation est défaillante.
2. Système FR2 à 7 composants : alimentation + 6 vannes.
   Les 6 vannes peuvent défaillir aléatoirement. Les vannes 1 à 4 sont des vannes de surface. Les vannes 5 et 6 sont des vannes de demi-fond. Les vannes de surface sont caractérisées par une débitance supérieure à celles des vannes de demi-fond .

	FR1 : alimentation + 2 vannes	FR2 : alimentation + 6 vannes
alimentation	$\lambda_{alim} = 7,2 \times 10^{-4}$	$\lambda_{alim} = 7,2 \times 10^{-4}$
vanne 1	$\gamma_{V_1} = 0,03$; $\lambda_{V_1} = 1,2$	$\gamma_{V_1} = 0,03$; $\lambda_{V_1} = 0,8$
vanne 2	$\gamma_{V_2} = 0,03$; $\lambda_{V_2} = 1,2$	$\gamma_{V_2} = 0,03$; $\lambda_{V_2} = 0,8$
vanne 3	$\gamma_{V_3} = 0$; λ_{V_3} non défini	$\gamma_{V_3} = 0,03$; $\lambda_{V_3} = 0,8$
vanne 4	$\gamma_{V_4} = 0$; λ_{V_4} non défini	$\gamma_{V_4} = 0,03$; $\lambda_{V_4} = 0,8$
vanne 5	$\gamma_{V_5} = 0$; λ_{V_5} non défini	; $\lambda_{V_5} = 0,6$
vanne 6	$\gamma_{V_6} = 0$; λ_{V_6} non défini	$\gamma_{V_6} = 0,03$; $\lambda_{V_6} = 0,6$

TABLE *5.1 – Paramètres de fiabilité des composants de chaque système*

Dans les deux systèmes, le processus d'ouverture d'une vanne de surface dure 50 minutes ($d_{ouv}^{(VS)} = 3000$ secondes), celui d'une vanne de demi-fond dure 66 minutes ($d_{ouv}^{(VDF)} = 4000$ secondes). Les défaillances des vannes ne peuvent ainsi avoir lieu que lors du processus d'ouverture. Le tableau 5.1 résume les données de fiabilité de chaque système.

Ces données de fiabilité ont été calibrées pour obtenir un nombre suffisant N d'histoires présentant l'événement redouté, sans obtenir une probabilité d'occurrence de l'événement indésirable trop fantaisiste. Le nombre $N = 350000$ est fixe et ne dépend pas des paramètres ni de la taille du système. Il est possible de réaliser une étude de convergence afin d'optimiser le nombre de simulations nécessaires à l'estimation de la probabilité d'occurrence de l'ER. Cette étude repose sur la largeur de l'intervalle de confiance. Cependant, les indicateurs proposés dans le chapitre 6 sont à notre connaissance ([Duflot, 2007], [Do Van, 2008]) incompatibles avec le calcul d'une variance et d'un intervalle de confiance. N a donc été choisi arbitrairement pour d'obtenir un échantillon d'histoires assez étoffé pour distinguer les causes de l'ER.

5.2 Probabilité d'occurrence de l'événement redouté

Un indicateur à estimer est la probabilité d'occurrence de l'événement redouté et son évolution dans le temps. Dans un premier temps, cette mesure est calculée de manière analytique. Pour cela, un cas-test très simple est considéré, composé du réservoir, de la crue et d'une seule vanne. Ce cas-test a déjà utilisé pour expliquer l'élaboration d'une base de connaissances dans la section 4.2.1.

5.2.1 Calcul analytique

La probabilité d'occurrence de l'événement redouté à l'instant t est donnée par $P_{ER}(t) = \mathbb{P}(h(t) > h_{max})$ où la fonction h désigne le niveau dans la retenue et h_{max} le seuil à ne pas dépasser. L'estimation de P_{ER} nécessite donc celle du niveau h mais aussi le calcul de l'instant d'atteinte du seuil de sûreté en fonction du temps de défaillance de la vanne. Pour éviter des calculs analytiques trop complexes, seuls les débits entrants en forme d'échelon et les débits sortants constants seront considérés. Ces débits sont introduits dans la section 4.2.1.1.1.

5.2.1.1 Évolution du niveau dans la retenue

Le débit entrant correspondant à une crue en forme d'échelon s'écrit

$$q_{ent}(t) = I_c \times \mathbf{1}_{[t_{c_0}; t_{c_f}]}(t) \tag{5.4}$$

où I_c est une constante représentant l'intensité de la crue, $\mathbf{1}$ la fonction indicatrice et t_{c_0} et t_{c_f} les instants respectifs de début et de fin de la crue.

Soit t_{v_0} l'instant du début de l'ouverture de la vanne v et d_{ouv} la durée du processus d'ouverture.

La débitance de la vanne v est nulle avant le début de l'ouverture de la vanne, constante après l'atteinte de l'ouverture maximale, et linéaire entre ces deux instants :

$$q_{sor}(t) = \begin{cases} 0 & \text{si } t \leq t_{v_0} \\ \frac{(t - t_{v_0}) \times q_{max}}{d_{ouv}} & \text{si } t_{v_0} \leq t \leq t_{v_0} + \min(u,\, d_{ouv}) \\ \min(q_u,\, q_{max}) & \text{sinon} \end{cases} \tag{5.5}$$

où q_{max} est le débit sortant maximal théorique de la vanne v, correspondant à son ouverture entière. Si la vanne tombe en panne au bout d'une durée u lors de son ouverture ($u \leq d_{ouv}$), alors $q_u = q_{sor}(t_{v_0} + u)$ est le débit sortant associé à cet instant.

L'évolution du niveau dans le réservoir en fonction du temps est donnée par l'équation différentielle

$$\frac{dh}{dt}(t) = q_{ent}(t) - q_{sor}(t). \tag{5.6}$$

L'expression analytique du niveau dans le réservoir a été calculée en résolvant cette équation différentielle, en considérant le cas où les événements interviennent dans l'ordre « instant de début de la crue, puis instant de l'amorce de l'ouverture de la vanne, puis atteinte de l'ouverture maximale, puis fin de la crue » c'est-à-dire $t_{c_0} \leq t_{v_0} \leq t_{v_f} \leq t_{c_f}$, où $t_{v_f} = t_{v_0} + d_{ouv}$ est l'instant de l'ouverture maximale de la vanne. h_0 est le niveau initial et S la superficie de la retenue.

Si $t \leq t_{c_0}$ alors

$$h(t) = h_0. \tag{5.7}$$

Si $t_{c_0} \leq t \leq t_{v_0}$ alors

$$h(t) = h_0 + \frac{I_c}{S}(t - t_{c_0}). \tag{5.8}$$

Si $t_{v_0} \leq t \leq t_{v_f}$ alors

$$h(t) = h_0 + \frac{I_c}{S}(t - t_{c_0}) - \frac{q_{max}}{2 \times d_{ouv} \times S}(t - t_{v_0})^2. \tag{5.9}$$

Si $t_{v_f} \leq t \leq t_{c_f}$ alors $h_{v_f} = h(t_{v_f})$ et

$$h(t) = h_{v_f} + \frac{I_c}{S}(t - t_{v_f}) - \frac{q_{max}(t - t_{v_f})}{S}. \tag{5.10}$$

Si $t_{c_f} \leq t$ alors

$$h(t) = h_{v_f} + \frac{I_c}{S}(t_{c_f} - t_{v_f}) - \frac{q_{max}(t - t_{v_f})}{S}. \tag{5.11}$$

Dans le cas où la vanne tombe en panne au bout d'une durée $u \leq d_{ouv}$, t_{v_f} est remplacé par $t_{v_0} + u$ et h_{v_f} par $h(t_{v_0} + u)$ et on obtient :

Si $t \leq t_{c_0}$ alors

$$h(t) = h_0.$$

Si $t_{c_0} \leq t \leq t_{v_0}$ alors

$$h(t) = h_0 + \frac{I_c}{S}(t - t_{c_0}).$$

Si $t_{v_0} \leq t \leq t_{v_0} + u$ alors

$$h(t) = h_0 + \frac{I_c}{S}(t - t_{c_0}) - \frac{q_{max}}{2 \times d_{ouv} \times S}(t - t_{v_0})^2. \tag{5.12}$$

Si $t_{v_0} + u \leq t \leq t_{c_f}$ alors

$$h(t) = h(t_{v_0} + u) + \frac{I_c}{S} \times (t - (t_{v_0} + u)) - \frac{q_{sor}(t_{v_0} + u)}{S} \times (t - (t_{v_0} + u)). \tag{5.13}$$

Si $t_{c_f} \leq t$ alors

$$h(t) = h(t_{v_0} + u) + \frac{I_c}{S} \times \Big(t_{c_f} - (t_{v_0} + u)\Big) - \frac{q_{sor}(t_{v_0} + u)}{S} \times (t - (t_{v_0} + u)). \tag{5.14}$$

5.2.1.2 Instant d'atteinte du seuil de sûreté en fonction du temps de défaillance

L'atteinte du seuil de sûreté par le niveau, qui constitue l'événement redouté (ER), a lieu avec un certain retard par rapport à l'instant de défaillance. Ce délai est lié au dimensionnement du réservoir : atteindre le niveau h_{max} à partir du niveau h_0 n'est pas immédiat.

Pour chaque instant de panne $u \geq 0$, l'instant d'occurrence de l'ER s'écrit

$$t_{ER}(u) = \frac{I_c \times t_{c_0} - (h_0 + h_{max})\, S - \frac{q_{max}}{d_{ouv}} t_{v_0} u - \frac{q_{max}}{2 \times d_{ouv}} u^2}{I_c - \frac{q_{max}}{d_{ouv}} u}. \tag{5.15}$$

La preuve de cette expression figure en annexe B.

On obtient ainsi la première date possible pour l'ER $t_{ER_0} = t_{ER}(0)$.

Les débits entrant et sortant ont une forme d'échelon, aussi la dernière date possible pour l'ER correspond-elle à l'instant t_{c_f} de fin de la crue.

5.2.1.3 Temps de défaillance en fonction de l'instant d'atteinte du seuil de sûreté

À partir de l'écriture de la fonction $t_{ER}(u)$, la fonction inverse t_{ER}^{-1} détermine l'instant de panne u en fonction de l'instant de l'ER t_{ER}. Notons u_{ER} cette fonction :

$$u_{ER}(t) = \frac{q_{max}}{d_{ouv}} (t - t_{v_0}) \left[1 - \sqrt{1 + 2 \frac{q_{max}}{d_{ouv}} \frac{(h_{max} - h_0) S + I_c (t_{c_0} - t)}{(t - t_{v_0})^2}} \right] \tag{5.16}$$

où S désigne la superficie du réservoir.

5.2.1.4 Expression de $P_{ER}(t)$

$$P_{ER}(t) = \begin{cases} 0 & \text{si } t \leq t_{ER_0} \\ p_{soll} + (1 - p_{soll}) \left(1 - e^{-\lambda u_{ER}(t)} \right) & \text{sinon} \end{cases} \tag{5.17}$$

La preuve de cette expression figure en annexe B.

5.2.2 Estimation par simulation de Monte Carlo sur le modèle ASH

5.2.2.1 Évolution du niveau

L'outil PyCATSHOO permet d'estimer ces grandeurs à partir des résultats de la simulation de Monte Carlo. Le module de suivi (*monitoring*) des variables déterministes continues a été utilisé. Ce module d'observation est capable de retourner l'évolution moyenne du niveau sur un nombre donné de simulations. Nous souhaitons obtenir les courbes d'évolution du niveau pour quatre situations afin de les comparer aux résultats analytiques. Ces situations sont la panne à sollicitation, la défaillance en fonctionnement au bout de $u = 1000$ secondes, la défaillance en fonctionnement au bout de $u = 2500$ secondes et l'ouverture totale de la vanne sans défaillance. Pour cela, il suffit de simuler pour chaque cas une seule histoire en forçant l'instant de défaillance à la date voulue. Le *monitoring* est ensuite effectué sur cette histoire. Ces quatre simulations sont réalisées indépendamment de l'estimation des grandeurs suivantes, en forçant les défaillances à la date u.

5.2.2.2 Dépendance de l'instant de panne et de l'instant d'atteinte du seuil de sûreté

La manipulation des résultats pour associer les instants de panne et les instants de l'événement redouté est relativement simple. Cela ne concerne que les simulations dont l'issue est l'ER. L'instant d'occurrence de cet événement est obtenu par les histoires. La durée de fonctionnement de la vanne avant sa défaillance provient du vecteur de durées de fonctionnement avant défaillance (VTTF).

5.2.2.3 Évolution de la probabilité P_{ER}

L'état **`SSA (seuil de sûreté atteint)`** est créé dans la base de connaissances de l'outil PyCATSHOO. Cet état est activé lorsque le niveau dépasse le seuil surveillé. Le *monitoring* est exercé sur l'activation ou non de l'état **`SSA`**. Cela permet de stocker l'information relative à l'occurrence de l'événement redouté et sa date, sans mémoriser les trajectoires prises par la variable continue.

Pour chaque histoire, PyCATSHOO connaît donc l'occurrence ou non de l'événement redouté, et sa date le cas échéant. L'outil est ensuite capable de compiler ces données pour retourner la probabilité d'occurrence pour une liste d'instants t :

$$p_{occ}(t) = \frac{\text{nombre d'histoires pour lesquelles l'état } \texttt{SSA} \text{ est activé avant } t}{\text{nombre d'histoires simulées}}. \tag{5.18}$$

D'après [Peyre, 2012], le nombre de simulations doit être supérieur à $N = (2\alpha\sigma/l)^2$, où $\alpha = 1,96$ est le quantile d'ordre 2,5% de la loi normale, σ l'écart-type et l la largeur désirée de l'intervalle de confiance. Avec $p_{occ}(t_{c_f})$ de l'ordre de 0,4, σ est de l'ordre

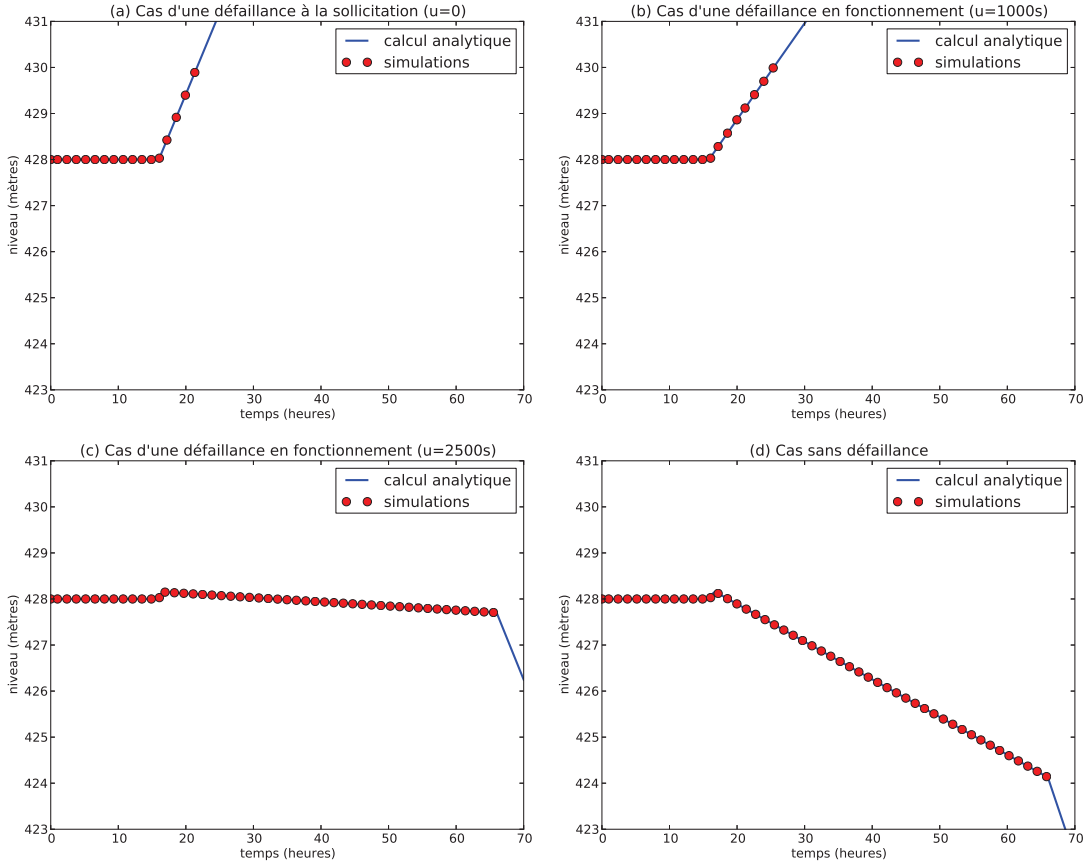

FIGURE 5.3 – Crue en forme d'échelon et débitance constante : évolution du niveau dans le réservoir

de 0,5 donc si l'on souhaite un IC d'une largeur $l = 10^{-2}$, environ 39000 simulations sont nécessaires. Cependant, d'autres indicateurs pour lesquels il n'est possible d'exprimer l'IC vont être estimés à partir de ces mêmes histoires, aussi avons-nous choisi de simuler 350000 histoires afin de disposer d'un grand nombre de données. Ces 350000 simulations nécessitent une vingtaine d'heures de calcul pour 7 processus. Dépasser cette limite maximale requiert une place mémoire non disponible lors du regroupement et du stockage des résultats de ces 7 processeurs.

5.2.3 Comparaison des résultats analytiques et du produit des simulations

5.2.3.1 Évolution du niveau

La figure 5.3 présente l'évolution du niveau dans la retenue pour quatre instants de panne différents. A chaque fois, le résultat du calcul analytique et les valeurs retournées

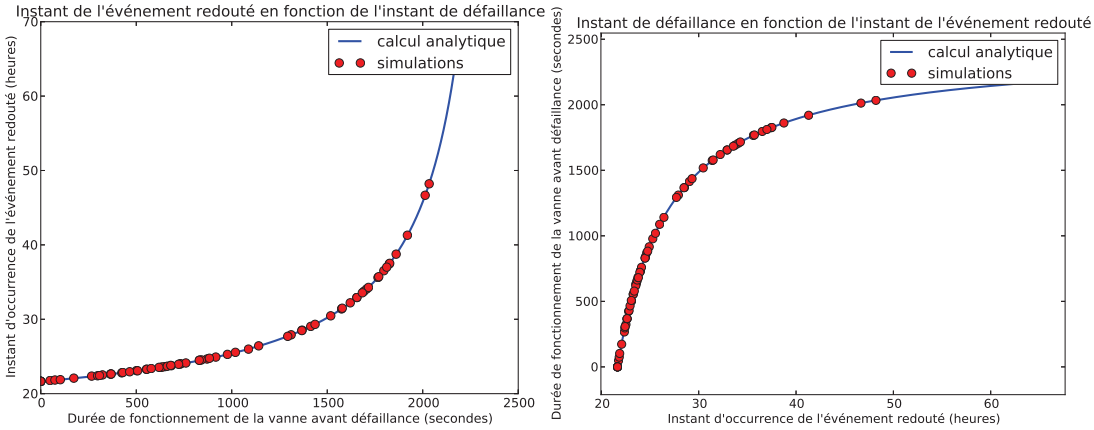

FIGURE 5.4 – Dépendance de l'instant de défaillance de la vanne et de l'instant de l'événement redouté

par PyCATSHOO sont identiques. Cela permet de valider l'outil de calcul de l'évolution de la variable déterministe continue de PyCATSHOO.

La courbe correspondant aux simulations est toujours plus « courte » que celle du calcul analytique. En effet, aucune borne supérieure ne va stopper le calcul analytique. Inversement, l'arrêt des simulations (atteinte du niveau h_{max} ou de la date de fin de la crue t_{c_f}) entraîne automatiquement l'arrêt de l'évaluation du niveau dans la retenue.

5.2.3.2 Dépendance de l'instant de panne et de l'instant de l'événement redouté

Les figures 5.4 illustrent la dépendance entre l'instant de défaillance de la vanne et celui de l'ER. Les résultats sont obtenus par calcul analytique et *via* les histoires de PyCATSHOO. Le bon accord entre les résultats valide à la fois l'exactitude du calcul théorique et la cohérence de l'outil de simulation.

5.2.3.3 Évolution de la probabilité P_{ER}

La figure 5.5 présente l'évolution de la probabilité d'occurrence de l'événement redouté. Ici aussi, l'expression analytique de l'équation 5.17 et les sorties de PyCATSHOO sont en accord.

Le pas de temps entre les cercles rouges a été fixé arbitrairement pour améliorer la visibilité de la figure. Il ne dépend pas du nombre de simulations. L'outil de simulation retourne l'évolution de la probabilité d'occurrence pour une liste d'instants aussi fine que désirée. Dans ce cas, les deux courbes se superposent.

La première date possible pour l'ER (vers 22 heures) correspond à un saut vertical. Ce saut illustre les défaillances à la sollicitation, qui ont toutes lieu au même instant.

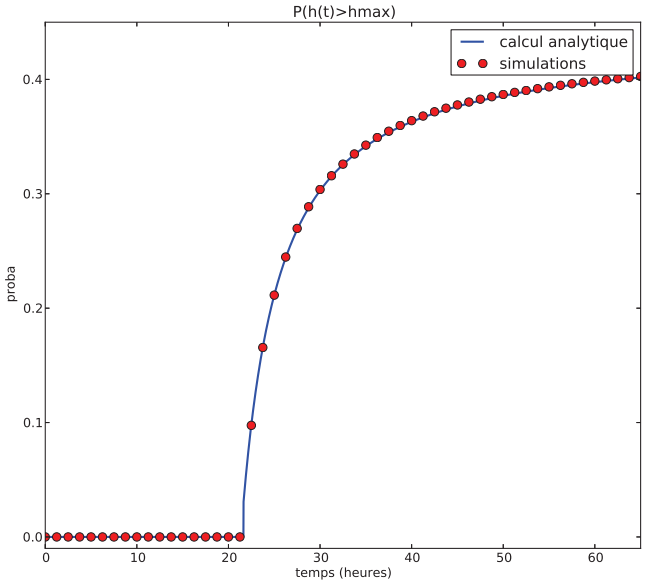

FIGURE 5.5 – Évolution de la probabilité d'occurrence de l'événement redouté

La courbe prend l'allure caractéristique de la fonction exponentielle de paramètre λ. La hauteur de ce saut a pour valeur $p_{soll} = 0,03$.

5.2.4 Vers un cas-test plus proche de la réalité : allure et interprétation de courbes de niveau h et de P_{ER}

Cette section a pour objectif de faire évoluer notre cas-test très simple vers un système plus réaliste. Le calcul analytique devient trop complexe, aussi seuls les résultats des simulations sont-ils utilisés. Les grandeurs h et P_{ER} sont donc obtenues « sans filet ».

Les différentes étapes sont la prise en compte d'un hydrogramme de crue, celle d'une débitance dépendant du niveau, le critère « laminer la crue », le changement de loi de probabilité pour la durée de fonctionnement avant défaillance de la vanne, puis l'ajout d'une deuxième vanne. Finalement, les deux systèmes « fil rouge » exposés dans la section 5.1.4 sont modélisés.

5.2.4.1 Vers une modélisation réaliste des débits entrant et sortant

5.2.4.1.1 Influence de la levée des hypothèses simplificatrices sur l'évolution du niveau h

La figure 5.6 illustre l'impact des hypothèses simplificatrices sur l'évolution du niveau, pour quatre scénarios de défaillance. Ces scénarios de défaillance sont (a) défaillance à la sollicitation, (b) défaillance en fonctionnement au bout de 1000 secondes, (c) défaillance

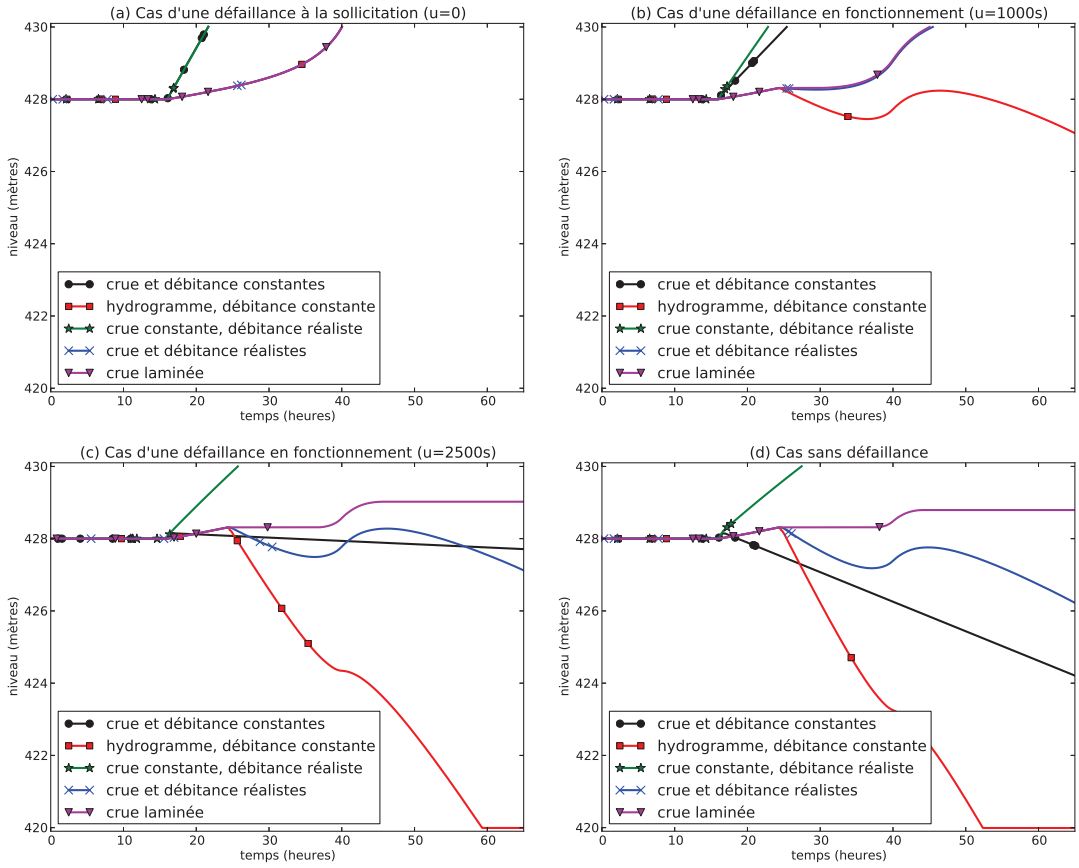

FIGURE 5.6 – Influence de la levée des hypothèses simplificatrices sur l'évolution du niveau dans le réservoir, pour quatre scénarios de défaillance

en fonctionnement au bout de 2500 secondes, (d) ouverture complète de la vanne sans défaillance.

1. La courbe en noir avec des cercles « crue et débitance constantes » rappelle l'évolution du niveau tel qu'il est calculé avec les hypothèses les plus simples. La crue est en forme d'échelon et la débitance de la vanne est constante. Si la débitance dépend de la hauteur d'ouverture de la vanne, elle ne dépend pas du niveau. Une crue en forme d'échelon provoque une brusque montée des eaux si elle n'est pas évacuée efficacement.
2. La courbe en rouge avec des carrés « hydrogramme, débitance constante » montre l'influence d'un débit entrant réaliste sur l'évolution du niveau. Même si la vanne ne s'ouvre pas, la montée des eaux est plus progressive. Si la débitance des vannes était réellement indépendante du niveau, celles-ci seraient surdimensionnées pour évacuer une crue modélisée par cet hydrogramme.
3. La courbe en vert avec des étoiles « crue constante, débitance réaliste » revient à une crue en forme d'échelon mais introduit la modélisation réaliste de la dé-

bitance des vannes. Dans ce cas, la débitance est une fonction du niveau d'eau. Cela signifie que, une fois la vanne entièrement ouverte, la débitance est proportionnelle au niveau dans la retenue. Ainsi modélisée, la vanne paraît moins apte à l'évacuation d'une crue brusque comme c'est le cas pour une crue en forme d'échelon. Dans ce cas complètement fictif, l'événement redouté arrive même si la vanne ne connaît pas de défaillance. Cependant une défaillance plus tardive retarde l'instant d'occurrence de l'événement redouté.

4. La crue en bleu avec des croix « crue et débitance réalistes » regroupe les contraintes des points 2 et 3. La crue est modélisée par un hydrogramme et la débitance de la vanne dépend du niveau. Les variations du niveau sont lissées, ce qui illustre le bon dimensionnement des vannes pour répondre à la crue.
5. La crue en violet avec des triangles introduit l'objectif laminer la crue. Le débit sortant ne doit plus être supérieur au débit entrant. Aussi le niveau devient-il constant dès que l'ouverture des vannes est suffisante pour évacuer la crue. Cette modification dans l'équation différentielle dispense temporairement du calcul de l'ouverture optimale pour chacune des vannes.

5.2.4.1.2 Influence de la levée des hypothèses simplificatrices sur l'évolution de la probabilité d'occurrence P_{ER}

La figure 5.7 illustre l'évolution de la probabilité d'occurrence de l'événement redouté lorsque la modélisation de l'évolution du niveau est réaliste. Cela signifie que la crue est modélisée par un hydrogramme, que la débitance de la vanne dépend du niveau et que le débit sortant ne doit pas être supérieur au débit entrant, pour respecter l'objectif « laminer la crue ». Ainsi, la courbe bleue en trait continu de la figure 5.7 correspond aux courbes violettes avec des triangles de la figure 5.6. Elle est comparée à l'évolution du niveau dans le cas d'une modélisation simplifiée : la courbe en tirets gris de la figure 5.7 correspond aux courbes noires avec des cercles de la figure 5.6.

Cette comparaison implique deux constats.

1. Les deux courbes sont décalées. En effet, la probabilité d'occurrence reste nulle beaucoup plus longtemps lorsque la modélisation est réaliste. Cela s'explique par le fait que la prise en compte de l'hydrogramme « lisse » l'évolution du niveau : la crue arrive progressivement, ce qui retarde l'atteinte du seuil maximal par le niveau. Ce délai d'une vingtaine d'heures est visible sur les courbes (a) et (b) de la figure 5.6.
2. Même à la fin de la crue, la probabilité de l'événement redouté reste inférieure lorsque la modélisation est réaliste. En effet, pour certains instants de panne de la vanne, l'ouverture correspondante est suffisante pour évacuer une crue modélisée par un hydrogramme, mais pas une crue en forme d'échelon. L'écart entre les courbes en trait continu bleu et en tirets gris s'explique donc par la part de la probabilité d'occurrence due à la réalisation de ces instants de panne.

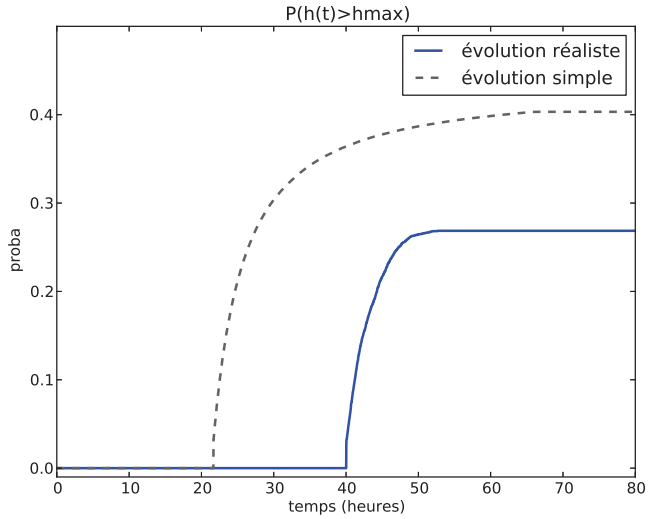

FIGURE 5.7 – Influence de la levée des hypothèses simplificatrices sur la probabilité d'occurrence de l'événement redouté

Modélisations simple et réaliste donnent finalement des résultats d'allure similaire en termes de probabilité d'occurrence. Dans la suite des travaux, nous choisissons de garder les hypothèses simplificatrices. Premièrement, parce que nos deux systèmes « Fil Rouge » sont inspirés d'un évacuateur réel. Au début de nos travaux, les données disponibles pour ce barrage ne permettaient qu'une modélisation simplifiée. Deuxièmement, cela permet de simuler plus souvent l'événement indésirable et de disposer de davantage d'histoires à exploiter. Troisièmement, le temps de simulation est plus court. Enfin, il est tout à fait possible d'obtenir des probabilités d'occurrences superposées en modifiant légèrement les paramètres de données de fiabilité et l'instant de début de la crue.

5.2.4.2 Vers des lois de probabilités variées : introduction de la loi de Weibull

Lorsque la durée de fonctionnement avant défaillance de la vanne suit une loi exponentielle, la probabilité d'occurrence de l'événement redouté à l'instant t s'écrit

$$P_{ER}^{(expo)}(t) = \begin{cases} 0 & \text{si } t \leq t_{ER_0} \\ p_{soll} + (1 - p_{soll}) \left(1 - e^{-\lambda u_{ER}(t)}\right) & \text{sinon} \end{cases} \tag{5.19}$$

où t_{ER_0} est la première date possible pour l'ER et $u_{ER}(t)$ désigne l'instant de panne associé à l'instant de l'événement redouté t, donné par l'équation (5.16).

Le calcul est similaire lorsque cette durée suit une loi de Weibull de paramètres (λ, β). Ainsi,

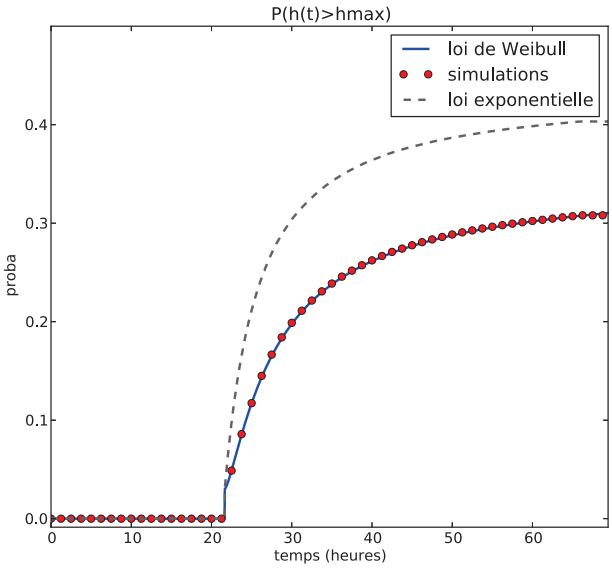

FIGURE 5.8 – Influence de la loi de Weibull sur l'évolution de la probabilité d'occurrence de l'événement redouté

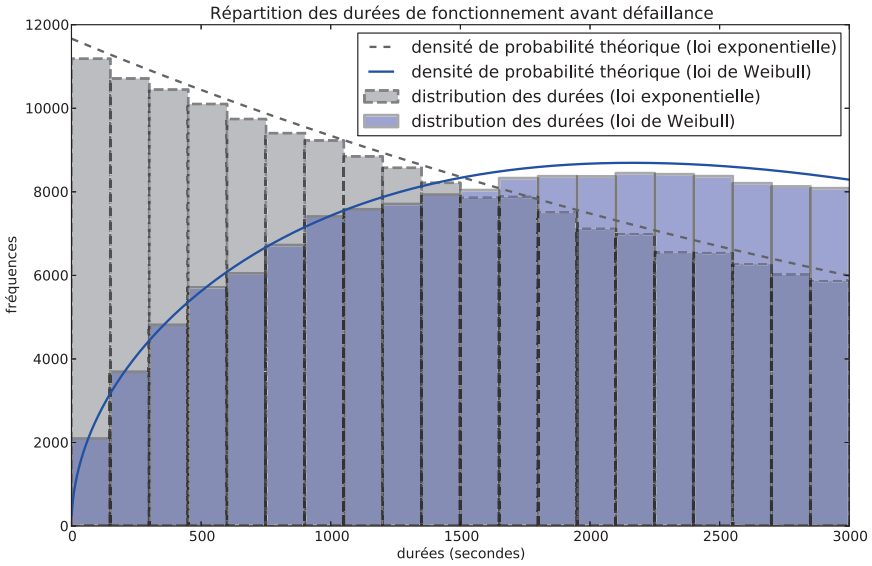

FIGURE 5.9 – Répartition des instants de défaillance pour la loi de Weibull et la loi exponentielle

$$P_{ER}^{(W)}(t) = \begin{cases} 0 & \text{si } t \leq t_{ER_0} \\ p_{soll} + (1 - p_{soll}) \left(1 - e^{-(\lambda u_{ER}(t))^{\beta}}\right) & \text{sinon} \end{cases}. \tag{5.20}$$

La figure 5.8 montre l'allure similaire des probabilités d'occurrence obtenues lorsque les durées suivent la loi exponentielle ou la loi de Weibull, avec le même λ et $\beta = 1,5$. Cette fois, il n'y a pas de décalage du premier instant possible pour l'événement redouté. On retrouve à cet instant le saut d'une hauteur équivalent à p_{soll}. Le calcul analytique et les résultats des simulations correspondent.

La probabilité d'occurrence de l'événement redouté lorsque les durées suivent la loi de Weibull est inférieure à celle calculée lorsque les durées suivent la loi exponentielle. Cela s'explique naturellement par la répartition des durées en fonction de la loi de probabilité utilisée. La loi de Weibull est à l'origine d'un glissement des données vers des valeurs plus importantes, ce qui est illustré par les histogrammes de la figure 5.9. La distribution des durées pour la loi de Weibull est telle que la plupart des défaillances arriveraient après l'ouverture totale des vannes. Or ce sont les plus petites durées qui sont à l'origine de l'événement redouté, d'où l'écart entre les deux courbes.

5.2.4.3 Vers un système de taille réaliste

5.2.4.3.1 Ajout d'une deuxième vanne

Notons « situation 1 » le réservoir pour lequel la probabilité d'occurrence de l'événement redouté a été estimée jusqu'ici. Ce réservoir simple ne possède qu'une seule vanne. L'objectif de ce paragraphe est de mesurer l'impact de l'introduction d'une deuxième vanne sur la probabilité d'occurrence de l'événement redouté, par rapport à la situation 1. Un deuxième objet de classe Vanne est donc instancié. On nomme V1 la première vanne et V2 la seconde.

V1 et V2 ont les mêmes données de fiabilité, qui sont identiques à la situation 1. En revanche, la débitance des deux vannes est divisée par deux. Il faut maintenant les deux vannes pour évacuer ce que débitait une seule vanne dans la situation 1. Soit $h_1^{(u)}$ le niveau dans la situation 1, sachant que la vanne a une défaillance au bout de la durée u. Soit $h_2^{(u)}$ le niveau dans la situation 2, sachant que les deux vannes ont une défaillance simultanée au bout de la durée u. Quelque soit l'instant t de la crue, $h_2^{(u)}(t) = h_1^{(u)}(t)$. Les défaillances sont définies par des lois de probabilité continues, aussi cette égalité n'est possible que si $u = 0$ (les deux vannes sont défaillantes à la sollicitation, avec la probabilité p_{soll}^2) ou si aucune des deux vannes n'est défaillante.

Un premier saut, situé au premier instant possible pour l'événement redouté, a pour hauteur $p_{soll}^2 = 9 \times 10^{-4}$ et n'est pas perceptible à l'oeil nu sur la figure 5.10. Il correspond à une double défaillance à la sollicitation.

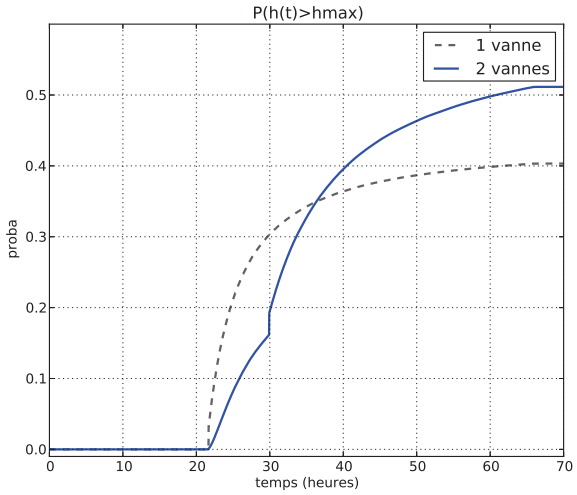

FIGURE 5.10 – Influence de l'introduction d'une seconde vanne sur l'évolution de la probabilité d'occurrence de l'événement redouté

En revanche un second saut, au bout de trente heures, est bien visible. De l'ordre de p_{soll}, il représente toutes les histoires où une vanne est défaillante à la sollicitation alors que l'autre finit son ouverture sans défaillance.

La probabilité d'occurrence de l'événement redouté augmente avec l'introduction de la seconde vanne. L'explication vient de la division de la débitance par deux, sans changer les données de fiabilité. La probabilité qu'une vanne sur les deux soit défaillante avant la durée u est supérieure à la probabilité que la seule vanne de la situation 1 soit défaillante avant u.

5.2.4.3.2 Réservoir muni d'une alimentation et de plusieurs vannes

Les systèmes considérés à présent sont les systèmes « fil rouge » de l'exploitation des résultats. Il s'agit des systèmes FR1 et FR2 décrits dans la section 5.1.4. FR1 possède trois composants : une alimentation et deux vannes. FR2 possède 7 composants : une alimentation et 6 vannes.

La figure 5.11 représente l'évolution de la probabilité d'occurrence de l'événement redouté pour les systèmes FR1 et FR2. L'analyse de cette figure permet l'observation de plusieurs phénomènes.

- Le premier instant possible pour l'événement redouté est retardé par rapport à la figure 5.10. Cela est dû à une différence dans la modélisation de la crue, dont le débit maximal est légèrement inférieur pour les systèmes « fil rouge ».
- Pour les deux systèmes, un saut vers 35 heures correspond à la défaillance de l'alimentation avant l'ouverture des vannes ou à une double défaillance à la sollicitation des vannes (5,3 vannes sont nécessaires pour évacuer la crue).

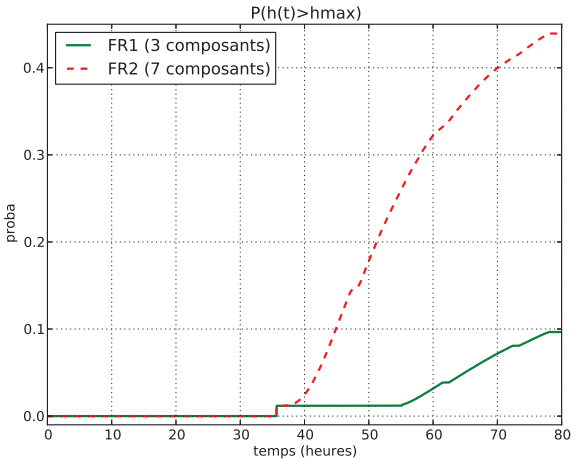

FIGURE 5.11 – Évolution de la probabilité d'occurrence de l'événement redouté pour les deux systèmes « Fil Rouge »

- La défaillance à la sollicitation d'une seule vanne ne suffit pas pour provoquer l'événement redouté. Il n'y a donc pas d'autres sauts sur la figure.
- Des paliers sont visibles, surtout pour le système à trois composants. Ils n'ont pas d'explication en rapport avec les défaillances des vannes. La courbe verte n'est pas constante entre 40 et 50 heures mais très légèrement croissante. Les histoires qui correspondent à cette période d'occurrence de l'événement redouté sont simplement peu nombreuses. Des simulations supplémentaires lisseraient ces paliers.
- Les débitances ne changent pas d'un système à l'autre. FR1 est constitué de deux vannes faillibles et quatre vannes parfaites. FR2 est constitué de six vannes faillibles. C'est ce qui explique l'importante probabilité d'occurrence de l'événement redouté pour FR2.

La probabilité d'occurrence de cet événement a été évaluée dans cette section à partir de l'estimation du niveau dans le réservoir et de la dépendance entre instant de panne et instant de l'événement redouté. Dans un cas où les équations déterministes sont simplifiées, les résultats provenant des simulations concordent avec ceux d'un calcul analytique. Cela permet de valider le fonctionnement de PyCATSHOO pour un réservoir simple. Après cette validation, la simulation de systèmes plus réalistes est rendue possible. Les applications présentes dans ce manuscrit se limiteront aux deux systèmes « Fil Rouge », relativement simple, pour faciliter l'interprétation des résultats.

5.3 Coupes équivalentes prépondérantes

Après avoir estimé la probabilité d'occurrence de l'événement redouté, la méthode GASPART a pour objectif de cibler quels composants (ou quels groupes de composants) contribuent le plus dans la réalisation de cet événement. A partir de la description du système implémenté sous KB3 et de la base de connaissances sous-jacente, l'outil GASPART est capable de construire l'arbre d'événements associé. Le module FigSeq explore chaque branche de l'arbre et en calcule la probabilité d'occurrence. Une branche, c'est-à-dire la succession des événements qui la constituent, est appelée « séquence » dans GASPART. Les séquences dont la probabilité d'occurrence figure parmi les plus importantes sont appelées séquences prépondérantes.

Certaines séquences gagnent à être rassemblées. C'est le cas des séquences qui concernent toutes le même type de composant. Le poids de ces séquences, agrégées en une séquence équivalente, fournit alors une information plus lisible que la liste des poids de toutes les séquences prises individuellement. L'outil GASPART dispose d'un module de post-traitement des séquences dont le but est de proposer une aide au regroupement des séquences.

Avant même d'étendre les possibilités au traitement des informations temporelles, un des objectifs de ces travaux est de faire en sorte que PyCATSHOO retourne le même type d'indicateur. La donnée temporelle contenue dans les histoires ne sera pas utilisée dans cette section. Quitte à perdre cette information, l'ordre des événements ne sera pas exploité non plus. Aussi parlerons-nous de coupes prépondérantes et non de séquences.

5.3.1 Méthodologie

Soit $O^{\#}$ l'ensemble des objets dont la défaillance constitue un événement intéressant. $O^{\#}$ exclut généralement les objets Crue et Réservoir. Notons $S^{\#}$ l'ensemble des composants-instances des classes de $O^{\#}$ et $s_{\#}$ le cardinal de $S^{\#}$. Chaque composant $i \in S^{\#}$ est associé à sa durée normale de fonctionnement $d_{op}^{(i)}$.

Soit $D = \{d_k\}_{k \leq N}$ l'ensemble des VTTF obtenus à partir des N simulations, où $d_k = \{v_k, i_k\}$ est le VTTF associé à la simulation de l'histoire k. $v_k = [T_1^k, ..., T_{s_{\#}}^k]$ et $i_k \in \{-1; 1\}$ désigne l'issue de la simulation k, c'est-à-dire la réalisation de l'événement redouté ($i_k = 1$) ou non ($i_k = -1$) avant la fin de la simulation.

En sûreté de fonctionnement, une coupe est une combinaison d'événements entrainant l'événement redouté. Les histoires concernées par les coupes prépondérantes sont celles dont l'issue est l'événement redouté. Soit $H^+ = \{k \leq N / i_k = 1\}$ l'ensemble des histoires dont l'issue est l'événement redouté et $V^+ = \{v_k \in d_k / d_k \in D \text{ et } i_k = 1\}$ l'ensemble des VTTF associés.

Chaque vecteur $v_k = \left[T_1^k, ... T_{s_{\#}}^k\right] \in V^+$ est comparé au vecteur des durées normales de fonctionnement $\left[d_{op}^{(1)}, ..., d_{op}^{(s_{\#})}\right]$.

Définition 31. La coupe c_k associée au VTTF $\left[T_1^k, ... T_{s_\#}^k\right]$ est constituée des composants i dont le TTF T_i est strictement inférieur à la durée normale $d_{op}^{(i)}$:

$$c_k = \left\{ i \in S^\# / T_i^k < d_{op}^{(i)} \right\}. \tag{5.21}$$

Remarque 2. Cette définition justifie l'attribution de la valeur $d_{op}^{(i)}$ à la place de T_i^k lorsque la simulation k s'achève avant la fin du fonctionnement du composant i.

Chaque simulation k est donc associée à une coupe c_k. Une histoire sans aucune défaillance est associée à l'ensemble vide $\emptyset$. Plusieurs histoires peuvent faire apparaître une même coupe. Soit $N_\neq$ le nombre de coupes distinctes, et $C_\neq = \left\{c_1^\neq, ..., c_{N_\neq}^\neq\right\}$ l'ensemble de ces coupes. La coupe $c_j^\neq$ est associée à l'ensemble des histoires qui lui correspondent $H_j = \left\{k \leq N / c_k = c_j^\neq\right\}$. $p_j = \frac{Card(H_j)}{Card(H^+)}$ est le poids de la coupe $c_j^\neq$, où $Card(E)$ désigne le cardinal de l'ensemble E. p_j correspond à la fréquence d'observation relative à la coupe $c_j^\neq$ sur l'ensemble des histoires simulées d'issue ER,

Exemple 6. L'histoire ([t_{v_0}, V1, **opening**], [t_{v_0}, V2, **opening**], [t_{v_0}, V3, **stopped**], [t_{v_1}(stopped), V1, **stopped**], [t_{v_2}(open), V2, **open**], [t_{c_f}, R, **SSA**]) est associée au VTTF $\{[T_1, T_2, T_3], 1\}$. T_i désigne le TTF de la vanne i avec $T_1 = t_{v_1}(\texttt{stopped}) - t_{v_0}$, $T_2 = t_{v_2}(\texttt{open}) - t_{v_0}$ et $T_3 = 0$. La coupe liée à cette histoire est {V1, V3}.

Exemple 7. Soit $S^\# = \{CCl1, CCl2, V1, V2\}$. L'histoire ([$t_1$, CCl1, **ok**], [$t_2$, CCl2, **ok**],
[t_3, CCl2, **nok**], [t_4, V1, **opening**], [t_5, V1, **open**], [t_6, R, **SSA**]) est associée au VTTF $\{[T_1, T_2, T_3, T_4], 1\}$ avec $T_1 = d_{op}^{(CCl)}$, $T_2 = t_3 - t_2$, $T_3 = t_5 - t_4 = d_{ouv}^{(v)}$ et $T_4 = d_{ouv}^{(v)}$. La coupe liée à cette histoire est {CCl2}.

Regroupement des coupes. Comme dans GASPART, les coupes peuvent être rassemblées en coupes équivalentes. Dans ce cas, l'analyste doit prédéfinir ces regroupements. Le regroupement le plus naturel est un rassemblement des composants par type d'objet.

Soit un système constitué des composants de $S^\#$. Un composant $s^o \in S^\#$ est une instance d'un objet $o \in O^\#$ et n_o est le nombre d'instances de l'objet o. $S^\#$ est une partition des composants répartis en classes : $S^\# = \cup_{o \in O^\#} (s_i^o)_{i \leq n_o}$.

Soit $c_\neq$ une coupe contenant les défaillances de s composants, avec $s \leq s_\#$. Soit $n_o^{c_\neq}$ le nombre de composants de la classe o présents dans la coupe $c_\neq$.

Définition 32. La coupe équivalente à la coupe $c_\neq$ est l'ensemble

$$c_= = \left\{ \left(\frac{n_o^{c_\neq}}{n_o}, o \right) \right\}_{o \in O^\#}. \tag{5.22}$$

$c_=$ est la liste de chaque type d'objet o, associé au ratio entre le nombre d'instances de o présentes dans la coupe $c_{\neq}$ sur le nombre d'instances de l'objet o dans le système $S_{\#}$.

Plusieurs coupes distinctes correspondent à une coupe équivalente. A fortiori, plusieurs histoires peuvent faire apparaître une même coupe équivalente. Soit $N_=$ le nombre de coupes équivalentes différentes, et $C_= = \left\{c_1^=, ..., c_{N_=}^=\right\}$ l'ensemble de ces coupes. Une simulation k est assortie d'une unique coupe équivalente notée $c_k^=$. La coupe $c_l^=$ est associée à l'ensemble des histoires qui lui correspondent $H_l = \{k \leq N / c_k^= = c_l^=\}$. $p_l = \frac{Card(H_l)}{Card(H^+)}$ est le poids de la coupe $c_l^=$.

Exemple 8. L'histoire ([t_{v_0}, V1, **opening**], [t_{v_0}, V2, **opening**], [t_{v_0}, V3, **stopped**], [t_{v_1}(stopped), V1, **stopped**], [t_{v_2}(open), V2, **open**], [t_{c_f}, R, **SSA**]) est caractérisée par la coupe distincte {V1, V3} et par la coupe équivalente {(2/3,V)} qui signifie « perte de 2 vannes sur les 3 ».

Exemple 9. Soit $S^{\#} = \{CCl1, CCl2, V1, V2\}$. L'histoire ([$t_1$, CCl1, **ok**], [$t_2$, CCl2, **ok**],
[t_3, CCl2, **nok**], [t_4, V1, **opening**], [t_5, V1, **open**], [t_6, R, **SSA**]) est caractérisée par la coupe distincte {CCl2} et par la coupe équivalente {(1/2, CCl)} qui signifie « perte d'un contrôle-commande local sur les 2 ».

Ce regroupement des coupes améliore la lisibilité des résultats lorsque le système présente une structure parallèle.

Remarque 3. Il est aussi possible d'envisager un regroupement par « branche » du système. Dans l'exemple précédent, CCl_i et V_i constituent la branche i du système. Dans ce cas, un composant $s^b \in S^{\#}$ appartient à la branche $b \in B = \{b_1, b_2\}$ et n_b est le nombre de composants de la branche b. $S^{\#}$ est une partition des composants répartis en branches : . Si $c_{\neq}$ est une coupe distincte, et si $n_b^{c_{\neq}}$ est le nombre de composants de la branche b présents dans la coupe $c_{\neq}$, alors la coupe équivalente à la classe $c_{\neq}$ est l'ensemble $c_= = \left\{\left(\frac{n_b^{c_{\neq}}}{n_b}, b\right) / b \in B\right\}$. Ainsi, la coupe équivalente à la coupe {CCl2} est {(1/2,b_2)} qui signifie « perte d'un des deux composants de la branche 2 ».

Nombre de simulations nécessaires à la convergence de la méthode
Le nombre de simulations dépend de la probabilité p_{occ} d'occurrence de l'ER et de l'ordre de grandeur du poids des coupes étudiées. Soit $p_{occ} = 0,44$ et une coupe représentant 8,8% des ER ($p_j = 0,088$). Malgré cette probabilité d'occurrence et ce poids relativement importants, la variance de l'estimateur de p_j est importante ($\sigma^2 = 5,17$). De ce fait, environ 795000 simulations sont nécessaires pour obtenir un intervalle de confiance à 95% d'une largeur de 10^{-2}.

5.3.2 Applications aux exemples « fil rouge »

Pour les systèmes munis d'une alimentation et de plusieurs vannes, les coupes sont regroupées par type de composants.

1. Le type « alimentation » ne contient qu'un composant.
2. Le type « vannes de surface » est représenté par les deux vannes du premier système, et par les quatre vannes de surface du second système.
3. Le type « vannes de demi-fond » n'est utilisé que pour les deux vannes de demi-fond du second système.

Afin d'améliorer la lisibilité des résultats, les coupes équivalentes prépondérantes sont affichées si elles représentent plus qu'un pourcentage p des événements redoutés et/ou si elles figurent parmi les n les coupes équivalentes qui provoquent le plus d'événements redoutés.

5.3.2.1 Système composé d'une alimentation et deux vannes

9,56% des 350000 histoires simulées se terminent par l'événement redouté. L'atteinte du seuil maximal par le niveau est provoqué par

1. la perte des deux vannes dans 87,42% des cas,
2. la perte de l'alimentation dans 12,34% des cas,
3. la perte d'une seule vanne dans 0,10% des cas,
4. la perte successive des deux vannes puis de l'alimentation dans 0,07% des cas,
5. la perte d'une seule vanne puis de l'alimentation dans 0,07% des cas.

chraibi2013gaspartnts redoutés sont provoqués par une de ces coupes équivalentes. Ici, la coupe équivalente prépondérante est « perte des deux vannes ».

Remarque 4. Sur l'ensemble des histoires, la perte de deux vannes est moins fréquente que la perte d'une seule vanne, mais la perte de deux vannes provoque plus d'ER que la perte d'une vanne.

5.3.2.2 Système composé d'une alimentation et six vannes

43,93% des 350000 histoires simulées se terminent par l'événement redouté. Le nombre de composants augmente la variation des coupes équivalentes. Les coupes équivalentes prépondérantes sont

1. la perte de trois vannes de surface et d'une vanne de demi-fond dans 23,09% des cas,
2. la perte de deux vannes de surface et d'une vanne de demi-fond dans 18,07% des cas,

3. la perte de deux vannes de surface et de deux vannes de demi-fond dans 14,44% des cas,
4. la perte de trois vannes de surface et de deux vannes de demi-fond dans 13,00% des cas,
5. la perte de trois vannes de surface dans 8,83% dans des cas.

La perte de l'alimentation se classe en dixième position et ne provoque plus que 2,67% des événements redoutés.

5.4 Classification des histoires

5.4.1 Introduction

Comme dans les sections précédentes, l'événement « atteinte d'un seuil maximal par le niveau de la retenue » sera nommé « événement redouté » (ER). L'issue de la simulation est notée $i \in \{1; -1\}$ et désigne la réalisation de l'occurrence ($i = 1$) ou non ($i = -1$) de cet événement à la fin d'une simulation. L'ensemble des simulations est donc classifiable en deux sous-ensembles H^+ et H^-, en fonction de leur issue.

L'influence des instants de panne des composants sur le déroulement du processus de crue a déjà été évoquée. La figure 5.3 dans la section 5.2.3.1 illustre les différentes trajectoires que peut prendre l'évolution du niveau dans la retenue, en fonction de l'instant de défaillance de la vanne chargée de vidanger le réservoir.

Sur cette figure, il est visible qu'une défaillance de la vanne à la sollicitation empêche toute évacuation de la crue et entraine l'événement redouté. Inversement, sans défaillance de la vanne, la crue est correctement évacuée et l'événement redouté n'a pas lieu. Une situation intermédiaire considère une défaillance pendant l'ouverture de la vanne : l'occurrence de l'événement redouté dépend du débit sortant de la vanne ; le débit sortant dépend de la position de la vanne au moment où elle est stoppée pendant son ouverture ; cette position dépend de l'instant de défaillance de la vanne.

Il existe un instant avant lequel la défaillance de la vanne entrainera systématiquement l'événement redouté, et après lequel le débit sortant sera suffisant pour évacuer la crue. Cet instant qui sépare les histoires en deux sous-ensembles sera noté t_{sep}.

En revanche, si le réservoir est vidangé par deux vannes, il n'existe pas d'instant unique t_{sep}. À chaque couple de TTF $(u_1, u_2)_k$ va être associée une trajectoire du niveau h, donnant lieu à l'événement redouté ($i_k = 1$) ou non ($i_k = -1$), où i_k est l'issue de la simulation k. Comment identifier l'ensemble des couples (u_1, u_2) qui caractérisent la frontière entre H^+ et H^- ? Cette problématique est généralisable à un système à $n \geq 2$ composants.

La dynamique de l'évolution du niveau dans la retenue est telle que l'événement redouté ne survient qu'au bout d'un certain délai après la défaillance des composants.

La classification des histoires en fonction des instants de défaillance, pour anticiper l'issue de nouvelles histoires, est une aide au pronostic. A partir des durées de fonctionnement sans défaillance de chaque composant, l'objectif de cette section est de proposer un modèle pour classifier les histoires en fonction de leur issue. Cette classification revient à déterminer une frontière caractérisée par une fonction f. Cette fonction dépend des dates de défaillance en fonctionnement (T_1, ..., T_n) des n composants et retourne l'occurrence ou non de l'événement redouté, ici un débordement du barrage. Les sections 5.4.2 et 5.4.3 détaillent les deux méthodes permettant d'obtenir cette classification dans le cas du système simple décrit dans la section 4.2.1.1.1. La première méthode repose sur un calcul analytique, la seconde sur les résultats des simulations. Les résultats sont ensuite comparés dans la section 5.4.4, que le réservoir soit vidangé par une ou par deux vannes. Finalement, les histoires obtenues après simulation des systèmes « fil rouge » sont à leur tour classifiées pour illustrer la démarche.

5.4.2 Détermination analytique de la frontière

Soit le système simple décrit dans la section 4.2.1.1.1. Ce réservoir de superficie S et de niveau initial h_0 est soumis à une crue en forme d'échelon, d'intensité I_c entre les dates t_{c_0} et t_{c_f}. La vidange est assurée par une vanne dont la débitance est constante et égale à q_{max} une fois son ouverture achevée. Cette ouverture débute à la date t_{v_0} et s'achève à $t_{v_f} = t_{v_0} + d_{ouv}$ où d_{ouv} est la durée d'ouverture de la vanne.

L'expression analytique du niveau dans le réservoir est calculée dans la section 5.2.1.1. Dans le cas d'une défaillance de la vanne au bout de la durée $u \leq d_{ouv}$,

$$h(t) = \begin{cases} h_0 & \text{si } t \leq t_{c_0}, \\ h_0 + \frac{I_c}{S}(t - t_{c_0}) & \text{si } t \in [t_{c_0};\, t_{v_0}], \\ h_0 + \frac{I_c}{S}(t - t_{c_0}) - \frac{q_{max}}{2 \times d_{ouv} \times S}(t - t_{v_0})^2 & \text{si } t \in [t_{v_0};\, t_{v_0} + u], \\ h(t_{v_0} + u) + \frac{I_c}{S} \times (t - (t_{v_0} + u)) - \frac{q_{sor}(t_{v_0}+u)}{S} \times (t - (t_{v_0} + u)) & \text{si } t \in \left[t_{v_0} + u\,;\, t_{c_f}\right], \\ h(t_{v_0} + u) + \frac{I_c}{S} \times \left(t_{c_f} - (t_{v_0} + u)\right) - \frac{q_{sor}(t_{v_0}+u)}{S} \times (t - (t_{v_0} + u)) & \text{si } t \geq t_{c_f} \end{cases} \tag{5.23}$$

où $q_{sor}(t_{v_0} + u) = \frac{q_{max}}{d_{ouv}} u$ désigne la débitance instantanée de la vanne à l'instant de panne $t_{v_0} + u$.

5.4.2.1 Réservoir vidangé par une vanne : calcul analytique de l'instant t_{sep}

L'instant de l'événement redouté est une fonction croissante de l'instant de la défaillance. Donc si la vanne se bloque à t_{sep}, l'événement redouté aura lieu le plus tard possible, c'est-à-dire à l'instant précis de la fin de la crue, noté t_{c_f}. Cette déduction repose sur l'hypothèse de constance des débits étudiés. La TTF $u_{sep} = t_{sep} - t_{v_0}$ est

donc l'instant de panne $u_{ER}(t_{c_f})$ qui correspond à l'instant de fin de crue , calculé dans la section 5.2.1.3. Finalement $t_{sep} = t_{v_0} + u_{ER}(t_{c_f})$, d'où

$$t_{sep} = t_{v_0} + \frac{q_{max}}{d_{ouv}} \left(t_{c_f} - t_{v_0}\right) \left[1 - \sqrt{1 + 2\frac{q_{max}}{d_{ouv}} \frac{(h_{max} - h_0)S + I_c(t_{c_0} - t_{c_f})}{(t_{c_f} - t_{v_0})^2}}\right]. \quad (5.24)$$

5.4.2.2 Réservoir vidangé par deux vannes : calcul analytique de la frontière $u_2^{sep}(u_1)$

Une deuxième vanne est ajoutée dans les mêmes conditions que la section 5.2.4.3.1. Les deux vannes commencent à s'ouvrir au même instant t_{v_0} et pendant la même durée d_{ouv}. Une fois complètement ouvertes, elles ont la même débitance q_{max}. Soit u_i le TTF de la vanne i.

La fonction u_2^{sep} est définie sur $[0, d_{ouv}]$ et retourne le TTF de la vanne 2 correspondant à l'occurrence de l'événement redouté à l'instant de fin de crue, sachant que le TTF de la vanne 1 est u_1 :

$$u_2^{sep}(u_1) = \left(t_{c_f} - t_{v_0} - u_1\right)$$

$$+ \sqrt{\left(t_{c_f} - t_{v_0} - u_1\right)^2 - 2\left\{\frac{d_{ouv}}{q_{max}}\left[S(h_0 - h_{max}) + I_c(t_{c_f} - t_{c_0})\right] - u_1^2 - (t_{c_f} - t_{v_0})u_1\right\}}. \quad (5.25)$$

Cette frontière est illustrée par la figure 5.12. La démonstration de cette expression figure en annexe B.

5.4.3 Classification des histoires simulées

Soit $D = \{d_k\}_{k \leq N}$ l'ensemble des VTTF obtenus à partir des N simulations, où $d_k = \{v_k, i_k\}$ est le VTTF associé à la simulation de l'histoire k. $v_k = [T_1^k, ..., T_{s_\#}^k]$ où $s_\#$ est le nombre de composants du système et $i_k \in \{-1; 1\}$ désigne l'issue de simulation k, c'est-à-dire la réalisation de l'événement redouté ($i_k = 1$) ou non ($i_k = -1$) avant la fin de la simulation.
Soit $H^+ = \{k \leq N / i_k = 1\}$ l'ensemble des histoires dont l'issue est l'événement redouté et $V^+ = \{v_k \in d_k / d_k \in D \text{ et } i_k = 1\}$ l'ensemble des VTTF associés. De même, $H^- = \{k \leq N / i_k = -1\}$ est l'ensemble des histoires dont l'issue est l'absence d'événement redouté et $V^- = \{v_k \in d_k / d_k \in D \text{ et } i_k = -1\}$ l'ensemble des VTTF associés.

5.4.3.1 Cas d'un composant défaillant

Lorsque le système n'est équipé que d'un composant défaillant, il est inutile d'utiliser une technique de classification complexe. Les vecteurs v_kDaont de dimension 1 et t_{sep} est défini par

$$t_{sep} = t_{v_0} + \max_{T_k \in v_k,\, v_k \in V^+}(T_k). \tag{5.26}$$

5.4.3.2 Cas de plusieurs composants

L'objectif est la séparation des VTTF en fonction de l'issue de l'histoire associée. Nous utilisons pour cela une technique d'apprentissage automatique pour la classification binaire des données. Ces SVM (de l'anglais *Support Vector Machine*, aussi nommés « séparateurs à vaste marge » dans la littérature) [Hasan et Boris, 2006], [Cristianini et Shawe-Taylor, 2000], [Wang, 2005] sont introduits dans la section 3.4.

5.4.3.2.1 Choix d'un noyau et de ses paramètres

Les VTTF ne sont pas des données linéairement séparables. Il faut donc projeter les vecteurs dans un autre espace afin d'y réaliser une séparation linéaire. Cette projection se fait par un noyau K.

Le séparateur obtenu est de la forme

$$f(\mathbf{x}) = \sum_{i=1}^{P} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b \tag{5.27}$$

où $\mathbf{x}_i$ désigne un vecteur support. α_i^* et b sont les solutions du problème d'optimisation.

Parmi les noyaux de référence, le noyau gaussien est le plus couramment utilisé. Ce noyau est de la forme

$$K(\mathbf{x}_i, \mathbf{x}) = \sum_{j=1}^{n} \exp\left(-\gamma(x_{ij} - x_j)^2\right) \tag{5.28}$$

où x_j est la $j^{ème}$ coordonnée du vecteur $\mathbf{x}$. Le paramètre γ est une donnée d'entrée du modèle SVM. C'est aussi le cas de l'erreur (ou coût) C présent dans le problème d'optimisation. Sans *a priori* sur γ et C, une solution est de proposer une liste de valeurs pour chacun d'eux et de sélectionner le modèle le plus précis. γ est généralement de l'ordre de $1/n$, il sera choisi parmi la liste $\left[0, \frac{1}{2n}, \frac{1}{n}, \frac{2}{n}, \frac{3}{n}, ..., 1\right]$. C aura pour valeur 10^m, $m \in \{0, ..., 4\}$.

5.4.3.2.2 Optimisation du nombre de VTTF appris

Les SVM sont une méthode d'apprentissage de la fonction séparatrice à partir des données. Le modèle est construit sur un échantillon des données appelé « ensemble

d'apprentissage ». Puis sa robustesse est testée en l'appliquant au reste des données et en comparant le résultat retourné à la donnée réelle. Deux critères permettent de quantifier la robustesse du modèle : la précision et le taux de faux négatifs.

Soit S^+ l'ensemble des données de test dont l'étiquette est positive et S^- celles dont l'étiquette est négative. Soit S^+_{svm}, respectivement S^-_{svm}, l'ensemble des données de test pour lesquelles le modèle retourne une étiquette positive, respectivement négative.

Définition 33. La précision *prec* d'un modèle est le pourcentage de données classées avec exactitude :

$$prec = \left[Card(S^+ \cap S^+_{svm}) + Card(S^- \cap S^-_{svm})\right] \times \frac{100}{N}. \tag{5.29}$$

Définition 34. Le taux de faux négatifs TFN d'un modèle est le pourcentage de données auxquelles le modèle a attribué une étiquette négative alors que l'étiquette réelle est positive. Cela équivaut à prédire à tort l'absence de l'événement redouté, ce qui est optimiste.

$$TFN = Card(S^+ \cap S^-_{svm}) \times \frac{100}{N}. \tag{5.30}$$

Un modèle robuste est caractérisé par une forte précision et un faible taux de faux négatifs.

Quelle proportion des données l'échantillon d'apprentissage doit-il représenter ? Cela dépend du nombre de simulations N et de la dimension des données p. Le risque lié à une taille trop importante de l'échantillon d'apprentissage est un risque de sur-apprentissage ainsi qu'un temps de calcul conséquent. Si la taille de l'échantillon d'apprentissage est trop faible, la précision et le taux de faux négatifs sont médiocres.

L'optimisation de la taille de l'échantillon d'apprentissage consiste donc à augmenter progressivement ce pourcentage des données jusqu'à obtenir la précision et le taux de faux négatifs désirés.

5.4.3.2.3 Options disponibles

Dans le cas d'un modèle séparateur global, les histoires sont séparées en fonction des TTF de tous les composants. Parfois, l'analyste n'est intéressé que par l'impact de quelques TTF sur l'issue de la simulation. Différentes options sont donc disponibles :

1. prise en compte de tous les composants (modèle global),
2. prise en compte des composants présents dans la coupe équivalente prépondérante,
3. prise en compte du composant le plus important,
4. prise en compte d'une liste de k composants paramétrée librement par l'utilisateur.

Pour les options 2 à 4, trois sous-options sont également disponibles.

1. Tous les composants sont parfaits, sauf les k composants choisis (données de dimension k).
2. Seules les histoires où aucun des k composants choisis ne démarre sont considérées (données de dimension $n-k$).
3. Seules les histoires où aucun des k composants choisis ne tombe en panne sont considérées (données de dimension $n-k$).

5.4.4 Comparaison des résultats

5.4.4.1 Système simple à une vanne

D'après le calcul analytique donné par l'équation 5.16, $u_{ER}(t_{c_f}) = 2179,32$ secondes. D'après les simulations, $\max_{T_k \in v_k,\, v_k \in V^+}(T_k) = 2179,13.$ secondes. Ces résultats sont en accord.

5.4.4.2 Système simple à deux vannes

La figure 5.13 illustre la classification des histoires pour le même système simple, mais vidangé par deux vannes. L'échantillon des données d'apprentissage est initialement constitué de 0,05% des histoires simulées, soit 75 histoires. Le nombre de données d'apprentissage est augmenté de 75 histoires jusqu'à ce que la précision du modèle soit supérieure à 99% et le taux de faux négatifs inférieur à 1% des données de test. Finalement, pour cet exemple, le modèle a besoin de 0,7% des histoires pour l'apprentissage des données en respectant ces critères, comme le montre la figure 5.12.

Le séparateur matérialisé par une frontière en trait continu bleu est issu du calcul analytique et a pour équation

$$S\,(h_0 - h_{max}) + I_c(t_{c_f} - t_{c_0}) + \frac{q_{max}}{2d_{ouv}}\left(u_1^2 + u_2^2\right) + \frac{q_{max}}{d_{ouv}}u_1u_2 + \frac{q_{max}}{d_{ouv}}(t_{c_f} - t_{v_0})\,(u_1 - u_2) = 0 \tag{5.31}$$

Les couples (u_1, u_2) de TTF solutions de cette équation sont à la frontière entre les deux issues possibles.

Le modèle SVM décrit par la frontière en tirets rouges fournit un séparateur de la forme

$$\sum_{i=1}^{P} \alpha_i^* \left[\exp\left(-\gamma(sv_{i,1} - u_1)^2\right) + \exp\left(-\gamma(sv_{i,2} - u_2)^2\right)\right] + b = 0 \tag{5.32}$$

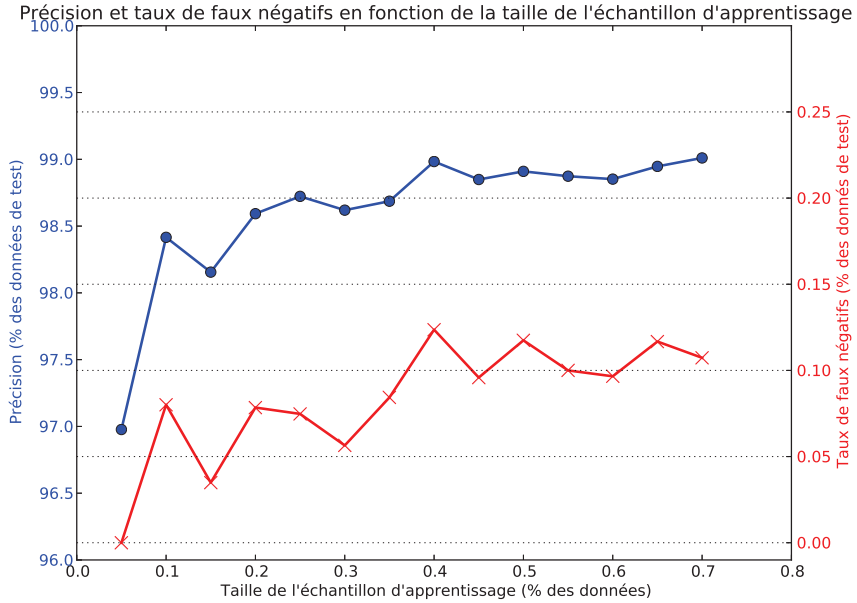

FIGURE 5.12 – Précision et taux de faux négatifs en fonction de la taille de l'échantillon d'apprentissage

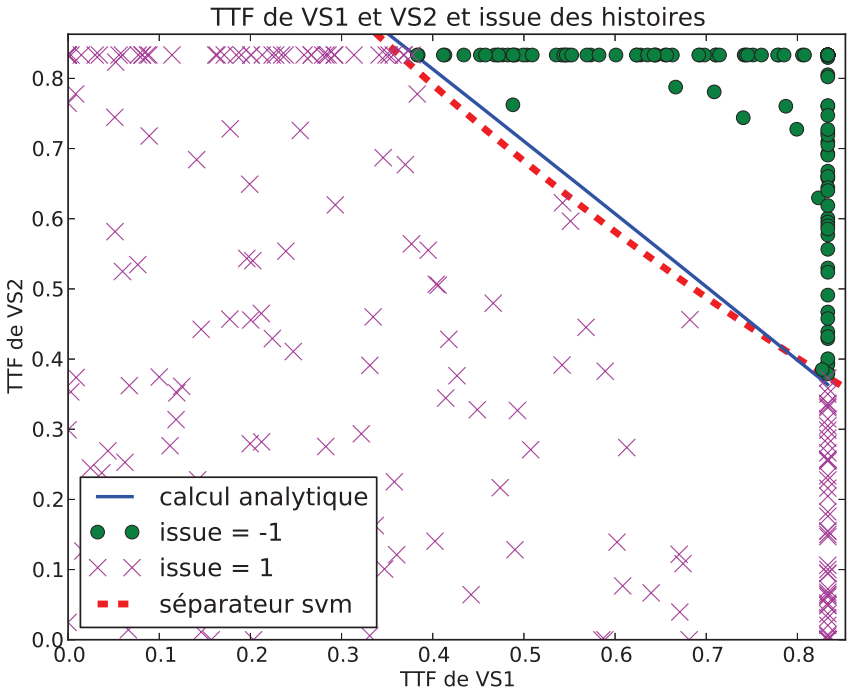

FIGURE 5.13 – Séparation des histoires en fonction des TTF de deux vannes

où $sv_{i,j}$ désigne la $j^{ème}$ coordonnée et α_i^* le coefficient du $i^{ème}$ support vecteur, avec $i \leq P = 335$. Cela signifie que les coordonnées et le coefficient de 335 vecteurs support caractérisent la frontière. Le modèle fournit également la valeur de γ et de b.

Les deux courbes séparent les données de manière similaire sur la figure 5.13.

5.4.5 Application aux exemples « fil rouge » et conclusion

Le modèle de classification SVM est également performant lors de son application aux deux systèmes « Fil Rouge ». Les VTTF pour ces exemples sont respectivement de dimension 3 et 7. Seules 0,1% des histoires sont nécessaires pour fournir un séparateur satisfaisant les critères de précision et de taux de faux négatifs pour le système à trois composants. Le système à sept composants en nécessite 1,6%.

Au-delà de la dimension 2, la représentation graphique des frontières n'est pas disponible, mais l'équation du séparateur est connue :

$$f(\mathbf{x}) = \sum_{i=1}^{P} \alpha_i^* \sum_{j=1}^{n} \exp\left(-\gamma(x_{ij} - x_j)^2\right) + b = 0$$

où P est le nombre de vecteurs supports du modèle.

Soit L la liste des composants dont on veut mesurer l'impact du TTF sur l'occurrence de l'événement redouté, les autres composants présents étant parfaits (option 4.1 du modèle). Si $Card(L) = 2$, il est théoriquement possible de visualiser la classification des histoires concernées, à condition que ces histoires soient suffisamment représentées dans chacune des classes. Cette condition n'est pas respectée dans les deux cas suivants, ce qui mène à une classification infructeuse :

1. Soit le système à trois composants {Alim, VS1, VS2} et L_1 ={Alim, VS1} la liste des composants dont on veut mesurer l'impact de le TTF sur l'occurrence de l'événement redouté, lorsque VS2 est parfait. Le taux de défaillance de l'alimentation est tel que seules 16 histoires sur 350000 présentent une panne de l'alimentation entre le début et la fin du processus d'ouverture des vannes. Le modèle ne peut donc pas déterminer précisément quel TTF de l'alimentation sépare les histoires.
2. Soit le système à sept composants {Alim, VS1, VS2, VS3, VS4, VDF5, VDF6} et L_1 ={VS1, VV5} la liste des composants dont on veut mesurer l'impact du TTF sur l'occurrence de l'événement redouté lorsque les autres composants sont parfaits. Le dimensionnement du barrage est tel que trois vannes de surface et une vanne de demi-fond suffisent à évacuer la crue. Donc même si la vanne de surface et la vanne de demi-fond refusent simultanément de s'ouvrir à la sollicitation, une seule issue est possible : l'évitement de l'événement redouté. La classification des histoires n'a donc pas de sens.

La séparation des histoires en fonction de leur issue et de leur TTF n'apporte donc pas forcément de résultats visuellement compréhensibles. Toutefois, l'application du

modèle de classification à un nouveau jeu de durées de fonctionnement avant défaillance pronostique l'issue de cette nouvelle histoire sans recourir à sa simulation complète.

5.5 Conclusion et perspectives

A partir des histoires générées par PyCATSHOO et de l'information temporelle qu'elles contiennent, différents indicateurs sont proposés.

1. La probabilité d'occurrence de l'événement redouté répond à l'objectif de caractérisation des risques. Cette probabilité est une probabilité conditionnée par l'arrivée d'une crue d'un type donné. Elle participe au calcul de fréquence annuelle de l'événement redouté et à la hiérarchisation des évacuateurs de crues (EdC) vis-à-vis du risque lié à l'événement redouté. Ce calcul de fréquence annuelle considère les probabilités d'occurrence de l'ER suite à chaque type de crue considéré et la fréquence de ce type de crue. La fréquence de l'événement redouté pour un type de crue est un autre résultat. Il hiérarchise les risques associés aux différentes crues. Par exemple, le niveau de risque le plus important peut provenir de la crue millénale ou de crues moins intenses mais plus fréquentes. Ces deux types de classements (hiérarchisation des EdC et hiérarchisation des types de crue) sont des indicateurs faciles à obtenir à partir de la probabilité d'occurrence de l'ER pour plusieurs EdC ou pour plusieurs types de crues.
2. Les coupes prépondérantes permettent d'identifier les événements (défaillances) les plus contributeurs à la probabilité de l'événement redouté. Les scénarios de défaillance, associés à leur probabilité d'occurrence, sont hiérarchisés par ordre d'importance de cette contribution probabiliste. Des leviers d'amélioration de la fiabilité de l'évacuateur sont identifiés, qu'il s'agisse de modification d'architecture, de dimensionnement de composants, d'actions qui augmentent la fiabilité matérielle ou humaine, ou de modifications des conditions d'exploitation.
3. La séparation des histoires par rapport à l'occurrence ou non de l'événement redouté, en fonction des durées de fonctionnement avant défaillance de chaque composant du système, est un modèle qui exploite au maximum les données temporelles contenues dans les histoires simulées. Cette classification pronostique, à partir d'un jeu de nouvelles durées de fonctionnement avant défaillance, l'issue de l'histoire associée.

En s'inspirant de ces indicateurs, comment envisager un module d'aide à la décision pour l'opérateur, en plein déroulement de la crue ? Imaginons que l'opérateur, à un instant t de la crue, dresse le constat des composants en panne. Il sait aussi quels composants ne sont pas encore en panne à cet instant t. Le modèle de séparation va classer ce jeu de données et pronostiquer l'occurrence ou non de l'événement redouté. Sachant cela, l'opérateur priorise ses actions de réparation en cas de pannes simultanées de plusieurs composants.

Comment rendre cette aide à la décision plus rapide et plus efficace ? PyCATSHOO va générer de nouvelles simulations en forçant les nouvelles histoires à respecter les défaillances et les bons fonctionnements connus de l'opérateur à l'instant t. Les indicateurs tels que la probabilité d'occurrence de l'événement redouté, les coupes équivalentes prépondérantes et la classification sont alors réactualisées. Sachant cela, l'opérateur va être plus ou moins optimiste sur l'issue de la crue. Le modèle de classification sépare plus précisément les nouvelles histoires générées. Les durées requises avant défaillance des composants encore en fonctionnement sont identifiées à partir de ce modèle, et l'opérateur va concentrer ses actions de maintenance sur les composants indispensables à l'évacuation de la crue.

Cette éventuelle démarche n'est pas instantanée et ne convient pas à la gestion d'une situation d'urgence. La simulation d'un nombre significatif d'histoires requiert en effet quelques heures. Après cette étape de simulations, les temps de calcul nécessaires à l'estimation des indicateurs réactualisés sont en revanche de l'ordre de quelques minutes. Enfin, la classification des histoires a été définie dans l'hypothèse de composants non réparables. L'accélération des simulations et de l'exploitation des résultats, ainsi que la prise en compte des réparations dans la séparation des histoires, figurent donc parmi les pistes de réflexion et les perspectives ouvertes par ces travaux de thèse.

Chapitre 6

Importance dynamique d'un composant

Le chapitre 5 propose une démarche prévisionnelle fondée sur la classification des histoires et l'estimation de la probabilité d'occurrence. Des perspectives pour améliorer l'aide à la décision, phase finale de cette démarche, ont été dégagées.

Ce dernier chapitre avance également un indicateur de fiabilité dynamique destiné à l'aide à la décision. Les histoires sont traitées selon une orientation différente. L'objectif est d'identifier quel est le composant le plus important dans un système dynamique hybride, à tout instant de son fonctionnement. L'importance d'un composant est mesurée par la variation de la probabilité d'occurrence de l'événement redouté provoquée par la défaillance de ce composant, par rapport à une situation de référence.

Cette section propose la définition d'un nouvel indicateur de fiabilité, dans un cadre dynamique. L'importance dynamique de Birnbaum est inspirée du facteur d'importance de Birnbaum, mais tient compte de la date des défaillances des composants.

Le but est de comparer ces importances et d'identifier le composant le plus important, c'est-à-dire le composant dont la défaillance a le plus d'impact sur la probabilité de l'événement indésirable, par rapport à la situation normale. Estimer l'importance dynamique d'un groupe de composants apportera également quelques informations. « L'importance dynamique d'un système » est en fait un raccourci de langage pour désigner l'estimation de « l'importance dynamique de chacun des composants de ce système ».

Après avoir défini l'importance dynamique et le périmètre des systèmes étudiés dans la section 6.1, la section 6.2 expose le calcul analytique de cette grandeur pour un cas-test simple à deux composants. La section 6.3 détaille ensuite la méthodologie pour retrouver ce type de résultats à partir des simulations. Puis la section 6.4 compare ces deux méthodes et donne des éléments d'interprétation des courbes obtenues. Finalement, la section 6.4.5 est une application aux systèmes « fil rouge ».

6.1 Introduction

Comme le montre l'état de l'art dans la section 2.2.3, fiabilité dynamique et mesures d'importance sont rarement associées. [Cocozza-Thivent, 1997] construit des facteurs d'importance dépendant du temps, appliqués à des systèmes de petite taille. La définition proposée pour le facteur d'importance de Birnbaum à l'instant t est $B_c(t) = P\left(\{S(t), c(t)\} \text{ et } \left\{\bar{S}(t), \bar{c}(t)\right\}\right)$ où $\{S(t), c(t)\}$ représente l'événement « le système S et le composant c sont en marche à l'instant t » et $\left\{\bar{S}(t), \bar{c}(t)\right\}$ représente l'événement « le système S et le composant c sont en panne à l'instant t ». $B_c(t)$ représente donc la probabilité que le système soit en marche à l'instant t si le composant c est en marche à cet instant et que le système soit en panne toujours au même instant si le composant c est défectueux.

Dans notre cas, la défaillance du système, c'est-à-dire l'atteinte d'un seuil de sûreté par le niveau de la retenue (événement redouté ER), est différé par rapport au temps de défaillance des composants. Ce délai correspond au temps nécessaire à la montée de l'eau dans le réservoir. Nous proposons donc une autre définition du facteur d'importance de Birnbaum.

6.1.1 Définition d'une mesure d'importance dynamique

Cette section rappelle les différents facteurs d'importance définis dans le chapitre 2.1.1.6 et en propose une expression dynamique afin d'utiliser l'information temporelle contenue dans les histoires.

Notons ER la réalisation de l'événement redouté et eb_c la réalisation d'un événement de base, liée à la défaillance du composant c. T_c est la date de la défaillance du composant c. $\bar{\omega}$ désigne le complémentaire de l'événement ω.

Parmi ces facteurs d'importance dynamique, nous avons porté arbitrairement notre intérêt sur l'indicateur de Birnbaum.

L'importance dynamique d'un composant n'a de sens et n'est définie que pendant la période où le composant c peut tomber en panne. On appellera cette période « temps mission du composant c ».

L'importance de Birnbaum, qu'elle soit statique ou dynamique, permet de mesurer la variabilité de la probabilité d'un événement indésirable causée par une situation particulière par rapport à une situation de référence.

Cette situation de référence sera appelée « situation normale » si elle représente le déroulement du processus de crue sans modification des paramètres de fiabilité, c'est-à-dire sans hypothèse a priori sur l'instant de défaillance d'un composant.

Le temps de mission du composant c est l'intervalle de temps pendant lequel il peut connaitre une défaillance. C'est également l'intervalle de définition de l'importance

	Version classique	Version dynamique
Indicateur de Birnbaum	$I_B(eb_c) = P(ER/eb_c) - P(ER/\overline{eb_c})$	$I_c(t) = P(ER/T_c \leq t) - P(ER/T_c > t)$
Facteur d'Accroissement de Risques (FAR)	$FAR(eb_c) = \frac{P(ER/eb_c)-P(ER)}{P(ER)}$	$FAR_c(t) = \frac{P(ER/T_c \leq t)-P(ER)}{P(ER)}$
Facteur de Diminution de Risques (FDR)	$FDR(eb_c) = \frac{P(ER)-P(ER/\overline{eb_c})}{P(ER)}$	$FDR_c(t) = \frac{P(ER)-P(ER/T_c>t)}{P(ER)}$
Risk Achievment Worth (RAW)	$RAW(eb_c) = \frac{P(ER/eb_c)}{P(ER)}$	$RAW_c(t) = \frac{P(ER/T_c \leq t)}{P(ER)}$
Risk Reduction Worth (RRW)	$RRW(eb_c) = \frac{P(ER)}{P(ER/\overline{eb_c})}$	$RRW_c(t) = \frac{P(ER)}{P(ER/T_c>t)}$
Facteur de Sensibilité (FS)	$FS(eb_c, \delta p_c) = \frac{P(ER/p'_c=p_c+\delta p_c)-P(ER/p_c)}{P(ER/p_c)}$	$FS_c(t, \delta t) = \frac{P(ER/T_c \leq t+\delta t)-P(ER/T_c \leq t)}{P(ER/T_c \leq t)}$

TABLE 6.1 – Proposition de facteurs d'importance dynamiques

dynamique du composant c. On appellera « début de la mission de c » la borne inférieure de la mission de c, et « fin de la mission de c » la borne supérieure de la mission du composant c.

6.1.2 Systèmes étudiés

Les systèmes étudiés sont les deux systèmes « Fil Rouge », inspirés du fonctionnement d'un évacuateur de crues muni de six vannes. FR2 est composé d'une alimentation et de six vannes. FR1 est le même système, mais les paramètres de fiabilité de quatre vannes sont modifiés. Celles-ci sont supposées parfaites ; FR1 est donc muni d'une alimentation et de seulement deux vannes faillibles.

Afin de procéder au calcul analytique de l'importance dynamique, ces systèmes seront encore simplifiés et un système à deux composants seulement sera étudié : l'alimentation et une vanne. Cela signifie que seule la vanne 1 peut défaillir de manière aléatoire. Or si les 5 autres vannes de FR2 étaient parfaites, l'atteinte du seuil de sûreté serait un événement très rare. Aussi, la vanne 2 est systématiquement considérée comme inopérationnelle, en lui attribuant une probabilité de défaillance à la sollicitation égale à 1. Les 4 autres vannes sont considérées comme parfaites ; il est impossible que leur processus d'ouverture soit stoppé avant l'ouverture totale, même si l'alimentation est défaillante.

Soit ce système à deux composants, constitué d'une alimentation électrique et d'une vanne. L'alimentation électrique a pour mission de fournir de l'énergie à la vanne pour son ouverture. Cette mission débute dès l'arrivée de la crue. La vanne a pour mission de s'ouvrir au moment où c'est nécessaire. Cette mission débute à la sollicitation de

la vanne pour son ouverture et s'achève lorsque la vanne est totalement ouverte. L'alimentation peut tomber en panne dès le début de la crue, et jusqu'à la fin de la crue. La vanne peut tomber en panne à partir du début de son ouverture, jusqu'à la fin de son ouverture. Comme dans la section 5.4, il existe un instant avant lequel la défaillance de l'un des deux composants entrainera systématiquement un débordement, et après lequel le débit sortant sera suffisant pour évacuer la crue sans débordement. Cet instant qui sépare les histoires en deux sous-ensembles sera noté t_{sep}.

6.2 Calcul analytique de l'importance dynamique pour le système à deux composants

Notre but est d'exprimer le facteur d'importance dynamique de Birnbaum :

$$I_c(t) = P(ER/T_c \leq t) - P(ER/T_c > t) \tag{6.1}$$

où ER désigne l'événement redouté (atteinte du seuil de sûreté dans notre cas) et T_c est la variable correspondant au temps de défaillance du composant c. Dans ce calcul, c'est le temps de la première défaillance qui compte. En effet, si l'alimentation tombe en panne avant la fin de sa mission, le processus d'ouverture de la vanne est stoppé, même si la vanne est en état de marche. Si c'est la vanne qui tombe en panne en premier, une défaillance de l'alimentation n'aura pas d'impact supplémentaire sur l'arrêt du processus d'ouverture. C'est pourquoi le calcul analytique de l'importance s'effectue par morceaux, en fonction de la position de l'instant de la première défaillance par rapport à t_{sep}.

En théorie, il est possible de déterminer t_{sep} en résolvant l'équation différentielle qui régit l'évolution du niveau dans la retenue. Mais cette résolution est complexe, aussi t_{sep} est-il évalué par l'observation des histoires simulées. En dimension 1, la classification des histoires ne nécessite pas de modèle type SVM. Un nombre suffisamment grand d'histoires simulées permet de situer t_{sep} avec précision. En effet, t_{sep} se situe après le dernier instant de défaillance qui provoque l'ER, mais avant le premier instant de défaillance qui n'en provoque pas. C'est la seule partie « non analytique » du calcul théorique de l'importance dynamique.

Afin d'effectuer cette partition par rapport à t_{sep}, la propriété suivante sera utilisée :

Proposition 35. *Soit A et B deux événements. Soit C un événement et $\bar{C}$ son complémentaire. Alors*

$$P(A/B) = P(A/B,\, C) \times P(C/B) + P(A/B,\, \bar{C}) \times P(\bar{C}/B) \tag{6.2}$$

La démonstration de cette proposition figure dans l'annexe C.

Cette proposition peut être étendue au cas où les événements $\{C_1, ..., C_n\}$ forment une partition.

Proposition 36. *Soit A et B deux événements. Soit* $\{C_1, ..., C_n\}$ *une partition d'événements. Alors*

$$P(A/B) = \sum_{i=1}^{n} P(A/B, C_i) \times P(C_i/B) \tag{6.3}$$

Pour écrire l'expression analytique, nous allons définir quelques fonctions qui seront utilisées de manière récurrente dans les calculs.

Définition 37. On note f_{alim} et f_V les fonctions de densités des lois des temps de défaillance de l'alimentation T_{alim} et de la vanne T_V. Ces lois sont des lois exponentielles pour simplifier les calculs.

$$f_{alim}(t_1) = \lambda_{alim} \exp(-\lambda_{alim} t_1) \mathbf{1}\{t_1 \geq 0\} \tag{6.4}$$

où $t_1 = 0$ désigne l'instant du début de la crue.

$$f_V(t_2) = \lambda_V \exp(-\lambda_V t_2) \mathbf{1}\{t_2 \geq 0\} \tag{6.5}$$

où $t_2 = 0$ désigne l'instant du début de l'ouverture t_{ouv}. On note γ_V la probabilité de défaillance à la sollicitation de la vanne.

Définition 38. On note F_{alim} et F_V les fonctions de répartition des lois des temps de défaillance de l'alimentation T_{alim} et de la vanne T_V.

$$F_{alim}(t_1) = P(T_{alim} \leq t_1) = 1 - \exp(-\lambda_{alim} t_1) \tag{6.6}$$

$$F_V(t_2) = P(T_V \leq t_2) = 1 - \exp(-\lambda_V t_2). \tag{6.7}$$

Les lois de T_{alim} et T_V sont indépendantes.

6.2.1 Expression littérale de l'importance dynamique de Birnbaum pour l'alimentation

- Calcul de $P(ER/T_{alim} \leq t)$
  - Pour $t \leq t_{sep}$

$$P(ER/T_{alim} \leq t) = 1 \text{ pour } t \leq t_{sep}. \tag{6.8}$$

– Pour $t > t_{sep}$

$$P(ER/T_{alim} \leq t) \tag{6.9}$$

$$= \frac{F_{alim}(t_{sep}) + [(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V]\,[F_{alim}(t) - F_{alim}(t_{sep})]}{F_{alim}(t)}.$$

La preuve de cette expression figure dans l'annexe C.

– Calcul de $P(ER/T_{alim} > t)$
  – Pour $t \leq t_{sep}$

$$P(ER/T_{alim} > t) \tag{6.10}$$

$$= \frac{[F_{alim}(t_{sep}) - F_{alim}(t)] + [(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V]\,[1 - F_{alim}(t_{sep})]}{1 - F_{alim}(t)}.$$

La preuve de cette expression figure dans l'annexe C.
  – Pour $t > t_{sep}$

$$P(ER/T_{alim} > t) = P\left(T_V \leq t_{sep}\right) = (1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V.$$

Finalement, pour $t \leq t_{sep}$,

$I_{alim}(t) = 1 - \frac{[F_{alim}(t_{sep}) - F_{alim}(t)] + [(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V][1 - F_{alim}(t_{sep})]}{1 - F_{alim}(t)}$

$= \frac{[(1-\gamma_V)(1-F_V(t_{sep}-t_{ouv}))][1-F_{alim}(t_{sep})]}{1-F_{alim}(t)}$

et pour $t > t_{sep}$,

$I_{alim}(t)$

$= \frac{F_{alim}(t_{sep}) + [(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V][F_{alim}(t) - F_{alim}(t_{sep})]}{F_{alim}(t)} - [(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V]$

$= \frac{(1-F_V(t_{sep}-t_{ouv}))(1-\gamma_V)F_{alim}(t_{sep})}{F_{alim}(t)}$

$$I_{alim}(t) = \begin{cases} \frac{[(1-\gamma_V)(1-F_V(t_{sep}-t_{ouv}))][1-F_{alim}(t_{sep})]}{1-F_{alim}(t)} & \text{si } t \leq t_{sep} \\ \frac{(1-F_V(t_{sep}-t_{ouv}))(1-\gamma_V)F_{alim}(t_{sep})}{F_{alim}(t)} & \text{si } \ t > t_{sep} \end{cases} \tag{6.11}$$

soit

$$I_{alim}(t) = \begin{cases} (1-\gamma_V)e^{-\lambda_V(t_{sep}-t_{ouv})}\exp\left(-\lambda_{alim}(t_{sep}-t)\right) & \text{si } t \leq t_{sep} \\ \frac{(1-\gamma_V)e^{-\lambda_V(t_{sep}-t_{ouv})}\left[1-e^{-\lambda_{alim}t_{sep}}\right]}{1-e^{-\lambda_{alim}t}} & \text{si } \ t > t_{sep} \end{cases}. \tag{6.12}$$

6.2.2 Expression littérale de l'importance dynamique de Birnbaum pour la vanne

L'importance dynamique de la vanne est définie seulement pendant la durée d'ouverture de celle-ci, pendant laquelle une défaillance est possible.

- Calcul de $P(ER/T_V \leq t)$
  - Pour $t \leq t_{sep}$

$$P(ER/T_V \leq t) = 1 \text{ pour } t \leq t_{sep}. \tag{6.13}$$

  - Pour $t > t_{sep}$

$$P(ER/T_V \leq t) \tag{6.14}$$
$$= \frac{\gamma_V + (1-\gamma_V)\left[F_V(t_{sep}-t_{ouv}) + F_{alim}(t_{sep})\left[F_V(t-t_{ouv}) - F_V(t_{sep}-t_{ouv})\right]\right]}{\gamma_V + (1-\gamma_V)F_V(t-t_{ouv})}.$$

    La preuve de cette expression figure dans l'annexe C.
- Calcul de $P(ER/T_V > t)$
  - Pour $t \leq t_{sep}$

$$P(ER/T_V > t) \tag{6.15}$$
$$= \frac{\left[F_V(t_{sep}-t_{ouv}) - F_V(t-t_{ouv})\right] + F_{alim}(t_{sep})\left[1 - F_V(t_{sep}-t_{ouv})\right]}{1 - F_V(t-t_{ouv})}.$$

    La preuve de cette expression figure dans l'annexe C.
  - Pour $t > t_{sep}$

$$P(ER/T_V > t) = P\left(T_{alim} \leq t_{sep}\right) = F_{alim}(t_{sep}). \tag{6.16}$$

Finalement, pour $t \leq t_{sep}$,

$I_V(t) = 1 - \frac{[F_V(t_{sep}-t_{ouv}) - F_V(t-t_{ouv})] + F_{alim}(t_{sep})[1-F_V(t_{sep}-t_{ouv})]}{1-F_V(t-t_{ouv})}$

$= \frac{[1-F_{alim}(t_{sep})][1-F_V(t_{sep}-t_{ouv})]}{1-F_V(t-t_{ouv})}$

et pour $t > t_{sep}$,

$I_V(t) = \frac{\gamma_V + (1-\gamma_V)[F_V(t_{sep}-t_{ouv}) + F_{alim}(t_{sep})[F_V(t-t_{ouv}) - F_V(t_{sep}-t_{ouv})]]}{\gamma_V + (1-\gamma_V)F_V(t-t_{ouv})} - F_{alim}(t_{sep})$

$= \frac{[\gamma_V + (1-\gamma_V)F_V(t_{sep}-t_{ouv})][1-F_{alim}(t_{sep})]}{\gamma_V + (1-\gamma_V)F_V(t-t_{ouv})}$.

D'où

$$I_V(t) = \begin{cases} \frac{[1-F_{alim}(t_{sep})][1-F_V(t_{sep}-t_{ouv})]}{1-F_V(t-t_{ouv})} & \text{si } t \leq t_{sep} \\ \frac{[\gamma_V + (1-\gamma_V)F_V(t_{sep}-t_{ouv})][1-F_{alim}(t_{sep})]}{\gamma_V + (1-\gamma_V)F_V(t-t_{ouv})} & \text{si } t > t_{sep} \end{cases} \tag{6.17}$$

soit

$$I_V(t) = \begin{cases} e^{-\lambda_{alim} t_{sep}} \exp\left(-\lambda_V(t_{sep}-t)\right) & \text{si } t \leq t_{sep} \\ \frac{\left[1+\gamma_V-(1-\gamma_V)e^{\lambda_V(t_{sep}-t_{ouv})}\right]e^{-\lambda_{alim} t_{sep}}}{\gamma_V + (1-\gamma_V)\left(1-e^{\lambda_V(t-t_{ouv})}\right)} & \text{si } t > t_{sep} \end{cases}. \tag{6.18}$$

6.3 Estimation à partir des histoires simulées

L'estimation de l'importance du composant c consiste à construire quatre matrices puis à les sommer. En effet,

$I_c(t) = P(ER/T_c \leq t) - P(ER/T_c > t) = \frac{P(ER, T_c \leq t)}{P(T_c \leq t)} - \frac{P(ER, T_c > t)}{P(T_c > t)}$

$= \frac{\#\{ER, T_c \leq t\}}{\#\{T_c \leq t\}} - \frac{\#\{ER, T_c > t\}}{\#\{T_c > t\}}$ où $\#\{E_i\}$ désigne le nombre d'histoires pour lesquelles l'événement E_i est réalisé.

Ceci nécessite l'introduction des notations suivantes :

- M_{infER}, M_{inf}, M_{supER} et M_{sup} sont les matrices dédiées aux comptages respectifs de $\#\{ER, T_c \leq t\}$, $\#\{T_c \leq t\}$, $\#\{ER, T_c > t\}$ et $\#\{T_c > t\}$;
- t_{ck} est l'instant de défaillance du composant c pour l'histoire k. Si le composant c n'est pas défaillant pendant l'histoire k, alors t_{ck} a pour valeur la date de fin de mission du composant c ;
- ndt est le nombre de divisions de la période sur laquelle est calculée l'importance du composant c ;
- N est le nombre d'histoires simulées et N_{ER} est le nombre d'histoires dont l'issue est l'ER. M_{inf} et M_{sup} sont composées de N lignes et de $ndt+1$ colonnes. M_{infER} et M_{supER} sont composées de N_{ER} lignes et de $ndt+1$ colonnes ;
- le vecteur $[t_0, t_1, ..., t_{ndt}]$ est composé des instants pour lesquels est estimée l'importance dynamique ;
- la fonction f est définie par $f : \begin{array}{ccc} [t_0, t_1, ..., t_{ndt}] & \mapsto & \{0,1\} \\ t_i & \to & \{t_{ck} \leq t_i\} \end{array}$.

La $k^{ème}$ ligne de M_{inf} est l'image de $[t_0, t_1, ..., t_{ndt}]$ par la fonction f. C'est donc un vecteur de 0 jusqu'à la $m^{ème}$ composante et de 1 à partir de la $(m+1)^{ème}$ composante. $m = E\left(\frac{t_{ck} \times ndt}{t_{ndt} - t_0}\right)$ où $E(x)$ désigne la partie entière du réel x.

Construite de manière complémentaire, la $k^{ème}$ ligne de M_{sup} est un vecteur de 1 jusqu'à la $m^{ème}$ composante et de 0 à partir de la $(m+1)^{ème}$ composante.

La construction de M_{infER} et M_{supER} est similaire à celle de M_{inf} et M_{sup} où seules sont gardées les N_{ER} lignes représentant les histoires avec ER.

Les colonnes de chaque matrice sont ensuite sommées et les vecteurs obtenus S_{infER}, S_{inf}, S_{supER} et S_{sup} sont utilisés pour calculer l'importance dynamique. Si S_{inf} ou S_{sup} contiennent des 0 (pas d'histoires avec $T_c \leq t$ pour les petites valeurs de t, ou d'histoires avec $T_c > t$ pour les grandes valeurs de t), alors l'importance n'est pas définie à t.

Les données utilisées sont les histoires. L'importance est alors définie pour tout $t \in [t_{c_0}, t_{c_f}]$ par $I_c(t) = P(ER/T_c \leq t) - P(ER/T_c > t)$ où t_{c_0} désigne l'instant de début de la crue, t_{c_f} l'instant de fin de la crue et T_c l'instant de défaillance du composant c à partir du début de la crue.

L'avantage de ce calcul est la prise en compte de délais éventuels du processus d'ouverture, causés par exemple par un retard de l'opérateur ou une défaillance d'un actionneur.

L'inconvénient est l'augmentation du temps de calcul de l'importance. Par exemple, le processus d'ouverture de la vanne ne dure qu'une heure, alors que la crue dure environ 80 heures. Pour ce type de composant, la valeur de $P(ER/T_c \leq t) - P(ER/T_c > t)$ va être estimée sur les intervalles $[t_{c_0}, t_{v_0}[$ et $[t_{v_f}, t_{c_f}[$, où t_{v_0} et t_{v_f} sont respectivement les instants de début et de fin d'ouverture de la vanne, alors qu'une défaillance du composant c est impossible sur ces intervalles. Un temps de calcul considérable est donc utilisé pour retourner le résultat d'une opération arithmétique invalide (NaN).

Une autre option, qui n'est pas considérée dans ce manuscrit, consiste à estimer l'importance dynamique d'un composant seulement sur la durée de son fonctionnement. Les données utilisées alors sont les vecteurs des durées de fonctionnement avant défaillance (VTTF), dont la construction est détaillée dans le chapitre 5.1.3. Les calculs sont plus rapides, mais l'information temporelle qui repère les dates des défaillances par rapport au déroulement de la crue n'est plus considérée.

Une troisième option réunit les avantages des deux premiers choix. L'utilisateur est invité à indiquer quelle est la fenêtre de temps sur laquelle il veut estimer l'importance de chaque composant . Cela requiert une connaissance préalable du déroulement du processus. Cette connaissance est obtenue par exemple en observant un échantillon d'histoires ou en estimant l'importance dynamique une première fois sur la durée de la crue. Cette première estimation utilise alors un pas de temps grossier ou un faible nombre d'histoires. L'utilisation de cette troisième option permet ensuite d'estimer finement l'importance dynamique de chaque composant, uniquement sur la période de la crue la plus intéressante.

6.4 Résultats : comparaison et interprétation, pour un système à deux composants

L'interprétation de l'importance dynamique est possible en suivant trois axes d'analyse. Il s'agit de la valeur de l'importance au début de la mission du composant, de la valeur de l'importance à la fin de sa mission et de l'allure de la courbe entre ces deux points.

6.4.1 Importance au début de la mission du composant

Soit t_{deb} l'instant du début de la mission du composant c. Estimer l'importance dynamique du composant c à la date t_{deb} revient à écrire $I_c(t_{deb}) = P(ER/T_c \leq t_{deb}) - P(ER/T_c > t_{deb})$. D'une part, une défaillance du composant c avant le début de sa mission revient à considérer que le composant c est systématiquement défaillant à la sollicitation (sa probabilité de défaillance à la sollicitation est égale à 1). Soit def_c l'événement élémentaire « défaillance à la sollicitation

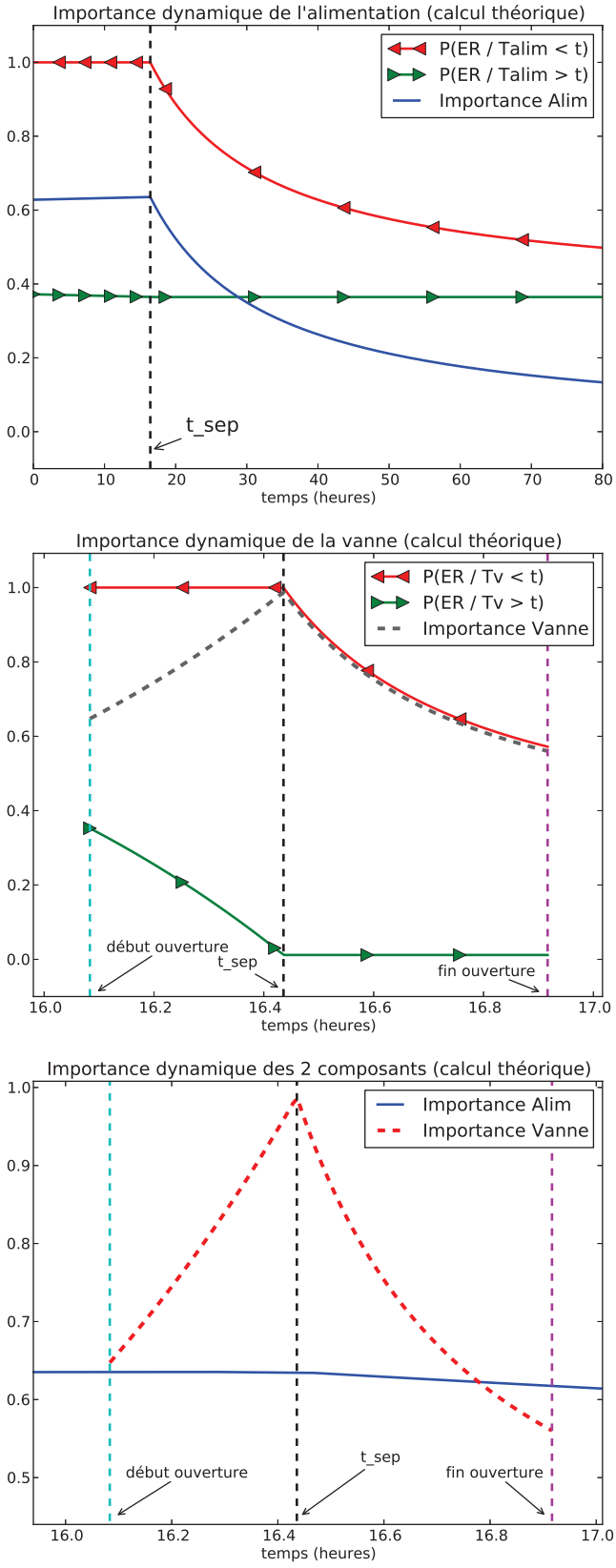

FIGURE 6.1 – Calcul analytique de l'importance dynamique pour l'alimentation et la vanne. Comparaison des importances dynamiques pour ces deux composants.

systématique de c ». D'autre part, l'événement $\{T_c > t_{deb}\}$ est toujours réalisé. D'où

$$I_c(t_{deb}) = P(ER/def_c) - P(ER). \tag{6.19}$$

Cela revient à dire que $I_c(t_{deb})$ mesure l'augmentation de la probabilité d'occurrence de l'événement redouté que représente une défaillance à la sollicitation systématique du composant c par rapport à la situation normale.

6.4.2 Importance à la fin de la mission du composant

Soit t_{fin} l'instant de la fin de la mission du composant c. Estimer l'importance dynamique du composant c à la date t_{fin} revient à écrire $I_c(t_{fin}) = P(ER/T_c \leq t_{fin}) - P(ER/T_c > t_{fin})$. D'une part, une défaillance de c a toujours lieu avant t_{fin} donc $P(ER/T_c \leq t_{fin}) = P(ER)$. D'autre part une défaillance du composant c après t_{fin} revient à considérer que le composant c est parfait. Soit $non\ def_c$ l'événement élémentaire « c parfait ». D'où

$$I_c(t_{fin}) = P(ER) - P(ER/non\ def_c). \tag{6.20}$$

Cela revient à dire que $I_c(t_{fin})$ mesure la diminution de la probabilité d'occurrence de l'événement redouté que représente un remplacement du composant c par un composant parfait par rapport à la situation normale.

6.4.3 Allure de la courbe

La figure 6.1 illustre le calcul analytique de l'importance dynamique pour l'alimentation et la vanne et la comparaison des importances dynamiques pour ces deux composants.

L'importance dynamique de l'alimentation s'écrit

$$I_{alim}(t) = \begin{cases} (1-\gamma_V)e^{-\lambda_V(t_{sep}-t_{ouv})}\exp\left(-\lambda_{alim}(t_{sep}-t)\right) & \text{si } t \leq t_{sep} \\ \frac{(1-\gamma_V)e^{-\lambda_V(t_{sep}-t_{ouv})}\left[1-e^{-\lambda_{alim}t_{sep}}\right]}{1-e^{-\lambda_{alim}t}} & \text{si } t > t_{sep} \end{cases}.$$

L'importance dynamique de la vanne s'écrit

$$I_V(t) = \begin{cases} e^{-\lambda_{alim}t_{sep}}\exp\left(-\lambda_V(t_{sep}-t)\right) & \text{si } t \leq t_{sep} \\ \frac{\left[1+\gamma_V-(1-\gamma_V)e^{\lambda_V(t_{sep}-t_{ouv})}\right]e^{-\lambda_{alim}t_{sep}}}{\gamma_V+(1-\gamma_V)\left(1-e^{\lambda_V(t-t_{ouv})}\right)} & \text{si } t > t_{sep} \end{cases}.$$

Cela se résume sous la forme $I_{alim}(t) = \begin{cases} \alpha_1^{alim}\exp\left(\lambda_{alim}t\right) & \text{si } t \leq t_{sep} \\ \frac{\alpha_2^{alim}}{\beta_2^{alim}-\gamma_2^{alim}e^{-\lambda_{alim}t}} & \text{si } t > t_{sep} \end{cases}$ et $I_V(t) = \begin{cases} \alpha_1^{V}\exp\left(\lambda_V t\right) & \text{si } t \leq t_{sep} \\ \frac{\alpha_2^{V}}{\beta_2^{V}-\gamma_2^{V}e^{-\lambda_V t}} & \text{si } t > t_{sep} \end{cases}$. Finalement, les deux courbes paraissent différentes

sur la figure 6.1 mais l'écriture de l'importance dynamique du composant c se généralise sous la forme

$$I_c(t) = \begin{cases} \alpha_1^c \exp(\lambda_c t) & \text{si } t \leq t_{sep} \\ \frac{\alpha_2^c}{\beta_2^c - \gamma_2^c e^{-\lambda_c t}} & \text{si } t > t_{sep} \end{cases}. \tag{6.21}$$

A la différence de l'alimentation, l'allure de l'importance dynamique pour la vanne donne une impression de « pic ». Cette différence s'explique par

1. la largeur de la durée de la mission de l'alimentation (du début de la crue à l'infini) par rapport à la durée de mission de la vanne (50 minutes) ;
2. les valeurs utilisées pour les taux de défaillance de l'alimentation et de la vanne : $\lambda_{alim} = 7,2 \times 10^{-4} \ll \lambda_V = 1,2$.

6.4.4 Comparaison avec l'importance dynamique obtenue à partir des histoires simulées

L'estimation de l'importance dynamique à partir des 350000 histoires disponibles est effectuée sur toute la durée de la crue, en divisant cette période en 10000 intervalles de temps. Sur la figure 6.2, les triangles représentent des valeurs prises par S_{infER} et S_{supER} (numérateurs des probabilités conditionnelles) et les étoiles représentent des valeurs prises par S_{inf} et S_{sup} (dénominateurs des probabilités conditionnelles). Les courbes en trait continu représentent l'évolution des probabilités $P(ER/T_c \leq t)$ (en rouge) et $P(ER/T_c > t)$ (en vert). Ces éléments graphiques permettent de comprendre l'évolution de l'importance dynamique de chaque composant, dont l'allure correspond à celle issue du calcul théorique, représentée sur la figure 6.1.

La figure 6.3 confirme le bon accord entre la théorie et les simulations. Le fastidieux calcul analytique peut donc être remplacé par l'estimation de l'importance dynamique à partir des simulations. De plus, l'observation du graphique obtenu permet d'apprendre que l'instant t_{sep} correspond au maximum de chacune des courbes et s'obtient donc par lecture graphique, où t_{sep} est l'instant avant lequel la défaillance de la vanne entrainera systématiquement l'événement redouté, et après lequel le débit sortant sera suffisant pour évacuer la crue.

L'analyse des histoires par la méthode des coupes prépondérantes révèle que 96,79 % des ER sont causés par une défaillance de la vanne, alors que 3,08 % des ER sont causés par une défaillance de l'alimentation. Ces résultats semblent à première vue contradictoires avec l'allure des courbes. L'alimentation parait en effet beaucoup plus « importante » sur le graphe.

Cela s'explique par le fait que $I_c(t_{deb})$ est de la forme $\alpha \exp(-\lambda_c t_{sep})$, où c désigne l'alimentation ou la vanne. Ainsi, l'importance d'un composant au début de sa mission est une fonction décroissante de son taux de défaillance. Cette explication est confirmée en considérant que $I_c(t_{deb})$ mesure l'augmentation de la probabilité d'occurrence de

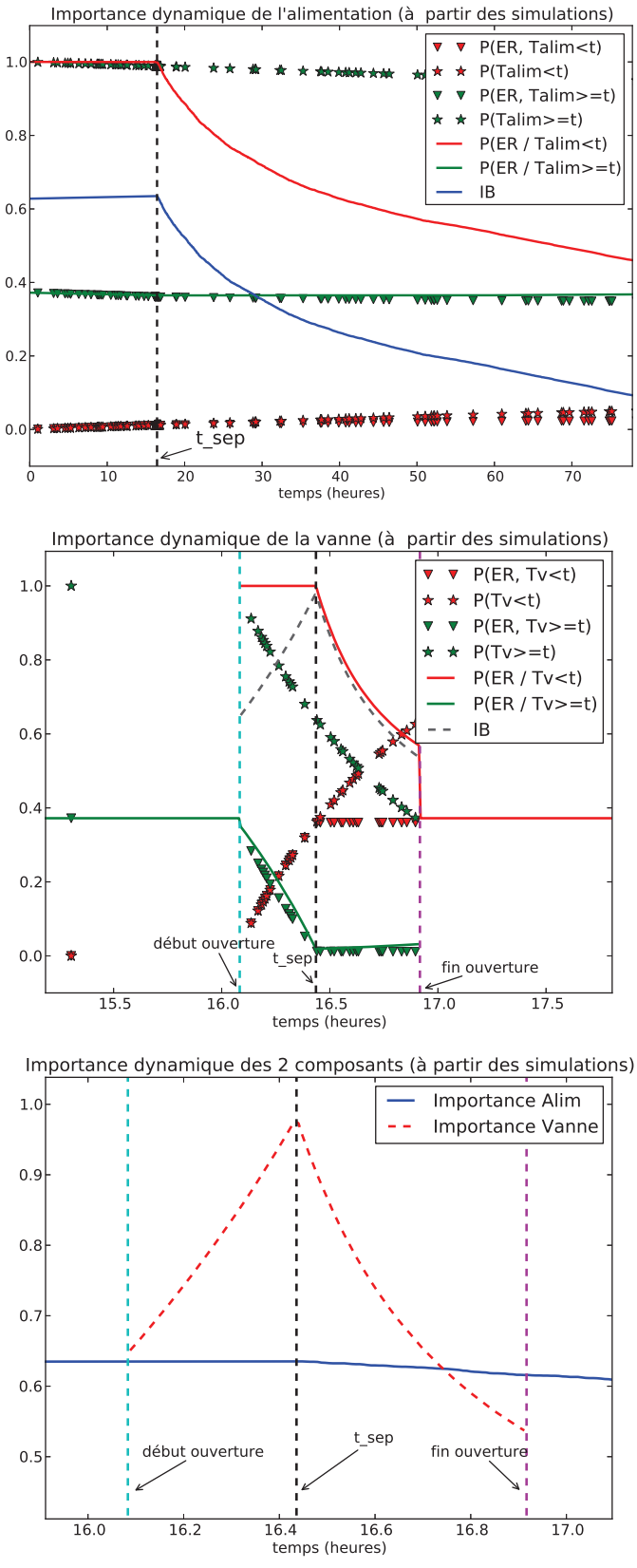

FIGURE 6.2 – Importance dynamique obtenue à partir des simulations, par comptage des histoires, pour l'alimentation et pour la vanne

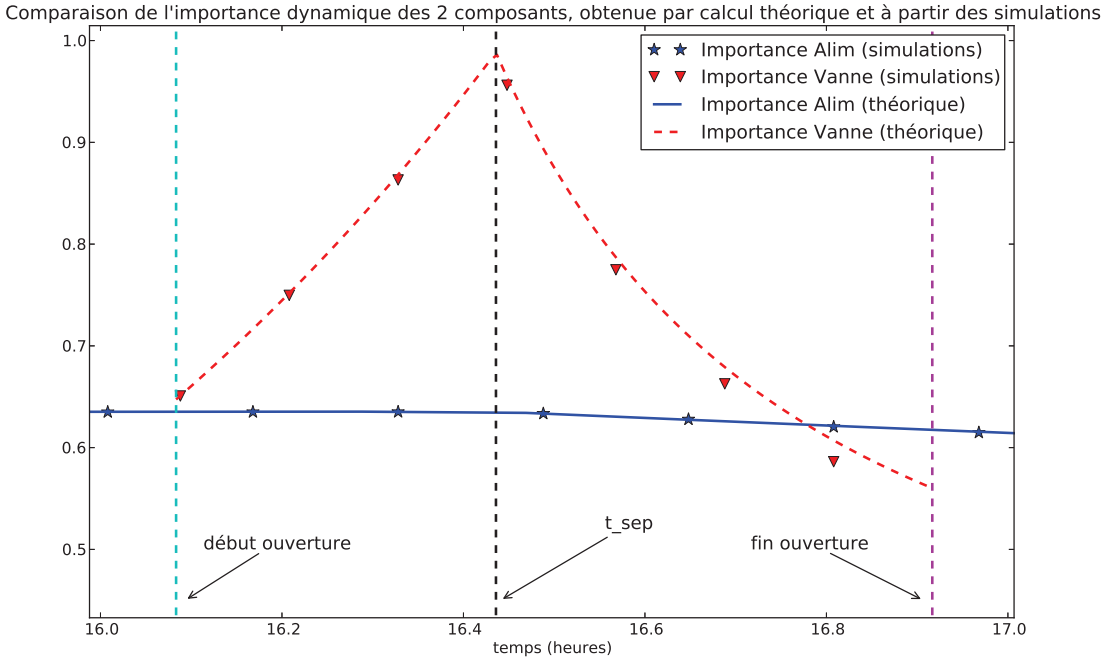

FIGURE 6.3 – Comparaison des importances dynamiques de l'alimentation et de la vanne obtenues par calcul analytique et à partir des simulations

l'événement redouté que représente une défaillance systématique et immédiate du composant c par rapport à la situation normale. Si λ_c augmente, la probabilité d'occurrence de l'ER en situation normale augmente également, et $I_c(t_{deb})$ diminue.

Par ailleurs, ce n'est pas parce qu'un composant est important qu'il y a nécessairement beaucoup de défaillances impliquant ce composant. En effet, des probabilités conditionnelles interviennent dans les calculs. Un composant est important si la probabilité de l'événement redouté augmente fortement en cas de panne de ce composant. En revanche, ce composant important peut être très fiable et ne provoquer que très rarement l'ER.

Confrontation avec l'importance statique

L'importance statique du composant c s'obtient à partir de $I_c(t_{deb})$ et $I_c(t_{fin})$ où t_{deb} et t_{fin} sont respectivement les dates de début et de fin de la mission du composant c. En effet

$I_c = P(ER/def_c) - P(ER/non\ def_c)$
$= [P(ER/def_c) - P(ER)] + [P(ER) - P(ER/non\ def_c)]$
$= I_c(t_{deb}) + I_c(t_{fin})$.

L'importance dynamique proposée généralise donc bien les mesures classiques d'importance statique.

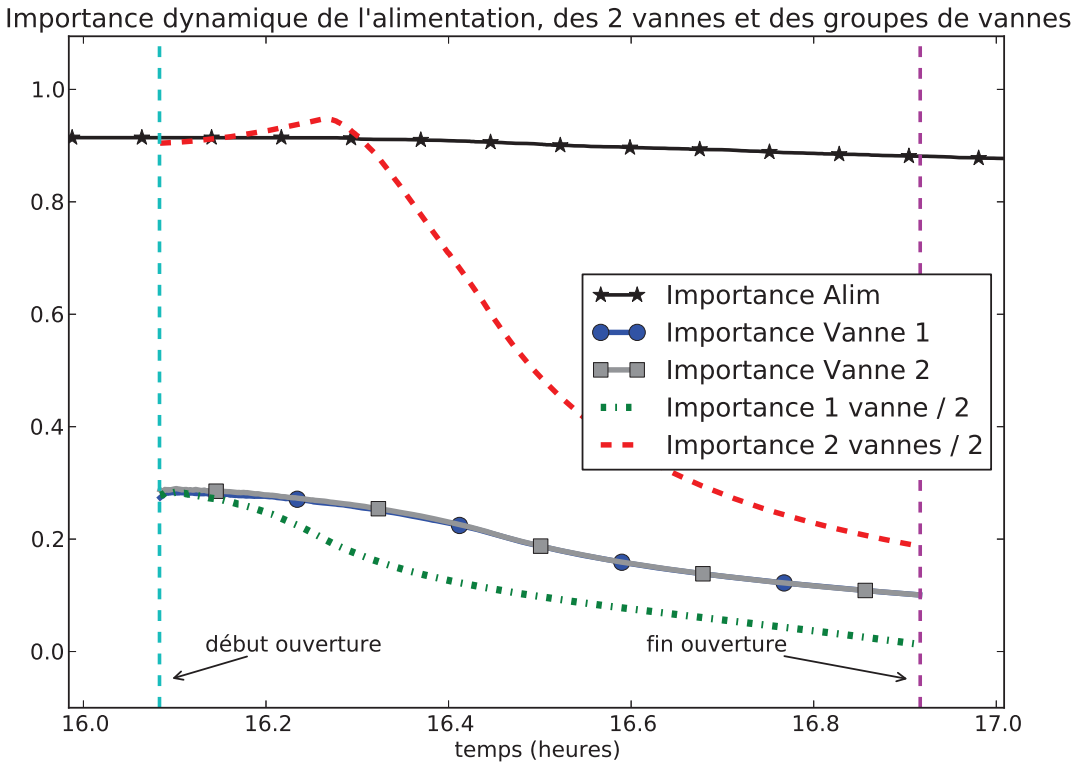

FIGURE 6.4 – Importance dynamique des composants du système FR1

Pour conclure, la définition d'un unique indicateur représentatif de l'importance dynamique sur toute la durée de la crue n'aurait pas de sens. Inutile donc par exemple de calculer l'aire sous les courbes de chaque composant. En revanche, l'estimation de l'importance dynamique nous permet d'identifier quel est le composant le plus important à chaque instant.

6.4.5 Application aux systèmes « Fil Rouge »

6.4.5.1 Système composé d'une alimentation et de deux vannes

Le système FR1 considéré est à présent constitué d'une alimentation et de deux vannes potentiellement défaillantes. L'importance dynamique de chacun des trois composants va être estimée à partir des histoires simulées.

L'estimation de l'importance dynamique du système précédent a indiqué que la fenêtre de temps située entre la $16^{ème}$ et la $17^{ème}$ heure de la crue était la plus intéressante. C'est donc cette période seulement qui sera analysée pour les deux prochains systèmes. En effet, l'ouverture des vannes est toujours programmé sur la même période, quelque soit le système considéré.

Une première analyse des histoires consistait en l'identification des coupes regroupées prépondérantes. Une coupe regroupée prépondérante désigne un groupe de composants qui représente un grand nombre d'ER, si tous les composants de ce groupe sont défaillants lors de la même histoire. Comment confronter les informations fournies par ces coupes regroupées prépondérantes à celles obtenues par l'estimation de l'importance dynamique ? Pour le savoir, il est intéressant d'estimer aussi les importances liées aux défaillances des groupes « 1 vanne sur 2 » et « 2 vannes sur 2 ».

L'instant de défaillance du groupe « 1 vanne sur 2 » est assimilé à celui de la première défaillance de vanne de chaque histoire. Cela revient à supposer que la deuxième défaillance n'aura pas lieu avant la fin de l'ouverture. Plus t est grand, moins cette hypothèse est forte.

L'instant de défaillance du groupe « 2 vannes sur 2 » est assimilé à celui de la deuxième défaillance de vanne de chaque histoire. Cela implique une perte d'information sur l'instant de la permière défaillance, mais pas d'hypothèse sur l'occurrence de celle-ci.

D'après la figure 6.4, la défaillance d'une seule vanne ne provoque d'ER que si elle a lieu pendant le premiers tiers de son ouverture, ce qui explique les valeurs peu élevées des importances dynamiques des deux vannes prises séparément (traits continus bleus et gris avec cercles et carrés). En effet, l'estimation de l'importance de chacune des deux vannes à l'instant t se fait indépendamment de l'instant de défaillance de l'autre vanne par rapport à t. L'importance du groupe « 1 vanne sur 2 » à l'instant t est d'autant plus basse que son calcul se base sur l'hypothèse qu'une éventuelle deuxième défaillance de vanne a lieu après t.

En revanche, le calcul de l'importance du groupe « 2 vannes sur 2 » à l'instant t suppose que la première des deux défaillances a eu lieu avant t, ce qui explique les valeurs plus élevées de l'importance de ce groupe.

Finalement, ces résultats sont cohérents avec les coupes équivalentes prépondérantes qui indiquent que 87,42 % des ER sont causés par la défaillance des deux vannes, alors que la défaillance de l'alimentation provoque 12,34 % de cet événement indésirable et que la défaillance d'une seule vanne ne représente que 0,09 % des ER. Toutefois, cette cohérence n'était pas évidente car ce sont des types d'informations complémentaires. En effet, un composant important mais très fiable peu n'être impliqué que dans peu d'histoires observées dont l'issue est l'ER.

Enfin, l'identification de l'équivalent de t_{sep} est possible par lecture graphique. La figure 6.4 montre qu'une défaillance des deux vannes ou de l'alimentation après 16,25 heures ne provoque plus de débordement, ce qui est confirmé par l'étude des histoires.

6.4.5.2 Système composé d'une alimentation et de six vannes

Le système FR2 considéré est constitué d'une alimentation et de six vannes potentiellement défaillantes. L'importance dynamique de chacun des sept composants va

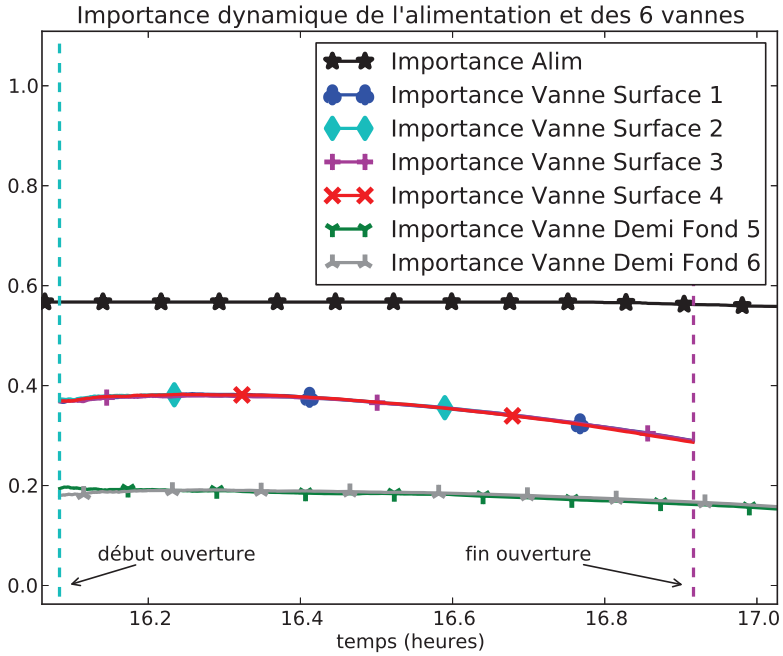

FIGURE 6.5 – Importance dynamique de chaque composant du système FR2, obtenue à partir des simulations

être estimée à partir des histoires simulées. C'est aussi le cas de l'importance de chaque groupe de composants, afin de comparer avec les coupes regroupées prépondérantes.

La figure 6.5 apporte peu d'informations, puisque chaque vanne prise indépendamment des autres a peu de chance de provoquer à elle seule un ER. Il est toutefois possible de savoir que les vannes de surface sont plus importantes que les vannes de demi-fond pendant leur processus d'ouverture. Une lecture graphique précise indique que l'importance dynamique de l'alimentation est décroissante à partir de 16,67 heures.

La figure 6.6 permet de tirer les enseignements suivants :

- Les quatre groupes les plus importants sont
  - « 2 vannes de surface sur les quatre » (trait continu vert avec trèfles),
  - « 2 vannes de surface sur les quatre + 1 vanne de demi-fond sur les deux » (traits continus verts avec losanges),
  - « 3 vannes de surface sur les quatre » (trait continu gris avec croix),
  - « 3 vannes de surface sur les quatre + 1 vanne de demi-fond sur les deux » (trait continu gris avec "+").

  En réalité, la coupe équivalente prépondérante est « 3 vannes de surface sur les quatre + 1 vanne de demi-fond sur les deux ». Elle provoque 22,56 % des ER. Or c'est la plus basse des quatre courbes. Cela s'explique par le fait que les histoires dans lesquelles ce groupe est présent sont aussi des histoires comptées dans l'estimation de l'importance des trois autres groupes. En effet, soit une

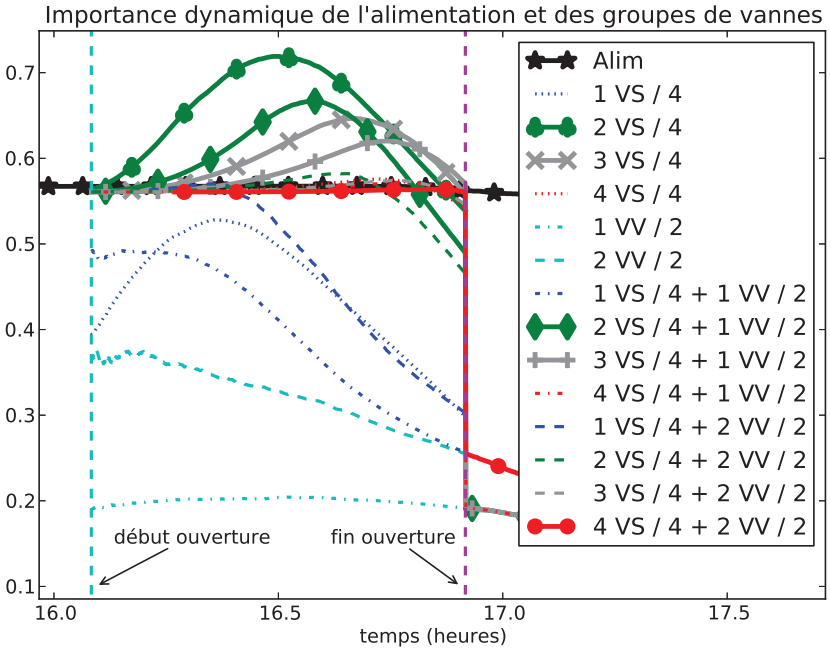

FIGURE 6.6 – Importance dynamique de chaque groupe de composants du système FR2 obtenue à partir des simulations

histoire k où l'événement E_{3+1} survient à l'instant t. L'événement E_{i+j} désigne l'union des événements « perte de la $i^{ème}$ vanne de surface sachant que j vannes de demi-fond sont en panne » et « perte de la $j^{ème}$ vanne de demi-fond sachant que i vannes de surface sont en panne ». L'histoire k a forcément vu se réaliser les instants E_{2+0}, E_{2+1} et E_{3+0} avant t. Et les défaillances de deux ou trois vannes sont suffisantes pour provoquer l'ER, surtout si elles ont eu lieu tôt dans le processus d'ouverture. C'est ce qui explique pourquoi la courbe correspondant à la coupe regroupée prépondérante n'est pas la courbe la plus haute, mais aussi le décalage dans le temps entre les maximums des différentes courbes.

– L'allure de l'importance dynamique du groupe « 4 vannes de surface sur les quatre + 2 vannes de demi-fond sur les deux », est similaire à celle de l'alimentation pendant le processus d'ouverture, ce qui est logique puisque ces deux événements ont pour effet l'arrêt de l'ouverture de toutes les vannes.

Conclusion

L'estimation de l'importance dynamique permet donc de savoir à tout instant quel est le composant dont la défaillance à cet instant précis aurait le plus d'impact sur la probabilité de l'événement indésirable, par rapport à la situation normale. Cette définition est généralisable à un groupe de composants. Cette analyse indique également

l'importance statique des composants et l'instant après lequel la dernière défaillance d'un groupe de composants ne provoque plus l'événement redouté.

La lecture graphique de ces résultats est aisée pour un système de dimension modeste. Cependant, un système de taille industrielle sera associé à un trop grand nombre de courbes. Il sera donc préférable d'afficher un classement des composants (ou des groupes de composants) les plus importants à certaines dates de la crue.

Nous avons donc proposé une définition originale d'importance dynamique, généralisant les facteurs d'importance statique, qui prend explicitement en compte l'information temporelle. Il est en effet possible de retrouver les facteurs d'importance statique à partir des mesures d'importance dynamiques proposées par le biais d'un calcul simple. Enfin, la lecture graphique permet d'identifier un instant t_{sep} séparant les histoires en deux sous-ensembles car cette date est caractérisée par une inversion de la courbe de l'importance dynamique.

Conclusion générale et perspectives

Dans le cadre de l'estimation de la sûreté de fonctionnement, nous avons proposé une démarche complète, de la modélisation à la synthèse d'indicateurs fiabilistes pour l'aide à la décision avec prise en compte de l'information temporelle.

Les ouvrages de production d'électricté hydraulique ont la particularité de dépendre d'événements aléatoires discrets, mais aussi de l'évolution d'une variable déterministe continue. A ce titre, ce sont des systèmes dynamiques hybrides (SDH). Un modèle de fiabilité dynamique permet de représenter les interactions entre des événements stochastiques discrets et des variables déterministes continues. L'événement redouté (ER) est réalisé lorsque le niveau de la retenue atteint un seuil de sûreté.

Dans un premier temps, un état de l'art des différentes méthodes utilisées en fiabilité dynamique a été réalisé. L'objectif était d'identifier une méthode capable d'une part de décrire des SDH de taille industrielle, d'autre part de tirer profit des variables temporelles en termes d'indicateurs. Les Automates Stochastiques Hybrides (ASH) distribués ont été choisis pour décrire les SDH suite à un arbitrage sur des critères tels que la taille du système à modéliser, la lisibilité de la représentation graphique, la complexité des interactions et l'existence d'un outil associé. Ce formalisme inclut le calcul de l'évolution des variables déterministes continues, dont les paramètres sont réajustés à chaque événement aléatoire. Une fois la description du système réalisée, l'objectif est d'exploiter le modèle afin de faire des analyses quantitatives. PyCATSHOO, outil en cours de développement et destiné à être open source, permet de décrire par des ASH des systèmes complexes de dimension importante, puis y associe la simulation de Monte Carlo pour l'exploitation quantitative des modèles développés. Par ailleurs, l'état de l'art constate que la variable temporelle n'est pas exploitée autant qu'elle pourrait l'être lors de la quantification des résultats.

La modélisation des SDH avec PyCATSHOO repose sur trois niveaux de programmation. Ces travaux ont apporté des contributions aux deuxième et troisième niveaux. Le premier niveau est la conception du logiciel à proprement parler. Le deuxième niveau est l'élaboration d'une Base de Connaissances (BdC) dédiée à la description d'une catégorie de systèmes donnée, qui détaille les caractéristiques de chaque type de composant. Après l'analyse fonctionnelle d'une classe de système et la formalisation d'hypothèses de modélisation, la création d'une BdC repose sur la définition d'états, de transitions et de boîtes à messages qui participent à la synchronisation des automates. La complexité

de la modélisation dépend de la complexité du système étudié, mais pas de la dimension du système. En effet, un SDH est rendu complexe par la variété des comportements que peuvent avoir les différents composants, mais aussi par le nombre et la nature des interactions entre les composants et par la considération de la dynamique d'une variable déterministe continue. La variété des interactions augmente le nombre de boîtes à messages à définir dans la BdC. Le nombre de classes dans cette base dépend de la variété des composants, mais pas de leur quantité. La croissance du système, en nombre de composants, implique donc l'ajout de nouveaux automates, mais pas l'explosion des automates déjà présents.

Le troisième niveau de programmation de PyCATSHOO est le modèle décrivant un système en particulier, le suivi de la simulation de Monte Carlo et le traitement des résultats. L'analyse des résultats fournit des indicateurs de fiabilité classique, tels que l'évolution de la probabilité d'occurrence de l'ER par rapport au temps. Les coupes équivalentes prépondérantes diagnostiquent les combinaisons d'événements les plus contributeurs dans la réalisation de l'ER. Des leviers d'amélioration de la fiabilité peuvent ainsi être identifiés.

Les résultats des simulations sont appelés « histoires ». Une histoire est la séquence de tous les états activés, ainsi que la date de cette activation, lors du passage de l'algorithme pendant une simulation. La complexité du système est caractérisée dans les résultats par le nombre et la longueur des histoires. Nous avons proposé une méthode pour extraire, synthétiser et utiliser de nouvelles informations temporelles dans l'évaluation des performances fiabilistes. Cette démarche était rarement associée à la fiabilité dynamique, aussi les indicateurs proposés dans cette thèse sont-ils innovants. L'ER est différé par rapport au temps de défaillance des composants. Ce délai correspond au temps nécessaire à la montée de l'eau dans le réservoir jusqu'au seuil de sûreté. Certaines notions de sûreté comme celle de coupe ou de mesure d'importance ont été redéfinies.

Nous avons proposé une démarche prévisionnelle fondée sur la classification des histoires. Cette classification est à l'origine de la caractérisation d'une fonction f des dates de défaillance (T_1, ..., T_n) des n composants, qui pronostique l'issue de l'histoire associée. Ce modèle exploite au maximum les données temporelles contenues dans les histoires simulées.

Enfin, l'estimation de l'importance dynamique permet de savoir à tout instant quel est le composant dont la défaillance à cet instant précis aurait le plus d'impact sur la probabilité de l'ER, par rapport à une situation de référence. Cette définition est généralisée à un groupe de composants qui contribuent le plus à l'ER et à quel moment.

Les étapes de cette démarche sont appliquées et exposées sur un réservoir simple, puis cet exemple illustratif a été progressivement étoffé. Les EdC constituent un support et une illustration pour ces travaux mais ne sont pas les seuls systèmes susceptibles d'être concernés par la prise en compte de l'information temporelle en fiabilité dynamique. La méthodologie proposée est adaptable à d'autres domaines industriels, condi-

tionnellement à l'élaboration d'une BdC spécifique à la nouvelle catégorie de systèmes étudiée. C'est possible dans le contexte nucléaire, et plus particulièrement dans le cadre des Etudes Probabilistes de Sûreté (EPS) de niveau 2, dont l'objectif est d'évaluer la nature, l'importance et les fréquences des rejets hors de l'enceinte de confinement, suite à un scénario accidentel de type « fusion du cœur ».

Perspectives

Les travaux réalisés au cours de cette thèse ont ouvert plusieurs pistes de recherche et de développement.

Les outils numériques de la plate-forme PyCATSHOO gagneraient à être améliorés. Des techniques existent pour accélérer la simulation de Monte Carlo mais ne sont pas forcément adaptées aux systèmes dynamiques. Une nouvelle thèse est envisagée pour traiter cette question. Le nombre de messages échangés entre les classes de la BdC doit être optimisé lors de la conception de celle-ci, surtout si des boîtes à messages sont synchronisées les unes par rapport aux autres. Si PyCATSHOO permet de paralléliser les simulations sur plusieurs processeurs et réduire ainsi les temps de calcul, ce n'est pas encore le cas pour le traitement des résultats.

Il serait possible d'enrichir encore la BdC afin de lever les hypothèses de modélisation évoquées dans la section 4.1.6. En particulier, l'abaissement du plan d'eau en prévision de l'arrivée de la crue, ou le respect de la contrainte « laminer la crue », pourraient être modélisés de façon plus réaliste, en optimisant le calcul de la débitance des vannes et donc leur degré d'ouverture. Modéliser le processus de fermeture des vannes impliquerait aussi d'enrichir les automates des actionneurs concernés, avec la possibilité de transmettre des ordres contradictoires.

L'exploitation des résultats pourrait également être révisée. La lisibilité d'indicateurs comme les coupes prépondérantes ou l'importance dynamique dépend du regroupement de composants similaires. Il serait intéressant de proposer une méthodologie de regroupement des composants pour les systèmes de dimension importante. Le traitement des histoires a été effectué et exposé dans le cas de systèmes non réparables. La prise en compte de réparations augmenterait la longueur des histoires et leur classification en fonction des dates de défaillance serait à redéfinir.

Les SDH présentés dans ces travaux sont des systèmes cohérents dans le sens où une panne supplémentaire ne rétablit pas le fonctionnement d'un système en panne, et où une réparation ne provoque pas la panne d'un système en état de marche. Considérer des systèmes non cohérents impliquerait de revoir la validation de certains indicateurs proposés, notamment la classification des histoires. En effet, la frontière associée à la classification d'histoires caractérisant des systèmes non cohérents pourrait avoir des propriétés de convexité différentes de celles caractérisant les systèmes cohérents.

Enfin, des pistes de réflexion ont été dégagées pour envisager l'élaboration d'un module d'aide à la décision plus rapide et plus efficace, afin d'assister l'opérateur dans la gestion d'une situation d'urgence. Dans ce cas, des simulations off-line fourniraient également un outil de pronostic plus léger que la mise en œuvre de PyCATSHOO. Cet outil donnerait le risque associé à une situation donnée et permettrait même de comparer le risque entre différentes actions possibles de l'opérateur (réparation d'un composant plutôt qu'un autre, mise en œuvre d'une voie de secours, etc.). Parallèlement à ce suivi des risques (de l'anglais *risk monitoring*), une analyse de risques système permettrait de quantifier les risques liés à l'exploitation du système.

Le développement d'une Interface Homme-Machine (IHM) améliorerait grandement l'accessibilité de PyCATSHOO. La visualisation simultanée de la structure globale d'un SDH et des automates qui modélisent chaque classe de composant n'est pas encore possible. Un support interactif permettrait, à partir de l'architecture d'un SDH, de cliquer sur le composant d'intérêt pour afficher son automate. Cela faciliterait le développement et la validation du modèle sur d'autres systèmes. Le développement de cette IHM ainsi que celui d'un support interactif sont à présent conditionnés par un projet de portage informatique du code Python vers C++, qui est l'un des langages de programmation les plus populaires permettant de construire des applications sur une grande variété de plateformes matérielles et de systèmes d'exploitation.

Suite à ces travaux de thèse, une phase d'appropriation va être entreprise par les chercheurs du département MRI d'EDF R&D. Parmi les indicateurs de fiabilité proposés, un tri va être effectué en fonction de leur intérêt pour l'utilisateur final. D'autres applications sont prévues, notamment dans le domaine des EPS dans le contexte nucléaire ; les EPS de niveau 2 seront concernées à long terme. Actuellement, un projet de recherche post-doctorale est en cours de lancement dans le cadre des EPS de niveau 1 (consacrées à la quantification du risque de fusion du cœur). L'objectif est d'étudier le comportement d'un dispositif d'évacuation de puissance résiduelle DHR (de l'anglais *Decay Heat Removal*) pour un projet de réacteur à neutrons rapides refroidi au sodium ASTRID (de l'anglais *Advanced Sodium Technological Reactor for Industrial Demonstration*), en utilisant l'outil PyCATSHOO. Ce système de refroidissement présente les propriétés d'un SDH dans les différents circuits qui le composent et la variation des températures du liquide s'y écoulant.

Annexe A

Déroulement de l'algorithme de PyCATSHOO

Cette section décrit l'enchainement de l'algorithme déroulé par PyCATSHOO. Le paramètre `maxTimeStep` (qui désigne l'intervalle de temps sur lequel est calculée l'évolution du niveau) est fixé à 1000 secondes. Les dates présentes dans cette chronologie dépendent de `maxTimeStep` et de l'atteinte de seuil par le niveau.

1. A partir de l'état **`surveillance`** de la crue, PyCATSHOO vérifie si les conditions pour réaliser la transition **`surveillance`**→**`détection`** sont réunies. Dans ce cadre, les méthodes `Crue.calculDelais()` puis `Crue.mustChange()` sont appelées.
2. La méthode `Crue.mustChange()` appelle `Crue.sendSignalToReservoir()`.
3. Ce signal est réceptionné par `Reservoir.getSignalFromCrue()`.
4. La méthode `Reservoir.getSignalFromCrue()` appelle `Reservoir.sendStartToCrue()`.
5. Ce signal est réceptionné par `Crue.getStartFromReservoir()`.
6. Retour à la méthode `Crue.mustChange()`, qui retourne **`True`** après réception du signal par `Crue.getStartFromReservoir()`.
7. Transition **`surveillance`**→**`détection`**.
8. L'arrivée dans l'état **`détection`** appelle la méthode `Crue.sendDiffNiveauToReservoir()`.
9. Cette méthode est synchronisée avec `Vanne.sendDiffNiveauToReservoir()`.
10. Ces informations sont réceptionnées par `Reservoir.getDiffNiveauFromAmont()`.
11. Mise à jour des paramètres de résolution de l'équation différentielle de la variable déterministe continue (niveau dans le réservoir) du PDMP.
12. Résolution de l'équation différentielle du niveau, sur l'intervalle [0, 1000]. Envoi du niveau actuel par la méthode `Reservoir.sendNiveauToCapteur()`. Répétition de cette étape jusqu'à l'intervalle [3000, 4000].

13. Pendant cet intervalle, la transition **détection→phase Veille** est réalisée à la date de début de la phase de veille, à t =3600 secondes.
14. L'arrivée dans l'état **phase Veille** appelle la méthode `Crue.sendDiffNiveauToReservoir()`.
15. Cette méthode est synchronisée avec `Vanne.sendDiffNiveauToReservoir()`.
16. Ces informations sont réceptionnées par `Reservoir.getDiffNiveauFromAmont()`.
17. Mise à jour des paramètres de résolution de l'équation différentielle de la variable déterministe continue (niveau dans le réservoir) du PDMP.
18. Résolution de l'équation différentielle du niveau, sur l'intervalle [3600, 4600]. Envoi du niveau actuel par la méthode `Reservoir.sendNiveauToCapteur()`. Répétition de cette étape jusqu'à l'intervalle [56600, 57600].
19. La transition **phase Veille→phase Crue** est réalisée à la date de début de la phase de crue, à t =57600 secondes.
20. L'arrivée dans l'état **phase Crue** appelle la méthode `Crue.sendPhaseToAmont()`.
21. Cette information est réceptionnée par `Vanne.getPhaseFromCrue()`.
22. L'arrivée dans l'état **phase Crue** appelle la méthode `Crue.sendDiffNiveauToReservoir()`.
23. Cette méthode est synchronisée avec `Vanne.sendDiffNiveauToReservoir()`.
24. Ces informations sont réceptionnées par `Reservoir.getDiffNiveauFromAmont()`.
25. Mise à jour des paramètres de résolution de l'équation différentielle de la variable déterministe continue (niveau dans le réservoir) du PDMP.
26. Résolution de l'équation différentielle du niveau, sur l'intervalle [57600, 58600]. Envoi du niveau actuel par la méthode `Reservoir.sendNiveauToCapteur()`.
27. L'arrivée de l'information de l'établissement de la crue valide les conditions de la transition **closed→opening** de la crue, à t =57900 secondes.
28. La date du début de l'ouverture de la vanne est mémorisée par la méthode `Vanne.mise_a_jour_t_ouv()`.
29. L'arrivée dans l'état **opening** appelle la méthode `Vanne.constructCurrentStateConfiguration()`, puis `Vanne.sendDiffNiveauToReservoir()`.
30. Cette méthode est synchronisée avec `Crue.sendDiffNiveauToReservoir()`.
31. Ces informations sont réceptionnées par `Reservoir.getDiffNiveauFromAmont()`.
32. Mise à jour des paramètres de résolution de l'équation différentielle de la variable déterministe continue (niveau dans le réservoir) du PDMP.
33. Résolution de l'équation différentielle du niveau, sur l'intervalle [57900, 58900]. Envoi du niveau actuel par la méthode `Reservoir.sendNiveauToCapteur()`. Répétition de cette étape jusqu'à l'intervalle [59900, 60900].

34. L'ouverture de la vanne est complète et la transition **opening→open** est réalisée, à t =60900 secondes.
35. L'arrivée dans l'état **open** appelle la méthode `Vanne.constructCurrentStateConfiguration()`, puis `Vanne.sendDiffNiveauToReservoir()`.
36. Cette méthode est synchronisée avec `Crue.sendDiffNiveauToReservoir()`.
37. Ces informations sont réceptionnées par `Reservoir.getDiffNiveauFromAmont()`.
38. Mise à jour des paramètres de résolution de l'équation différentielle de la variable déterministe continue (niveau dans le réservoir) du PDMP.
39. Résolution de l'équation différentielle du niveau, sur l'intervalle [60900, 61900]. Envoi du niveau actuel par la méthode `Reservoir.sendNiveauToCapteur()`. Répétition de cette étape jusqu'à l'intervalle [236900, 237900].
40. La date de fin de la cue est atteinte et la transition **SSNA→Fin Crue** est réalisée, à t =237600 secondes.
41. Fin de l'histoire.

PyCASTHOO stocke l'enchaînement des transitions de cette simulation et initialise une nouvelle simulation. Cet enchaînement de transitions est appelé histoire. L'histoire retournée pour cette simulation est de la forme
[(0, Reservoir, **SSNA**),
(0, Vanne, **closed**),
(0, Crue, **avant**),
(0, Crue, **après**),
(3600, Crue, **phase Veille**),
(57600, Crue, **phase Crue**),
(57900, Vanne, **opening**),
(60900, Vanne, **open**),
(237600, Reservoir, **Fin Crue**)].

Annexe B

Démonstrations du chapitre 5

B.1 Instant d'atteinte du seuil de sûreté en fonction du temps de défaillance

L'atteinte du seuil de sûreté par le niveau, qui constitue l'événement redouté (ER), a lieu avec un certain retard par rapport à l'instant de défaillance. Ce délai est lié au dimensionnement du réservoir : atteindre le niveau h_{max} à partir du niveau h_0 n'est pas immédiat.

Pour chaque instant de panne $u \geq 0$, l'instant d'occurrence de l'ER s'écrit

$$t_{ER}(u) = \frac{I_c \times t_{c_0} - (h_0 + h_{max})\, S - \frac{q_{max}}{d_{ouv}} t_{v_0} u - \frac{q_{max}}{2 \times d_{ouv}} u^2}{I_c - \frac{q_{max}}{d_{ouv}} u}.$$

Démonstration. A cet instant de l'événement redouté, $h(t_{ER}) = h_{max}$. L'expression de $h(t)$ pour $t \in \left[t_{v_0} + u\,;\, t_{c_f}\right]$ s'écrit

$$h(t) = h(t_{v_0} + u) + \frac{I_c}{S} \times (t - (t_{v_0} + u)) - \frac{q_{sor}(t_{v_0} + u)}{S} \times (t - (t_{v_0} + u))$$

avec

$$h(t_{v_0} + u) = h_0 + \frac{I_c}{S} \times (t_{v_0} + u - t_{c_0}) - \frac{q_{max}}{2 \times d_{ouv} \times S} (t_{v_0} + u - t_{v_0})^2$$

et

$$q_{sor}(t_{v_0} + u) = \frac{(t_{v_0} + u - t_{v_0}) \times q_{max}}{d_{ouv}}$$

d'où

$$h(t_{ER}) = h_0 + \frac{I_c}{S} \times (t_{v_0} + u - t_{c_0}) - \frac{q_{max}}{2 \times d_{ouv} \times S} u^2 + \frac{I_c}{S} \times (t_{ER} - (t_{v_0} + u))$$

$$
\begin{aligned}
&-\frac{q_{max}}{d_{ouv}S}u\left(t_{ER}-(t_{v_0}+u)\right)=h_{max}\\
\Leftrightarrow\ & h_0-h_{max}+\frac{I_c}{S}\times(t_{ER}-t_{c_0})+\frac{q_{max}}{2\times d_{ouv}S}u^2-\frac{q_{max}}{d_{ouv}S}u\left(t_{ER}-t_{v_0}\right)=0\\
\Leftrightarrow\ & t_{ER}\times\left(I_c-\frac{q_{max}}{d_{ouv}}u\right)+(h_0+h_{max})\,S-I_c\times t_{c_0}+\frac{q_{max}}{2\times d_{ouv}}u^2+\frac{q_{max}}{d_{ouv}}t_{v_0}u=0\\
\Leftrightarrow\ & t_{ER}(u)=\frac{I_c\times t_{c_0}-(h_0+h_{max})\,S-\frac{q_{max}}{d_{ouv}}t_{v_0}u-\frac{q_{max}}{2\times d_{ouv}}u^2}{I_c-\frac{q_{max}}{d_{ouv}}u}.
\end{aligned}
$$

□

B.2 Expression de $P_{ER}(t)$

$$
P_{ER}(t)=\begin{cases}0 & \text{si } t\leq t_{ER_0}\\ p_{soll}+(1-p_{soll})\left(1-e^{-\lambda u_{ER}(t)}\right) & \text{sinon}\end{cases}
$$

Démonstration. Pour $t\leq t_{ER_0}$, $\mathbb{P}\left(h(t)>h_{max}\right)=0$ car t_{ER_0} correspond à l'instant de l'ER lorsque la vanne est défaillante à la sollicitation. Avant t_{ER_0}, la physique du barrage ne permet aucun événement redouté.

Soit p_{soll} la probabilité de défaillance à la sollicitation de la vanne et λ le paramètre de la loi exponentielle qui caractérise la durée de fonctionnement de la vanne avant défaillance, notée *panne*.

Pour $t\geq t_{ER_0}$,

$$
\begin{aligned}
\mathbb{P}\left(h(t)>h_{max}\right)&=p_{soll}\underbrace{\mathbb{P}\left(h(t)>h_{max}/panne=0\right)}_{1}\\
&+(1-p_{soll})\int_0^{\infty}\mathbb{P}\left(h(t)>h_{max}/panne=u\right)\lambda e^{-\lambda u}du\\
&=p_{soll}+(1-p_{soll})\int_0^{\infty}\mathbb{P}\left(h(t)>h_{max}/panne=u\right)\lambda e^{-\lambda u}du.
\end{aligned}
$$

Or $\mathbb{P}\left(h(t)>h_{max}/panne=u\right)=1$ si $t\geq t_{ER}(u)\Leftrightarrow t_{ER}^{-1}(t)\geq u\Leftrightarrow u_{ER}(t)\geq u$.

D'où

$$
\begin{aligned}
\mathbb{P}\left(h(t)>h_{max}\right)&=p_{soll}+(1-p_{soll})\int_0^{u_{ER}(t)}\lambda e^{-\lambda u}du\\
&=p_{soll}+(1-p_{soll})\left(1-e^{-\lambda u_{ER}(t)}\right).
\end{aligned}
$$

Finalement,

$$
P_{ER}(t)=\begin{cases}0 & \text{si } t\leq t_{ER_0}\\ p_{soll}+(1-p_{soll})\left(1-e^{-\lambda u_{ER}(t)}\right) & \text{sinon}\end{cases}.
$$

□

B.3 Réservoir vidangé par deux vannes : calcul analytique de la frontière $u_2^{sep}(u_1)$

La fonction u_2^{sep} est définie sur $[0,\, d_{ouv}]$ et retourne le TTF de la vanne 2 correspondant à l'occurrence de l'événement redouté à l'instant de fin de crue, sachant que le TTF de la vanne 1 est u_1 :

$$u_2^{sep}(u_1) = \Big(t_{c_f} - t_{v_0} - u_1\Big)$$

$$+\sqrt{\Big(t_{c_f} - t_{v_0} - u_1\Big)^2 - 2\left\{\frac{d_{ouv}}{q_{max}}\Big[S(h_0 - h_{max}) + I_c(t_{c_f} - t_{c_0})\Big] - u_1^2 - (t_{c_f} - t_{v_0})u_1\right\}}. \tag{B.1}$$

Cette frontière est illustrée par la figure 5.12.

Démonstration. Pour ce calcul $u_1 \leq u_2$. L'évolution du niveau s'écrit

$$h(t) = \begin{cases} h_0 + \frac{I_c}{S}(t - t_{c_0}) - \frac{q_{max}}{2\times d_{ouv}\times S}(t - t_{v_0})^2 & \text{si } t \in [t_{v_0};\, t_{v_0} + u_1]\,, \\ h(t_{v_0} + u_1) + \frac{I_c}{S} \times (t - (t_{v_0} + u_1)) - \frac{q_{sor}(t_{v_0}+u_1)}{S} \times (t - (t_{v_0} + u_1)) & \\ \qquad - \frac{q_{max}}{2\times d_{ouv}\times S}(t - (t_{v_0} + u_1))^2 & \text{si } t \in [t_{v_0} + u_1\,;\, t_{v_0} + u_2]\,, \\ h(t_{v_0} + u_2) + \frac{I_c}{S} \times (t - (t_{v_0} + u_2)) - \frac{q_{sor}(t_{v_0}+u_1)}{S} \times (t - (t_{v_0} + u_2)) & \\ \qquad - \frac{q_{sor}(t_{v_0}+u_2)}{S} \times (t - (t_{v_0} + u_2)) & \text{si } t \in \Big[t_{v_0} + u_2\,;\, t_{c_f}\Big]\,, \end{cases} \tag{B.2}$$

avec $q_{sor}(t_{v_0} + u) = \frac{q_{max}}{d_{ouv}}u$.

Donc $h(t_{c_f}) = h_{max}$

$\Leftrightarrow \quad h_0 + \frac{I_c}{S}(t_{c_f} - t_{c_0}) - \frac{q_{max}}{d_{ouv}S}\left(\frac{u_1^2 - u_2^2}{2} - u_1u_2 + (t_{c_f} - t_{v_0})(u_1 + u_2)\right) = h_{max}$
$\Leftrightarrow \frac{q_{max}}{2d_{ouv}}u_2^2 + \frac{q_{max}}{d_{ouv}}u_1u_2 - \frac{q_{max}}{d_{ouv}}(t_{c_f} - t_{v_0})u_2$
$= S\,(h_{max} - h_0) - I_c(t_{c_f} - t_{c_0}) - \frac{q_{max}}{2d_{ouv}}u_1^2 - \frac{q_{max}}{d_{ouv}}(t_{v_f} - t_{v_0})u_1.$

À partir de la résolution de cette équation du second degré, la fonction u_2^{sep} est définie sur $[0,\, d_{ouv}]$ niveetourne le TTF de la vanne 2 correspondant à l'occurrence de l'événement redouté à l'instant de fin de crue, sachant que le TTF de la vanne 1 est u_1 :

$$u_2^{sep}(u_1) = \Big(t_{c_f} - t_{v_0} - u_1\Big)$$

$$+\sqrt{\Big(t_{c_f} - t_{v_0} - u_1\Big)^2 - 2\left\{\frac{d_{ouv}}{q_{max}}\Big[S(h_0 - h_{max}) + I_c(t_{c_f} - t_{c_0})\Big] - u_1^2 - (t_{c_f} - t_{v_0})u_1\right\}}$$

□

Annexe C

Démonstrations du chapitre 6

C.1 Démonstration de la proposition 6.2

Proposition. *Soit A et B deux événements. Soit C un événement et $\bar{C}$ son complémentaire. Alors*

$$P(A/B) = P(A/B,\, C) \times P(C/B) + P(A/B,\, \bar{C}) \times P(\bar{C}/B)$$

Démonstration. $P(A/B) = \frac{P(A,B)}{P(B)} = \frac{1}{P(B)} \left[P(A,\, B,\, C) + P(A,\, B,\, \bar{C})\right]$

$= \frac{1}{P(B)} \left[P(A/B,\, C) \times P(B,\, C) + P(A/B,\, \bar{C}) \times P(B,\, \bar{C})\right]$

$= P(A/B\,, C) \times \frac{P(B,C)}{P(B)} + P(A/B\,, \bar{C}) \times \frac{P(B,\bar{C})}{P(B)}$

$= P(A/B,\, C) \times P(C/B) + P(A/B,\, \bar{C}) \times P(\bar{C}/B)$ □

C.2 Expression littérale de l'importance dynamique de Birnbaum pour l'alimentation

C.2.1 Calcul de $P(ER/T_{alim} \leq t)$

– Pour $t > t_{sep}$

$$P(ER/T_{alim} \leq t) \tag{C.1}$$

$$= \frac{F_{alim}(t_{sep}) + \left[(1-\gamma_V)F_V(t_{sep} - t_{ouv}) + \gamma_V\right]\left[F_{alim}(t) - F_{alim}(t_{sep})\right]}{F_{alim}(t)}.$$

Démonstration. $P(ER/T_{alim} \leq t) =$
$P\left(ER/T_{alim} \leq t,\ \min(T_{alim},\, T_V) \leq t_{sep}\right) \times P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_{alim} \leq t\right)$
$+P\left(ER/T_{alim} \leq t,\ \min(T_{alim},\, T_V) > t_{sep}\right) \times P\left(\min(T_{alim},\, T_V) > t_{sep}/T_{alim} \leq t\right).$

Une défaillance de l'alimentation ou de la vanne avant t_{sep} implique systématiquement l'ER donc $P\left(ER/T_{alim} \leq t,\ \min(T_{alim},\, T_V) \leq t_{sep}\right) = 1.$

La bonne marche de l'alimentation et de la vanne jusque t_{sep} empêche tout ER donc $P\left(ER/T_{alim} \leq t,\ \min(T_{alim},\, T_V) > t_{sep}\right) = 0.$

Finalement, $P(ER/T_{alim} \leq t) = P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_{alim} \leq t\right).$

Or $P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_{alim} \leq t\right) = \frac{P(\min(T_{alim}, T_V) \leq t_{sep},\, T_{alim} \leq t)}{P(T_{alim} \leq t)}.$

Au numérateur, $P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} \leq t\right)$
$= \int_0^t P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} \leq t/T_{alim} = x\right) f_{alim}(x)dx$
$= \int_0^{t_{sep}} \underbrace{P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} \leq t/T_{alim} = x\right)}_{1} f_{alim}(x)dx$
$+ \underbrace{\int_{t_{sep}}^{t} P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} \leq t/T_{alim} = x\right) f_{alim}(x)dx}_{P(T_V \leq t_{sep})\ (T_{alim} \text{ et } T_V \text{ indépendantes})}$
$= F_{alim}(t_{sep}) + \int_{t_{sep}}^{t} \left[(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V\right] f_{alim}(x)dx$
$= F_{alim}(t_{sep}) + \left[(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V\right]\left[F_{alim}(t) - F_{alim}(t_{sep})\right].$

Au dénominateur, $P\left(T_{alim} \leq t\right) = F_{alim}(t).$

Finalement, pour tout $t > t_{sep}$,

$$P(ER/T_{alim} \leq t)$$
$$= \frac{F_{alim}(t_{sep}) + \left[(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V\right]\left[F_{alim}(t) - F_{alim}(t_{sep})\right]}{F_{alim}(t)}.$$

□

C.2.2 Calcul de $P(ER/T_{alim} > t)$

– Pour $t \leq t_{sep}$

$$P(ER/T_{alim} > t) \tag{C.2}$$

$$= \frac{\left[F_{alim}(t_{sep}) - F_{alim}(t)\right] + \left[(1-\gamma_V)F_V(t_{sep}-t_{ouv}) + \gamma_V\right]\left[1 - F_{alim}(t_{sep})\right]}{1 - F_{alim}(t)}.$$

Démonstration. $P(ER/T_{alim} > t) =$
$P\left(ER/T_{alim} > t,\ \min(T_{alim},\, T_V) \leq t_{sep}\right) \times P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_{alim} > t\right)$
$+P\left(ER/T_{alim} > t,\ \min(T_{alim},\, T_V) > t_{sep}\right) \times P\left(\min(T_{alim},\, T_V) > t_{sep}/T_{alim} > t\right).$

Une défaillance de l'alimentation ou de la vanne avant t_{sep} implique systématiquement l'ER donc $P\left(ER/T_{alim} > t,\ \min(T_{alim},\, T_V) \leq t_{sep}\right) = 1$.

La bonne marche de l'alimentation et de la vanne jusque t_{sep} empêche tout ER donc $P\left(ER/T_{alim} > t,\ \min(T_{alim},\, T_V) > t_{sep}\right) = 0$.

Finalement, $P(ER/T_{alim} > t) = P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_{alim} > t\right)$.

Or $P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_{alim} > t\right) = \frac{P(\min(T_{alim}, T_V) \leq t_{sep},\ T_{alim} > t)}{P(T_{alim} > t)}$.

Au numérateur, $P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} > t\right)$
$= \int_t^{\infty} P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} > t/T_{alim} = x\right) f_{alim}(x)dx$
$= \int_t^{t_{sep}} \underbrace{P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} > t/T_{alim} = x\right)}_{1} f_{alim}(x)dx$
$+ \int_{t_{sep}}^{\infty} \underbrace{P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_{alim} > t/T_{alim} = x\right)}_{P(T_V \leq t_{sep})\ (T_{alim} \text{ et } T_V \text{ indépendantes})} f_{alim}(x)dx$
$= \left[F_{alim}(t_{sep}) - F_{alim}(t)\right] + \int_{t_{sep}}^{\infty} \left[(1-\gamma_V)F_V(t_{sep} - t_{ouv}) + \gamma_V\right] f_{alim}(x)dx$
$= \left[F_{alim}(t_{sep}) - F_{alim}(t)\right] + \left[(1-\gamma_V)F_V(t_{sep} - t_{ouv}) + \gamma_V\right]\left[1 - F_{alim}(t_{sep})\right].$

Au dénominateur, $P\left(T_{alim} > t\right) = 1 - F_{alim}(t)$.

Finalement, pour tout $t \leq t_{sep}$,

$$P(ER/T_{alim} > t)$$
$$= \frac{\left[F_{alim}(t_{sep}) - F_{alim}(t)\right] + \left[(1-\gamma_V)F_V(t_{sep} - t_{ouv}) + \gamma_V\right]\left[1 - F_{alim}(t_{sep})\right]}{1 - F_{alim}(t)}.$$

□

C.3 Expression littérale de l'importance dynamique de Birnbaum pour la vanne

C.3.1 Calcul de $P(ER/T_V \leq t)$

– Pour $t > t_{sep}$

$$P(ER/T_V \leq t) \tag{C.3}$$
$$= \frac{\gamma_V + (1-\gamma_V)\left[F_V(t_{sep} - t_{ouv}) + F_{alim}(t_{sep})\left[F_V(t - t_{ouv}) - F_V(t_{sep} - t_{ouv})\right]\right]}{\gamma_V + (1-\gamma_V)F_V(t - t_{ouv})}.$$

Démonstration. $P(ER/T_V \leq t) =$
$P\left(ER/T_V \leq t,\ \min(T_{alim},\ T_V) \leq t_{sep}\right) \times P\left(\min(T_{alim},\ T_V) \leq t_{sep}/T_V \leq t\right)$
$+P\left(ER/T_V \leq t,\ \min(T_{alim},\ T_V) > t_{sep}\right) \times P\left(\min(T_{alim},\ T_V) > t_{sep}/T_V \leq t\right).$

Une défaillance de l'alimentation ou de la vanne avant t_{sep} implique systématiquement l'ER donc $P\left(ER/T_V \leq t,\ \min(T_{alim},\ T_V) \leq t_{sep}\right) = 1$.

La bonne marche de l'alimentation et de la vanne jusque t_{sep} empêche tout ER donc $P\left(ER/T_V \leq t,\ \min(T_{alim},\ T_V) > t_{sep}\right) = 0$.

Finalement, $P(ER/T_V \leq t) = P\left(\min(T_{alim},\ T_V) \leq t_{sep}/T_V \leq t\right)$.

Or $P\left(\min(T_{alim},\ T_V) \leq t_{sep}/T_V \leq t\right) = \frac{P(\min(T_{alim}, T_V) \leq t_{sep},\ T_V \leq t)}{P(T_V \leq t)}$.

Au numérateur, $P\left(\min(T_{alim},\ T_V) \leq t_{sep},\ T_V \leq t\right)$
$= \gamma_V + (1-\gamma_V) \int_{t_{ouv}}^{t} P\left(\min(T_{alim},\ T_V) \leq t_{sep},\ T_V \leq t/T_V = x\right) f_V(x - t_{ouv})dx$
$= \gamma_V + (1-\gamma_V) \left[\int_{t_{ouv}}^{t_{sep}} \underbrace{P\left(\min(T_{alim},\ T_V) \leq t_{sep},\ T_V \leq t/T_V = x\right)}_{1} f_V(x - t_{ouv})dx \right.$
$\left. + \int_{t_{sep}}^{t} \underbrace{P\left(\min(T_{alim},\ T_V) \leq t_{sep},\ T_V \leq t/T_V = x\right)}_{P(T_{alim} \leq t_{sep})\ (T_{alim} \text{ et } T_V \text{ indépendantes})} f_V(x - t_{ouv})dx \right]$
$= \gamma_V + (1-\gamma_V) \left[F_V(t_{sep} - t_{ouv}) + \int_{t_{sep}}^{t} F_{alim}(t_{sep}) f_V(x - t_{ouv})dx \right]$
$= \gamma_V + (1-\gamma_V) \left[F_V(t_{sep} - t_{ouv}) + F_{alim}(t_{sep}) \left[F_V(t - t_{ouv}) - F_V(t_{sep} - t_{ouv}) \right] \right].$

Au dénominateur, $P\left(T_V \leq t\right) = \gamma_V + (1-\gamma_V) F_V(t - t_{ouv})$.

Finalement, pour tout $t > t_{sep}$,

$$P(ER/T_V \leq t)$$
$$= \frac{\gamma_V + (1-\gamma_V) \left[F_V(t_{sep} - t_{ouv}) + F_{alim}(t_{sep}) \left[F_V(t - t_{ouv}) - F_V(t_{sep} - t_{ouv}) \right] \right]}{\gamma_V + (1-\gamma_V) F_V(t - t_{ouv})}.$$

□

C.3.2 Calcul de $P(ER/T_V > t)$

– Pour $t \leq t_{sep}$

$$P(ER/T_V > t) \tag{C.4}$$
$$= \frac{\left[F_V(t_{sep} - t_{ouv}) - F_V(t - t_{ouv}) \right] + F_{alim}(t_{sep}) \left[1 - F_V(t_{sep} - t_{ouv}) \right]}{1 - F_V(t - t_{ouv})}.$$

Démonstration. $P(ER/T_V > t) =$
$P\left(ER/T_V > t,\ \min(T_{alim},\, T_V) \leq t_{sep}\right) \times P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_V > t\right)$
$+P\left(ER/T_V > t,\ \min(T_{alim},\, T_V) > t_{sep}\right) \times P\left(\min(T_{alim},\, T_V) > t_{sep}/T_V > t\right).$

Une défaillance de l'alimentation ou de la vanne avant t_{sep} implique systématiquement l'ER donc $P\left(ER/T_V > t,\ \min(T_{alim},\, T_V) \leq t_{sep}\right) = 1$.

La bonne marche de l'alimentation et de la vanne jusque t_{sep} empêche tout ER donc $P\left(ER/T_V > t,\ \min(T_{alim},\, T_V) > t_{sep}\right) = 0$.

Finalement, $P(ER/T_V > t) = P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_V > t\right)$.

Or $P\left(\min(T_{alim},\, T_V) \leq t_{sep}/T_V > t\right) = \frac{P(\min(T_{alim}, T_V) \leq t_{sep},\, T_V > t)}{P(T_V > t)}$.

Au numérateur, $P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_V > t\right)$
$= \int_t^{\infty} P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_V > t/T_V = x\right) f_V(x - t_{ouv})dx$
$= \int_t^{t_{sep}} \underbrace{P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_V > t/T_V = x\right)}_{1} f_V(x - t_{ouv})dx$
$+ \int_{t_{sep}}^{\infty} \underbrace{P\left(\min(T_{alim},\, T_V) \leq t_{sep},\ T_V > t/T_V = x\right)}_{P(T_{alim} \leq t_{sep})\ (T_{alim} \text{ et } T_V \text{ indépendantes})} f_V(x - t_{ouv})dx$
$= \left[F_V(t_{sep} - t_{ouv}) - F_V(t - t_{ouv})\right] + \int_{t_{sep}}^{\infty} F_{alim}(t_{sep}) f_V(x - t_{ouv})dx$
$= \left[F_V(t_{sep} - t_{ouv}) - F_V(t - t_{ouv})\right] + F_{alim}(t_{sep})\left[1 - F_V(t_{sep} - t_{ouv})\right].$

Au dénominateur, $P\left(T_V > t\right) = 1 - F_V(t - t_{ouv})$.

Finalement, pour tout $t \leq t_{sep}$,

$$P(ER/T_V > t)$$
$$= \frac{\left[F_V(t_{sep} - t_{ouv}) - F_V(t - t_{ouv})\right] + F_{alim}(t_{sep})\left[1 - F_V(t_{sep} - t_{ouv})\right]}{1 - F_V(t - t_{ouv})}.$$

□

Bibliographie

[EDF, 2011] (2011). La production d'électricité d'origine hydraulique. http ://energie.edf.com/fichiers/fckeditor/Commun/En_Direct_Centrales/Nucleaire /General/Notes_Info/note_energie_hydraulique.pdf.

[pyt, 2013] (2013). Python. http ://www.python.org ; http ://www.numpy.scipy.org ; http ://www.scipy.org ; http ://www.simpy.sourceforge.net.

[Adolfsson *et al.*, 2012] Adolfsson, Y., Holmberg, J.-E., Karanta, I. et Kudinov, P. (2012). Proceedings of the IDPSA-2012 : Integrated deterministic-probabilistic safety analysis workshop November 2012. Rapport technique VTT-R-08589-12, Stockholm, Sweden.

[Aldemir, 2012] Aldemir, T. (2012). A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants. *Annals of Nuclear Energy*, 52:113–124.

[Aldemir *et al.*, 2007] Aldemir, T., Stovsky, M., Kirschenbaum, J., Mandelli, D., Bucci, P., Mangan, L., Miller, D., Sun, X., Ekici, E., Guarro, S. *et al.* (2007). *Dynamic reliability modeling of digital instrumentation and control systems for nuclear reactor probabilistic risk assessments*. Numéro NUREG/CR-6942.

[Aubry *et al.*, 2012] Aubry, J.-F., Babykina, G., Barros, A., Brinzei, N., Deleuze, G., de Saporta, B., Dufour, F., Langeron, Y. et Zhang, H. (2012). Rapport final du projet APPRODYN : APPROches de la fiabilité DYNamique pour modéliser des systèmes critiques. Rapport technique.

[Baptiste et Bournez, 2009] Baptiste, P. et Bournez, O. (2009). Programmation et Algorithmique. Polycopié du cours INF 421, Ecole Polytechnique.

[Barker *et al.*, 2006] Barker, M., Vivian, B. et Bowles, D. S. (2006). Reliability assessement for a spillway gate upgrade design in Queensland, Australia. *In United States Society on Dams Conference*.

[Birnbaum, 1968] Birnbaum, Z. W. (1968). On the importance of different components in a multicomponent system. Rapport technique, DTIC Document.

[Bouissou, 2007] Bouissou, M. (2007). Comparison of two Monte Carlo schemes for simulating Piecewise Deterministic Markov Processes. *In Proceedings of Mathematical Methods in Reliability MMR*, Glasgow, UK.

[Bouissou et Bon, 2003] Bouissou, M. et Bon, J.-L. (2003). A new formalism that combines advantages of fault-trees and Markov models : Boolean logic driven Markov processes. *Reliability Engineering & System Safety*, 82(2):149–163.

[Bouissou et Bourreau, 2012] Bouissou, M. et Bourreau, B. (2012). Revue des applications des réseaux bayésiens dynamiques en analyse des risques. *In Congrès Lambda-Mu 18*, Tours, France.

[Brandejsky, 2012] Brandejsky, A. (2012). *Méthodes numériques pour les processus markoviens déterministes par morceaux*. Thèse de doctorat, Université Bordeaux 1.

[Brinzei *et al.*, 2009] Brinzei, N., Perez Castaneda, G. A. et Aubry, J.-F. (2009). Sûreté de fonctionnement prévisionnelle en contexte dynamique. *In 2ème Workshop Surveillance, Sûreté et Sécurité des Grands Systèmes, 3SGS'09*, Nancy, France.

[Brissaud, 2010] Brissaud, F. (2010). *Contributions à la Modélisation et à l'Évaluation de la Sûreté de Fonctionnement de Systèmes de Sécurité à Fonctionnalités Numériques*. Thèse de doctorat, Université de Technologie de Troyes.

[Brissaud *et al.*, 2012] Brissaud, F., Barros, A. et Bérenguer, C. (2012). Probability of failure on demand of safety systems : impact of partial test distribution. *Proceedings of the Institution of Mechanical Engineers, Part O : Journal of Risk and Reliability*, 226(4):426–436.

[Broy *et al.*, 2011a] Broy, P., Chraibi, H. et Donat, R. (2011a). Using Dynamic Bayesian Networks to solve a dynamic reliability problem. *In Annual Conference of the European Safety and Reliability Association, ESREL 2011*, pages 335–341, Troyes, France.

[Broy *et al.*, 2013] Broy, P., Chraibi, H., Donat, R., Bérenguer, C. et Dijoux, Y. (2013). A new methodology to model and assess reliability of large dynamic hybrid systems. *In Proceedings of Mathematical Methods in Reliability MMR*, pages 28–32, Stellenbosch, South Africa.

[Broy *et al.*, 2011b] Broy, P., Donat, R., Chraibi, H., Dijoux, Y. et Bérenguer, C. (2011b). Dynamic Bayesian Networks for assessing reliability of hybrid systems. *In Proceedings of Mathematical Methods in Reliability MMR*, pages 252–257, Beijing, China.

[Cabarbaye et Etienne, 2010] Cabarbaye, A. et Etienne, K. (2010). Dimensionnement probabiliste et optimisation des systèmes par des modèles de simulation hybrides. *In Congrès Lambda-Mu 17*, La Rochelle, France.

[Čepin et Mavko, 2002] Čepin, M. et Mavko, B. (2002). A dynamic fault tree. *Reliability Engineering & System Safety*, 75(1):83–91.

[Chabot *et al.*, 1998] Chabot, J., Ducamp, F., Signoret, J., Joulain, P. et Hutinet, T. (1998). Hybrid Monte-Carlo simulation using Petri-nets and fire code for analysing fire scenarios in nuclear power plants. *In Proceedings of the Fourth International Conference on Probabilistic Safety Assessment and Management (PSAM IV), New York*, volume 2, pages 867–71.

[Chabot, 1998] Chabot, J.-L. (1998). *Approche probabiliste relative à l'étude des scénarios d'incendie*. Thèse de doctorat, Université de Poitiers.

[Chang et Lin, 2011] Chang, C.-C. et Lin, C.-J. (2011). LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

[Chaux et Deleuze, 2010] Chaux, P.-Y. et Deleuze, G. (2010). Comparaison de deux méthodes dynamiques d'évaluation de la sûreté de fonctionnement : BDMP et DFM. *In Congrès Lambda-Mu 17*, La Rochelle, France.

[Chraibi, 2013a] Chraibi, H. (2013a). Dynamic reliability modeling and assessment with PyCATSHOO : Application to a test case. *In PSAM Topical Conference*, Tokyo, Japan.

[Chraibi, 2013b] Chraibi, H. (2013b). GASPART : Manuel de référence. Note technique EDF R&D. Rapport technique, H-T51-2013-02138-FR.

[Cocozza-Thivent, 1997] Cocozza-Thivent, C. (1997). *Processus stochastiques et fiabilité des systèmes*, volume 28. Springer.

[Cocozza-Thivent, 2012] Cocozza-Thivent, C. (2012). *Processus de renouvellement markovien. Processus de Markov déterministes par morceaux*. http ://perso-math.univ-mlv.fr/users/cocozza.christiane/recherche-pageperso/RMetPDMP.pdf.

[Cocozza-Thivent *et al.*, 2006a] Cocozza-Thivent, C., Desgrouas, M. et Mercier, S. (2006a). Algorithme de calcul de disponibilité asymptotique en fiabilité dynamique. *In Congrès Lambda-Mu 15*, Lille, France.

[Cocozza-Thivent *et al.*, 2004] Cocozza-Thivent, C., Eymard, R. et Mercier, S. (2004). Méthodes et algorithmes de fiabilité dynamique pour la quantification de petits systèmes redondants. *In Congrès Lambda-Mu 14*, pages 498–505, Bourges, France.

[Cocozza-Thivent *et al.*, 2006b] Cocozza-Thivent, C., Eymard, R. et Mercier, S. (2006b). A finite-volume scheme for dynamic reliability models. *IMA journal of numerical analysis*, 26(3):446–471.

[Cocozza-Thivent *et al.*, 2006c] Cocozza-Thivent, C., Eymard, R., Mercier, S. et Roussignol, M. (2006c). Characterization of the marginal distributions of Markov processes used in dynamic reliability. *International Journal of Applied Mathematics and Stochastic Analysis*, 2006:1–18.

[Colby *et al.*, 2012] Colby, J. B., Rudie, J. D., Brown, J. A., Douglas, P. K., Cohen, M. S. et Shehzad, Z. (2012). Insights into multimodal imaging classification of ADHD. *Frontiers in systems neuroscience*, 6(59).

[Cristianini et Shawe-Taylor, 2000] Cristianini, N. et Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

[Davis, 1993] Davis, M. (1993). *Markov models and optimization, volume 49 of Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

[Davis, 1984] Davis, M. H. (1984). Piecewise-deterministic Markov processes : A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 353–388.

[De Saporta *et al.*, 2010] De Saporta, B., Dufour, F. et Gonzalez, K. (2010). Numerical method for optimal stopping of piecewise deterministic Markov processes. *The Annals of Applied Probability*, 20(5):1607–1637.

[Do Van, 2008] Do Van, P. (2008). *Contribution au développement et à l'étude de facteurs d'importance fiabilistes pour les systèmes markoviens*. Thèse de doctorat, Université de Technologie de Troyes.

[Donat, 2009] Donat, R. (2009). *Modélisation de la fiabilité et de la maintenance par modèles graphiques probabilistes*. Thèse de doctorat, Institut National des Sciences Appliquées de Rouen.

[Donat *et al.*, 2010] Donat, R., Leray, P., Bouillaut, L. et Aknin, P. (2010). A dynamic bayesian network to represent discrete duration models. *Neurocomputing*, 73(4):570–577.

[Duflot, 2007] Duflot, N. (2007). *Les mesures d'importance fiabilistes issues des études probabilistes de sûreté nucléaires : contrôle des incertitudes et nouvelles applications pour l'aide à la décision*. Thèse de doctorat, Université de Technologie de Troyes.

[Dugan *et al.*, 1992] Dugan, J. B., Bavuso, S. J. et Boyd, M. A. (1992). Dynamic fault-tree models for fault-tolerant computer systems. *Reliability, IEEE Transactions on*, 41(3):363–377.

[Dutuit *et al.*, 1997] Dutuit, Y., Chatelet, E., Signoret, J.-P. et Thomas, P. (1997). Dependability modelling and evaluation by using stochastic Petri nets : application to two test cases. *Reliability Engineering & System Safety*, 55(2):117–124.

[Estes *et al.*, 2005] Estes, A. C., Foltz, S. D. et McKay, D. T. (2005). Estimating Risk from Spillway Gate Systems on Dams Using Condition Assessment Data. Rapport technique, DTIC Document.

[Faghihi, 2012] Faghihi, F. (2012). Dynamic Probabilistic Risk Analysis of the Fast Cascade Phase of Large Disturbances in Power System. *In Proc. of the 11th International Probabilistic Safety Assessment and Management Conference & the Annual European Safety and Reliability Conference - PSAM 11/ESREL 2012*, pages 586–595, Helsinki, Finlande. Curran Associates, Inc.

[Flori et Donat, 2011] Flori, A. et Donat, R. (2011). Manuel utilisateur de KB3 V3. Note technique EDF R&D. Rapport technique, H-T52-2011-01722-FR.

[Garavaglia, 2011] Garavaglia, F. (2011). *Méthode SCHADEX de prédétermination des crues extrêmes*. Thèse de doctorat, Université de Grenoble.

[Ghostine, 2008] Ghostine, R. (2008). *Influence des fautes transitoires sur la fiabilité d'un système commandé en réseau*. Thèse de doctorat, Institut National Polytechnique de Lorraine-INPL.

[Glasserman, 2004] Glasserman, P. (2004). *Monte Carlo methods in financial engineering*, volume 53. Springer.

[Gnouma, 2006] Gnouma, R. (2006). *Aide à la calibration d'un modèle hydrologique distribué au moyen d'une analyse des processus hydrologiques : application au bassin versant de l'Yzeron*. Thèse de doctorat, Institut National des sciences Appliquées de Lyon.

[GRIF, 2012] GRIF (2012). GRIF 2012. Réseaux de Petri à prédicats. Manuel utilisateur. Version 27 Mars 2012. Rapport technique, Total.

[Hartford et Baecher, 2004] Hartford, D. N. et Baecher, G. B. (2004). *Risk and uncertainty in dam safety*. Inst of Civil Engineers Pub.

[Hasan et Boris, 2006] Hasan, M. et Boris, F. (2006). SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges. Polycopié du cours BD Web, ISTY3, Université de Versailles St Quentin.

[Henzinger, 2000] Henzinger, T. A. (2000). *The theory of hybrid automata*. Springer.

[Izquierdo et Labeau, 2004] Izquierdo, J. et Labeau, P.-E. (2004). The Stimulus-Driven Theory of Probabilistic Dynamics as a Framework for Probabilistic Safety Assessment. *In Proceedings of of PSAM7–Esrel'2004*, pages 687–693, Berlin, Germany.

[Jensen, 1996] Jensen, F. V. (1996). *An introduction to Bayesian networks*, volume 210. UCL press London.

[Jourdain et Labeau, 2011] Jourdain, A. et Labeau, P. (2011). A Monte Carlo algorithm for dynamic PSA based on the concept of stimulus. *In ANS PSA 2011 International Topical Meeting on Probabilistic Safety Assessment and Analysis*, pages 535–546, Wilmington, USA.

[Julius, 2006] Julius, A. A. (2006). Approximate abstraction of stochastic hybrid automata. *Hybrid Systems : Computation and Control*, pages 318–332.

[Kalman *et al.*, 1960] Kalman, R. E. *et al.* (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

[Kalos et Whitlock, 1986] Kalos, M. H. et Whitlock, P. A. (1986). Monte Carlo Methods, Volume I. *Basics (2nd edn.) Wiley, New York*.

[Karanta, 2013] Karanta, I. (2013). Implementing dynamic flowgraph methodology models with logic programs. volume 227, pages 302–314. SAGE Publications.

[Kermisch et Labeau, 2000] Kermisch, C. et Labeau, P.-E. (2000). Approche dynamique de la fiabilité des systèmes. Project 6/2000 de l'ISdF.

[Kloos, 2011] Kloos, M. (2011). Research activities of Germany's GRS in the field if dynamic PSA. *In ANS PSA 2011 International Topical Meeting on Probabilistic Safety Assesment and Analysis*, Wilmington, USA.

[Kloos et Peschke, 2006] Kloos, M. et Peschke, J. (2006). MCDET : A probabilistic dynamics method combining monte carlo simulation with the discrete dynamic event tree approach. *Nuclear science and engineering*, 153(2):137–156.

[Labeau *et al.*, 2000] Labeau, P.-E., Smidts, C. et Swaminathan, S. (2000). Dynamic reliability : towards an integrated platform for probabilistic risk assessment. *Reliability Engineering & System Safety*, 68(3):219–254.

[Labeau et Zio, 2002] Labeau, P.-E. et Zio, E. (2002). Procedures of Monte Carlo transport simulation for applications in system engineering. *Reliability Engineering & system safety*, 77(3):217–228.

[Lair, 2011] Lair, W. (2011). *Modélisation dynamique de systèmes complexes pour le calcul de grandeurs fiabilistes et l'optimisation de la maintenance*. Thèse de doctorat, Université de Pau et des Pays de l'Adour.

[Lair *et al.*, 2011] Lair, W., Mercier, S., Roussignol, M. et Ziani, R. (2011). Piecewise deterministic Markov processes and maintenance modeling : application to maintenance of a train air-conditioning system. *Proceedings of the Institution of Mechanical Engineers, Part O : Journal of Risk and Reliability*, 225(2):199–209.

[Lambert, 1975] Lambert, H. E. (1975). Fault trees for decision making in systems analysis. Rapport technique, California Univ., Livermore (USA). Lawrence Livermore Lab.

[Maljovec *et al.*, 2013] Maljovec, D., Wang, B., Pascucci, V., Bremer, P.-T. et Mandelli, D. (2013). Analyzing Dynamic Probabilistic Risk Assessment Data through Topology-Based Clustering. *In ANS PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis*, Columbia, USA.

[Mandelli *et al.*, 2008] Mandelli, D., Aldemir, T., Bucci, P., Mangan, L. A., Kirschenbaum, J., Stocsky, M., Ekici, E. et Arudt, S. (2008). Markov/CCMT Modeling of the Benchmark System and Incorporation of the Results into an Existing PRA. *In PSAM Topical Conference*, Hong Kong, China.

[Marseguerra et Zio, 1996] Marseguerra, M. et Zio, E. (1996). Monte Carlo approach to PSA for dynamic process systems. *Reliability Engineering & System Safety*, 52(3):227–241.

[Mechraoui *et al.*, 2010] Mechraoui, A., Thiriet, J.-M., Gentil, S. *et al.* (2010). Aide à la décision et diagnostic par réseaux bayésiens d'un robot mobile commandé en réseau. *In Actes de la Sixième Conférence Internationale Francophone d'Automatique (CIFA 2010)*.

[Medjoudj, 2006] Medjoudj, M. (2006). *Contribution à l'analyse des systèmes pilotés par calculateurs : extraction de scénarios redoutés et vérification de contraintes temporelles*. Thèse de doctorat, Université Paul Sabatier - Toulouse III.

[Medjoudj et Yim, 2007] Medjoudj, M. et Yim, P. (2007). Extraction of critical scenarios in a railway level crossing control system. *International Journal of Computers, Communication and Control (IJCCC) Vol. II*, (3):252–268.

[Mercier et Roussignol, 2008] Mercier, S. et Roussignol, M. (2008). Sensitivity estimates in dynamic reliability. *Advances in mathematical modeling for reliability. IOS, Amsterdam*, pages 208–216.

[Merle, 2010] Merle, G. (2010). *Modélisation algébrique des arbres de défaillance dynamiques, contribution aux analyses qualitative et quantitative*. Thèse de doctorat, École normale supérieure de Cachan-ENS Cachan.

[Metropolis et Ulam, 1949] Metropolis, N. et Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247):335–341.

[Murphy, 2002] Murphy, K. P. (2002). *Dynamic bayesian networks : representation, inference and learning*. Thèse de doctorat, University of California.

[Parent, 1991] Parent, E. (1991). *Élaboration des consignes de gestion des barrages-réservoirs*. Thèse de doctorat, Ecole Nationale des Ponts et Chaussées.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann.

[Perez Castaneda, 2009] Perez Castaneda, G. A. (2009). *Évaluation par simulation de la sûreté de fonctionnement de systèmes en contexte dynamique hybride*. Thèse de doctorat, Institut National Polytechnique de Lorraine-INPL.

[Peschke et Kloos, 2012] Peschke, J. et Kloos, M. (2012). Options to Consider Reliability Information in a Dynamic PSA with the MCDET Method. *In Proc. of the 11th International Probabilistic Safety Assessment and Management Conference & the Annual European Safety and Reliability Conference - PSAM 11/ESREL 2012*, Helsinki, Finlande. Curran Associates, Inc.

[Peyras *et al.*, 2006] Peyras, L., Kovarik, J.-B. et Royet, P. (2006). Vers l'adaptation aux Eurocodes de la justification des barrages-poids. *Revue européenne de génie civil*, 10(1):83–109.

[Peyre, 2012] Peyre, R. (2012). La méthode de Monte-Carlo. Polycopié du cours SG241 2011-12, Ecole des mines de Nancy.

[Podofillini *et al.*, 2010] Podofillini, L., Zio, E., Mercurio, D. et Dang, V. (2010). Dynamic safety assessment : scenario identification via a possibilistic clustering approach. *Reliability engineering & System safety*, 95(5):534–549.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Raimond et Durin, 2007] Raimond, E. et Durin, T. (2007). Level 2 PSA - Comparison between classical and dynamic reliability methods. Specification and results of a benchmark exercise on consequences of hydrogen combustion during invessel core degradation. *In SARNET Seminar ERMSAR 07*, Karsruhe, Germany.

[Raimond *et al.*, 2007] Raimond, E., Laurent, B., Rahni, N., Chevalier-Jabet, K. et Durin, T. (2007). Application des EPS de niveau 2 et des techniques de fiabilité dynamique.

[Rausand et Høyland, 2004] Rausand, M. et Høyland, A. (2004). *System reliability theory : models, statistical methods, and applications*, volume 396. John Wiley & Sons.

[Sadou, 2007] Sadou, N. (2007). *Aide à la conception des systèmes embarqués sûrs de fonctionnement*. Thèse de doctorat, INSA de Toulouse.

[Sadou et Demmou, 2009] Sadou, N. et Demmou, H. (2009). Reliability analysis of discrete event dynamic systems with Petri nets. *Reliability Engineering & System Safety*, 94(11):1848–1861.

[Signoret *et al.*, 2013] Signoret, J.-P., Dutuit, Y., Cacheux, P.-J., Folleau, C., Collas, S. et Thomas, P. (2013). Make your Petri nets understandable : Reliability Block Diagrams driven Petri nets. *Reliability Engineering & System Safety*, 113:61–75.

[Siu, 1994] Siu, N. (1994). Risk assessment for dynamic systems : an overview. *Reliability Engineering & System Safety*, 43(1):43–73.

[Smidts, 1994] Smidts, C. (1994). Probabilistic dynamics : a comparison between continuous event trees and a discrete event tree model. *Reliability Engineering & System Safety*, 44(2):189–206.

[Smith, 2006] Smith, M. (2006). Dam risk analysis using Bayesian networks. *"Geohazards", Farrokh Nadim, Rudolf Pattler, Herbert Einstein, Herbert Klapperich, and Steven Kramer Eds, ECI Symposium Series, Volume P07.*

[Strubbe et van der Schaft, 2005] Strubbe, S. et van der Schaft, A. (2005). Stochastic equivalence of CPDP-Automata and piecewise deterministic Markov processes. *In IFAC world congress*, volume 16, pages 25–30.

[Strubbe et van der Schaft, 2006] Strubbe, S. et van der Schaft, A. (2006). Communicating piecewise deterministic Markov processes. *Stochastic Hybrid Systems*, pages 65–104.

[Tchangani et Noyes, 2005] Tchangani, P. et Noyes, D. (2005). Attempt to modeling dynamic reliability using dynamic bayesian networks. *In 6e Congrès International pluridisciplinaire, qualité et sûreté de fonctionnement, Qualita 2005, Bordeaux.*

[Tombuyses et Aldemir, 1997] Tombuyses, B. et Aldemir, T. (1997). Continuous cell-to-cell mapping. *Journal of sound and Vibration*, 202(3):395–415.

[Tyrväinen, 2013] Tyrväinen, T. (2013). Risk importance measures in the dynamic flowgraph methodology. *Reliability Engineering & System Safety*, 118:35–50.

[Varuttamaseni et Lee, 2011] Varuttamaseni, A. et Lee, John C.and Youngblood, R. W. (2011). Bayesian Network Representing System Dynamics in Risk Analysis of Nuclear Systems. *In ANS PSA 2011 International Topical Meeting on Probabilistic Safety Assessment and Analysis*, Wilmington, USA.

[Villemeur *et al.*, 1988] Villemeur, A., Caseau, P. et d'Harcourt, A. (1988). *Sûreté de fonctionnement des systèmes industriels : fiabilité, facteurs humains, informatisation.* Eyrolles.

[Wang, 2005] Wang, L. (2005). *Support Vector Machines : theory and applications*, volume 177. Springer.

[Weber et Jouffe, 2006] Weber, P. et Jouffe, L. (2006). Complex system reliability modelling with dynamic object oriented bayesian networks (DOOBN). *Reliability Engineering & System Safety*, 91(2):149–162.

[Zaytoon, 2001] Zaytoon, J. (2001). *Systèmes dynamiques hybrides*. Traité IC2 Information, commande, communication. Série Systèmes automatisés. Hermes Science Publications.

[Zhang *et al.*, 2013] Zhang, H., De Saporta, B., Dufour, F. et Deleuze, G. (2013). Dynamic reliability by using simulink and stateflow. *Chemical Engineering Transactions*, 33:529–534.

[Zio, 1995] Zio, E. (1995). Biasing the transition probabilities in direct Monte Carlo. *Reliability Engineering & System Safety*, 47(1):59–63.

[Zio et Maio, 2009] Zio, E. et Maio, F. D. (2009). Processing dynamic scenarios from a reliability analysis of a nuclear power plant digital instrumentation and control system. *Annals of Nuclear Energy*, 36(9):1386–1399.