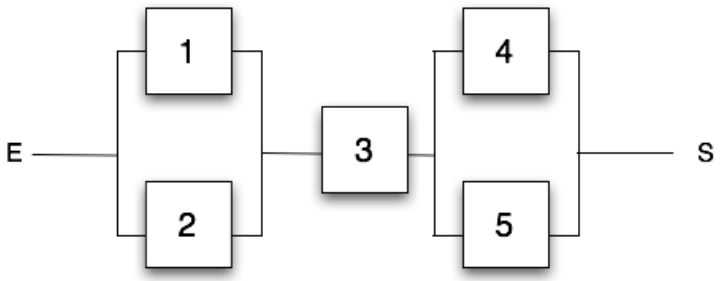
Ensimag - 3ème année



Fiabilité des Systèmes et des Logiciels

Notes de cours

Olivier Gaudoin

Table des matières

Chapitre 1

Problématique de la sûreté de fonctionnement des systèmes informatiques

1.1 Contexte

Ces dernières années, les problèmes liés à la **maîtrise des risques** et la **sûreté de fonctionnement** ont vu leur importance et leur retentissement considérablement augmenter.

Exemples :

- risque sanitaire : vache folle, SRAS, grippe aviaire, grippe A, ...
- risque environnemental : inondations, canicule, ouragan Katrina, ...
- risque industriel : explosion des navettes spatiales Challenger et Columbia, d'Ariane 5, de l'usine AZF, accidents d'avion (5 accidents mortels en août 2005, 2 en juin 2009), pannes d'électricité géantes, ...
- risque financier : Enron, subprimes, faillite de Lehmann-Brothers, affaires Kerviel et Madoff, ...
- risque géopolitique : guerre, terrorisme, ...

L'opinion publique accepte de moins en moins la notion de risque et demande des niveaux de sûreté et de sécurité de plus en plus élevés.

Les systèmes informatiques, devenus omniprésents dans tous les domaines, sont fatalement de plus en plus impliqués dans les problèmes de sûreté. Par exemple, en 2004, la France a été confrontée en moins de 6 mois à 4 pannes majeures de réseaux informatiques :

- 15-18 Juillet : le système de réservation Mosaic de la SNCF a été totalement bloqué pendant près de 4 jours, empêchant la vente et la réservation de billets. La panne s'est déclenchée quelques heures après la mise en place d'une nouvelle version de Mosaic. *La défaillance a été causée par un programme de contrôle installé ponctuellement pour s'assurer de la qualité des transmissions entre le central de réservation et les postes de travail. Son fonctionnement, combiné à la forte charge de transmission de données liée à la période de grands départs, a saturé le système et provoqué l'ensemble*

de ces perturbations (communiqué de la SNCF). Le bug a provoqué de longues files d'attente devant les guichets, mais n'a pas affecté les finances de la SNCF, car 90% des voyageurs de cette période avaient acheté leurs billets à l'avance.

- 30 et 31 Octobre : une panne de plus d'une journée est survenue sur le réseau fixe de France Télécom. 26 commutateurs sur 600 ont été affectés dans la région parisienne, le nord et l'ouest de la France. Quelques milliers d'appels ont été rejetés ou ne sont pas parvenus à leur destinataire. La panne provenait d'une *anomalie logicielle dans un équipement de traitement de la voix sur IP (VoIP) situé à Reims. Cette anomalie logicielle a provoqué des anormalités dans le formatage de la numérotation de certains appels, qui ont déclenché des protections de sécurité pour éviter une panne du réseau* (communiqué de France Télécom).
- 17-18 Novembre : une panne de plus d'une journée est survenue sur le réseau mobile de Bouygues Télécom. 1.37 millions de clients n'ont pas pu du tout utiliser leur portable et 5.6 millions ont eu des difficultés pour accéder au réseau. La facture de la panne s'est élevée à 16 millions d'euros : une journée de chiffre d'affaires perdue et l'équivalent en non facturé aux clients en compensation du désagrément subi. La panne provenait du dysfonctionnement de la base de données qui sert à repérer le mobile du client, lui permet d'acheminer ses appels et d'en recevoir. Deux serveurs centraux jumeaux sont tombés en panne au même moment, dans deux endroits différents. *Les deux serveurs sont du matériel classique et largement utilisé par de nombreux opérateurs de téléphonie mobile avec, jusqu'à ce jour, une fiabilité sans faille. En fonctionnement normal, ils se secourent en prenant le relais l'un de l'autre. La panne est de nature exceptionnelle* (communiqué de Bouygues Télécom).
- 3 Décembre : 800 terminaux de vente de la SNCF (sur 4000) ont été paralysés pendant plus d'une journée. La panne provenait d'un *algorithme défectueux qui a progressivement contaminé les terminaux de vente en gare. Cet algorithme a pour objectif de définir la zone de travail informatique de la transaction de paiement* (communiqué de la SNCF). Un nouveau bug du même type est survenu en novembre 2009.

Dans ces 4 cas, les problèmes ont été causés par des défaillances logicielles. Le site `www5.in.tum.de/~huckle/bugse.html` recense une impressionnante collection de défaillances logicielles. Parmi celles-ci :

- 4 Juin 1996 : la fusée Ariane 5 a explosé à cause d'une faute de conception logicielle, et plus précisément un problème de réutilisation. Un programme de contrôle d'un gyroscope de la fusée (en ADA) avait été transféré sans modification d'Ariane 4 à Ariane 5. Or les trajectoires de vol d'Ariane 5 sont sensiblement différentes de celles d'Ariane 4. Cela a provoqué un dépassement de mantisse qui a induit en erreur le calculateur de pilotage et fini par faire exploser le lanceur. Cette erreur a coûté 500 millions de dollars uniquement en matériel, sans parler des retards engendrés et des conséquences sur l'image de marque d'Arianespace.
- 3 mai 2000 : la moitié des ressources du réseau national de France Télécom ont été paralysées.
- 3 juin 2004 : le trafic aérien britannique a été fortement perturbé pendant plusieurs heures suite à une défaillance du système informatique de contrôle aérien, qui a privé tous les appareils en vol de ressources informatiques pendant deux heures.

- 8 octobre 2005 : un défaut dans le système de contrôle en vol de la fusée russe Rockot provoque la perte du satellite CryoSat chargé d'étudier l'influence du réchauffement climatique sur la fonte des glaces polaires. Coût de la mission : 136 millions d'euros.
- Septembre 2007 : un bug dans Excel 2007 provoque des résultats de multiplication fantaisistes comme $77.1 \times 850 = 100000$ (au lieu de 65535)i.
- 2007 : les retards de livraisons de l'Airbus A380 sont en partie dus à un problème logiciel dont l'origine tient au fait que les sites d'Hambourg et Toulouse ont utilisé deux versions différentes du logiciel de CAO de Dassault Systèmes CATIA. Les pénalités versées aux compagnies aériennes se chiffrent à plusieurs milliards d'euros.
- Mars 2008 : Chrysler rappelle 25000 Jeeps Cherokee et Commander pour corriger un défaut dans le logiciel de transmission automatique.
- Janvier 2010 : un bug dans le logiciel de lecture des puces électroniques des cartes bancaires fabriquées par Gemalto a empêché 30 millions d'allemands de se servir de leur carte de crédit pendant près d'une semaine. Pour certains, ce "bug de l'an 2010" a largement surpassé le fameux "bug de l'an 2000"...

Ces pannes majeures et répétées ont sérieusement entamé la confiance du public dans la sûreté des réseaux de télécommunication et des systèmes informatiques en général. Dans de nombreux secteurs, on a pu constater que les défaillances sont de plus en plus fréquentes. En effet, dans un contexte fortement concurrentiel, les entreprises cherchent à réduire leurs coûts et avancer les dates de lancement de leurs produits, au détriment en particulier de la sûreté de fonctionnement.

Par exemple, la fiabilité des ordinateurs de bureau a énormément chûté, en même temps que les coûts. La garantie constructeur est passée de 3 ans à 1 an en quelques années et les extensions de garantie sont très chères, ce qui indique clairement que les pannes des PC sont très fréquentes.

Extraits d'un article de Libération du 19 novembre 2004 (voir document sur le Kiosk), suite à la succession de pannes des réseaux informatiques français :

- *A la racine de toutes ces pannes, on trouve des défaillances de l'informatique.*
- *Il s'agit d'anomalies logicielles.*
- *On contrôle la qualité des services ou encore la couverture des réseaux, mais pas la vulnérabilité des systèmes.*
- *Si la panne survient, c'est que les opérateurs rognent sur tout, y compris sur la fiabilité de leur système.*
- *Le consommateur est bien en droit de réclamer aujourd'hui la garantie d'une fiabilité absolue. Celle-ci doit devenir un motif d'achat, donc un argument de vente impératif.*

Pour que la fiabilité devienne un argument de vente impératif, il faut être capable de l'évaluer ou la mesurer. C'est l'objectif essentiel de ce cours.

1.2 Terminologie générale de la sûreté de fonctionnement

Toutes les entreprises et les collectivités locales, nationales et internationales, sont concernées par la mesure, la gestion et la prévention des risques. Cela englobe les risques industriels, environnementaux, sanitaires, financiers, etc... Une des composantes principales de la gestion des risques est la sûreté de fonctionnement.

Un **système** est un ensemble de composants en interaction destiné à accomplir une tâche donnée. C'est le cas par exemple des systèmes de production, systèmes de transport, systèmes informatiques, etc...

La **sûreté de fonctionnement** (SdF, en anglais *dependability*) d'un système est la propriété qui permet à ses utilisateurs de placer une confiance justifiée dans le service qu'il leur délivre. On dit aussi que la SdF est la science des défaillances.

Un système subit une **défaillance** quand il ne peut plus délivrer le service attendu. La **panne** est l'état du système résultant d'une défaillance.

La sûreté de fonctionnement comprend 5 composantes : la fiabilité, la disponibilité, la maintenabilité, la sécurité-innocuité et la sécurité-confidentialité :

- La **fiabilité** (*reliability*) est la caractéristique du système exprimée par la probabilité qu'il délivre le service attendu dans des conditions données et pendant une durée déterminée. La fiabilité exprime l'aptitude à la continuité du service.
- La **disponibilité** (*availability*) est exprimée par la probabilité que le système délivre le service attendu dans des conditions données et à un instant donné. La disponibilité caractérise donc l'aptitude du système à fonctionner quand on a besoin de lui.
- La **maintenabilité** (*maintainability*) caractérise l'aptitude du système à être réparé quand il est défaillant, ou à évoluer.
- La **sécurité-innocuité** (*safety*) caractérise l'aptitude du système à ne pas encourir de défaillances catastrophiques.
- La **sécurité-confidentialité** (*security*) caractérise l'aptitude du système à se prémunir contre les accès ou manipulations non autorisées (virus, attaques,...).

Un **système non réparable** est un système qui est mis au rebut dès qu'il tombe en panne. C'est le cas des petits systèmes (par exemple des ampoules) ou des systèmes qui coûtent plus cher à réparer qu'à remplacer.

Un **système réparable** est un système qui, après sa défaillance, peut être remis en état de marche par des actions de réparation ou maintenance. C'est le cas de tous les systèmes complexes et en particulier des systèmes informatiques. Pour ces derniers, au lieu de réparation, on parle plutôt de correction, débogage ou mise à jour.

La maintenance des systèmes est de deux types :

- La **maintenance corrective** ou **réparation** remet en fonctionnement un système après sa défaillance.

- La **maintenance préventive** est effectuée alors que le système fonctionne et a pour but de retarder l'occurrence des défaillances futures.

Dans les études de sûreté de fonctionnement, on distingue les approches boîte noire et boîte blanche :

- **Approche boîte blanche** ou **structurelle** : on considère qu'un système complexe est constitué de composants et que sa fiabilité dépend à la fois de la fiabilité de ses composants et de la façon dont le bon fonctionnement ou la panne de chaque composant influe sur le bon fonctionnement ou la panne du système tout entier. En particulier, on considère souvent qu'un système réparable est constitué de composants non réparables. Quand un composant tombe en panne, on le remplace par un neuf, mais le système complet, lui, n'est pas remis à neuf.
- **Approche boîte noire** ou **globale** : on considère le système comme un tout, qu'on ne cherche pas à décomposer en composants. On s'intéresse alors à la suite des défaillances et réparations successives du système.

1.3 Fiabilité des matériels ou des logiciels

A priori, les défaillances des systèmes informatiques peuvent être soit d'origine matérielle, soit d'origine logicielle. En pratique, plus de 80 % sont d'origine logicielle. C'est le cas de tous les exemples présentés dans la section 1.1. On s'intéressera donc en priorité aux problèmes logiciels. Mais on peut noter qu'il existe des différences fondamentales entre la fiabilité des matériels et celle des logiciels.

- Les défaillances des matériels sont essentiellement dues à l'usure (ou vieillissement) et aux facteurs environnementaux, tandis que celles des logiciels sont dues à des fautes de conception (ou bugs), c'est-à-dire à des erreurs humaines.
- Un matériel s'use, un logiciel ne s'use pas.
- La maintenance des matériels ralentit le vieillissement des systèmes mais ne l'empêche pas, tandis que la correction des logiciels augmente leur fiabilité.
- Un logiciel non utilisé ne tombe pas en panne (le terme de panne est d'ailleurs peu utilisé pour le logiciel). Un matériel non utilisé peut tomber en panne du fait de l'usure ou des facteurs environnementaux.
- En logiciel, une faute bien repérée et corrigée est éliminée définitivement et ne peut plus se manifester. En matériel, on peut observer des défaillances répétables ou chroniques.
- La sensibilité d'un matériel à son environnement est assez forte, mais on peut néanmoins considérer qu'un matériel a une fiabilité en soi : les constructeurs quantifient la fiabilité d'une ampoule électrique quel que soit son environnement. En revanche, la sensibilité d'un logiciel a son environnement, à travers son profil opérationnel, est extrêmement forte. Un logiciel truffé de fautes peut très bien fonctionner sans défaillance si le profil opérationnel n'active pas ces fautes. Un matériel ayant beaucoup de défauts ou très usé tombera fatalement en panne, quelle que soit la manière dont on l'utilise.

- Quand un matériel est en panne, il ne peut pas fonctionner tant qu'on ne l'a pas réparé. Au contraire, un logiciel peut être relancé immédiatement après une défaillance.

On voit que les différences sont nombreuses entre fiabilité des matériels et fiabilité des logiciels. On ne pourra donc pas traiter les deux aspects de la même manière. Historiquement, les concepts de la fiabilité ont été introduits pour les matériels. On commencera donc par introduire ces concepts pour les matériels, puis on présentera les spécificités des logiciels.

1.4 Le risque logiciel

Les défaillances des logiciels sont causées par des **fautes** dans les programmes. Or, d'une part, une étude [16] a montré qu'un programmeur professionnel fait en moyenne 6 fautes pour 1000 lignes de code (LOC) écrites, et d'autre part, la taille et la complexité des logiciels ne cesse d'augmenter. Par exemple :

- la navette spatiale américaine a besoin pour voler de 500 000 LOC logiciel embarqué et 3 millions et demi de LOC au sol.
- les réseaux téléphoniques utilisent des millions de LOC pour fonctionner.
- le nombre de LOC est passé de moins de 5 millions pour Windows 95 à plus de 50 millions pour Windows Vista.
- plus généralement, un logiciel commercial standard fait en moyenne 350 000 LOC.

Par conséquent, cela fait environ 2000 fautes potentielles pour un logiciel standard, 24 000 pour la navette spatiale et 300 000 pour Vista !

Evidemment, tout est fait pour éliminer ces fautes, essentiellement par le **test** du logiciel. Mais il est extrêmement difficile et coûteux de détecter et corriger des fautes dans un logiciel. En effet, une étude de Microsoft [16] a établi qu'il fallait en moyenne 12 heures de travail pour détecter et corriger une faute. Si un logiciel contient 2000 fautes, il faut donc passer 24 000 heures pour le déboguer, soit presque 3 ans de travail cumulé. C'est pourquoi Microsoft emploie autant de personnel pour tester, vérifier et valider les programmes que pour les créer. Et malgré cela, chacun a pu expérimenter qu'il subsiste des erreurs dans les logiciels de Microsoft. Une étude plus récente [17] évalue à 60% du budget total d'un projet informatique le coût de la détection et correction des erreurs logicielles (ou maintenance du logiciel).

Malgré tous ces efforts, la complexité de la tâche fait qu'il reste toujours des fautes dans un logiciel. Comme partout, et peut-être même moins que partout, le zéro défaut est impossible. Quand ces fautes résiduelles se manifestent, leurs conséquences peuvent aller du minime au franchement catastrophique, comme on l'a vu dans la section 1.1.

Il est donc impératif de tout faire pour éviter que des pannes informatiques majeures se produisent. Pour cela, on dispose de nombreuses méthodes dont le but est de produire des logiciels de fonctionnement sûr. On peut classer ces méthodes en 4 catégories [10, 11] :

- **La prévention des fautes** : ces méthodes ont pour but d'empêcher l'occurrence et l'introduction de fautes dès la conception du logiciel. Par exemple, on a de plus en plus souvent recours à des méthodes formelles pour développer les spécifications.

- **L'élimination des fautes** : ces méthodes ont pour but de détecter des fautes dans un programme déjà écrit. Elles comprennent les preuves de programmes, les inspections formelles, la vérification et surtout le test du logiciel.
- **La tolérance aux fautes** : ces méthodes ont pour but de permettre au système de fonctionner correctement même en présence de fautes. On peut par exemple utiliser de la redondance.
- **La prévision des fautes** : ces méthodes ont pour but d'estimer la présence des fautes et de prévoir les défaillances futures du sytème.

Les méthodes présentées dans ce cours rentrent dans la dernière catégorie. En effet, il ne suffit pas d'avoir utilisé tous les moyens possibles pour développer un logiciel fiable, encore faut-il s'assurer qu'il l'est effectivement : il faut des méthodes pour atteindre des objectifs de fiabilité (les trois premières catégories, qui concernent le génie logiciel) et faire appel à d'autres méthodes pour savoir si ces objectifs sont atteints (la quatrième catégorie, qui fait intervenir des concepts probabilistes et statistiques). Par conséquent, il est très important de pouvoir prévoir l'occurrence des défaillances, et donc d'**évaluer**, **mesurer** ou **quantifier** la fiabilité d'un logiciel. Notons que cette quatrième catégorie de méthodes est aujourd'hui moins utilisée dans l'industrie que les trois autres [19], mais que cela va évoluer, notamment avec la prise en compte de ces notions dans les normes [2].

Pour les logiciels, on réduit parfois le concept de sûreté de fonctionnement à celui de sécurité-confidentialité. Cependant la sécurité-confidentialité concerne exclusivement les dysfonctionnements dus à des actes de malveillance volontaires (virus, attaques, etc.) alors que la sûreté de fonctionnement prend également en compte les dysfonctionnements non volontaires : pannes matérielles, fautes logicielles, erreurs humaines non intentionnelles, etc.

1.5 Méthodes d'évaluation de la fiabilité des logiciels selon les étapes du cycle de vie

Les méthodes d'évaluation de la fiabilité des logiciels varient suivant la nature des informations disponibles. Celles-ci sont étroitement liées au cycle de vie du logiciel, comme on le voit dans le tableau 1.1 [16] .

Phase du cycle de vie	Pourcentage d'erreurs introduites	Pourcentage d'erreurs détectées
Analyse	55%	18%
Conception	30%	10%
Codage et test	10%	50%
Vie opérationnelle	5%	22%

TABLE 1.1 – Pourcentages d'erreurs introduites et détectées selon les phases du cycle de vie du logiciel

Les types d'erreurs dans les différentes phases sont les suivantes :

- Analyse : le logiciel ne répond pas à l'attente des utilisateurs.
- Conception : mauvaise traduction des spécifications.
- Codage et test : erreurs de programmation ou de correction.
- Vie opérationnelle : erreur dans les mises à jour du système.

On constate que la majeure partie des erreurs sont introduites dans les premières phases du cycle de vie (85% en analyse et conception) et détectées dans les dernières phases (72% en codage, test et vie opérationnelle).

Dans les phases d'analyse, conception et codage, le système n'est pas encore construit, donc il ne peut pas être utilisé et aucune défaillance n'est observée. Les éléments pouvant être utilisés pour des prévisions de fiabilité sont la structure du système et les métriques logicielles (nombre de lignes de code, nombre cyclomatique du graphe de contrôle, mesures d'architecture et de spécifications, etc... [8]). A ce niveau, on peut évaluer la qualité du logiciel, mais pas sa fiabilité. Or on ne sait pas mesurer la corrélation entre qualité et fiabilité d'un logiciel.

En phase de test et en vie opérationnelle, le système fonctionne, des défaillances sont observées et des corrections sont apportées au logiciel pour remédier aux fautes apparues. L'essentiel des méthodes d'évaluation de la fiabilité repose sur l'observation et l'analyse statistique de cette suite de défaillances et corrections successives.

Tout comme les matériels, les logiciels complexes sont constitués de modules unitaires que l'on assemble. Si on est capable d'évaluer la fiabilité de chaque module et d'analyser les liens entre les différents modules, on peut appliquer les méthodes structurelles (boîte blanche) d'évaluation de fiabilité. Ce n'est pas du tout facile en pratique. Aussi considère-t-on en général un logiciel comme un tout et on évalue sa fiabilité par une approche boîte noire. C'est ce que nous ferons dans ce cours.

1.6 Utilisation des évaluations de fiabilité des logiciels

Dans un premier temps, les évaluations de fiabilité permettent de quantifier la confiance d'un utilisateur envers un système informatique, c'est-à-dire d'évaluer quantitativement le risque que l'on prend en le faisant fonctionner. Puis elles permettent de s'assurer que le logiciel a atteint un niveau de fiabilité conforme aux objectifs exprimés dans les spécifications. Un objectif de fiabilité est usuellement exprimé en termes de **taux de panne** ou **taux de défaillance**. Par exemple, pour le récent métro parisien sans conducteur Meteor, les objectifs annoncés étaient un taux de panne par rame et par heure inférieur à 10^{-9} pour le matériel et inférieur à 10^{-11} pour le logiciel.

Pour les systèmes faisant l'objet d'une garantie, les évaluations de fiabilité permettent de déterminer la durée et le coût de la garantie.

Si les mesures de fiabilité montrent que l'objectif n'est pas atteint, elles peuvent permettre d'évaluer l'effort de test à fournir pour atteindre l'objectif, et en particulier d'estimer le temps nécessaire pour y parvenir. Par conséquent, les mesures de fiabilité fournissent un **critère d'arrêt des tests** : on arrête les tests dès qu'on peut prouver, avec un niveau de confiance raisonnable, qu'un objectif donné de fiabilité est atteint. Une expérience menée à AT&T a montré que la mise en place des mesures de fiabilité a permis

une réduction de 15% de la durée de la période de tests, ce qui a entrainé un gain de 4% sur le coût total du projet, alors que le surcoût du aux mesures n'a représenté que 0.2% de ce coût total [16]. D'autres exemples sont mentionnés dans [20].

Par ailleurs, une mesure de fiabilité est un moyen d'évaluer quantitativement la qualité d'une méthode de génie logiciel donnée. Elle peut aussi fournir un indicateur de performance d'un programmeur ou d'un testeur. Cette dimension humaine délicate est parfois un frein à l'utilisation effective des évaluations de fiabilité.

1.7 Terminologie spécifique aux logiciels

En première approche, la **fiabilité** d'un logiciel est la **probabilité** qu'il fonctionne sans défaillances pendant une **durée** donnée et dans un environnement spécifié. C'est donc une notion temporelle. Le temps considéré peut être le temps d'exécution CPU (temps effectivement passé par la machine pour exécuter le programme), le temps calendaire, voir également un nombre d'opérations ou de transactions. A terme, seul le temps calendaire est important. Notons que pour certains systèmes (les systèmes réactifs), le temps n'est pas l'élément primordial : ce qui compte, c'est qu'une exécution se déroule correctement. Alors, la fiabilité est définie comme la probabilité qu'une exécution soit correcte. Dans la suite, nous ne nous intéresserons pas à ce type de système et nous conserverons donc la définition temporelle de la fiabilité.

Une **défaillance** se produit quand le résultat fourni par le logiciel n'est pas conforme au résultat prévu par les spécifications. Pour éclaircir cette notion, on peut considérer qu'un **logiciel** est un système qui, par l'intermédiaire d'un **programme**, transforme des **données d'entrée** en résultats ou **données de sortie**. Un **programme** est une suite finie d'instructions codées qui exécute une ou plusieurs tâches spécifiées. L'**exécution** d'un programme peut donc être vue (voir figure 1.1) comme une application de l'ensemble des données d'entrée dans l'ensemble des données de sortie (appelés **espace des entrées** et **espace des sorties**).

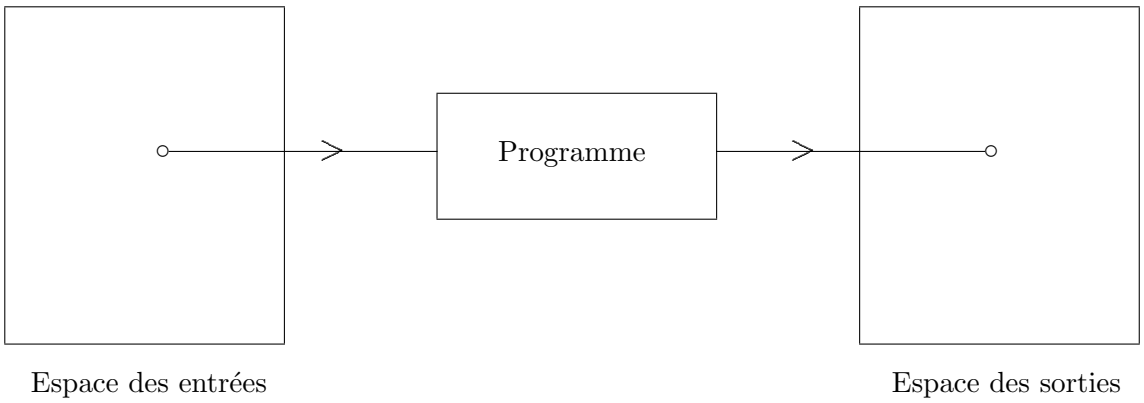


FIGURE 1.1 – L'exécution d'un programme

Les **spécifications** définissent quelle doit être la donnée de sortie pour chaque donnée d'entrée possible. Si, pour une donnée d'entrée particulière, la sortie fournie par le pro-

gramme n'est pas celle prévue par les spécifications, il y a défaillance. On voit ainsi apparaître une relation forte entre donnée d'entrée et défaillance, sur laquelle nous reviendrons.

Une **faute logicielle** ou **bug** est un défaut du programme qui, exécuté dans certaines conditions, entraînera une défaillance. Une faute est un phénomène intrinsèque au programme, elle existe même quand le logiciel n'est pas utilisé. A l'inverse, une défaillance est un phénomène dynamique : le programme doit être exécuté pour qu'elle se manifeste. Une faute est créée suite à une **erreur** humaine de l'analyste, du concepteur ou du programmeur. Aussi, on emploie le terme de **faute de conception**. Les erreurs peuvent être de spécification, de conception ou de codage. Il est également possible qu'une défaillance d'un système informatique soit due à un problème matériel. Ce type de défaillance est en général facilement identifiable. Nous ne le traiterons pas dans la suite, où nous supposerons que toutes les défaillances sont dues à des fautes de conception (au sens large).

Si on pouvait tester toutes les entrées possibles d'un logiciel, on détecterait fatalement toutes les fautes. Mais le nombre d'entrées possibles est beaucoup trop grand pour cela. Il faut donc déterminer dans l'espace des entrées un sous-ensemble d'entrées à tester. Le **profil opérationnel** définit le choix des entrées et la fréquence de sollicitation du logiciel, en associant à chaque entrée ou groupe d'entrées sa probabilité d'être fournie au programme à un instant donné. Le profil opérationnel est en général très différent en phase de test et en vie opérationnelle.

Quand une défaillance survient, on cherche à détecter la faute qui a provoqué cette défaillance et à l'éliminer. On effectue alors une **correction** ou **débogage**. Parfois, un logiciel est amené à changer radicalement certaines de ses fonctionnalités. On procède alors à un **changement de spécifications**, qui va aboutir à une nouvelle **version** du logiciel. Les corrections et les changements de spécifications peuvent être interprétés de la même manière comme des modifications du programme. Une modification d'un logiciel est parfois appelée une **maintenance logicielle**. La correction a pour but de réduire l'occurrence d'apparition des défaillances, donc elle devrait augmenter la fiabilité du logiciel.

1.8 Exemple de données

Le système observé est une machine UNIX sur laquelle tourne un gros logiciel de bases de données en phase de test. Le système a été observé sur un an, du 14 juin 1995 à 16h14 au 11 juin 1996 à minuit.

Un démon a noté tous les instants de panne, les causes des pannes et les instants de redémarrage de la machine. Suivant la nature des interruptions, des corrections ont été apportées ou pas au logiciel. Le temps pris en compte est le temps calendaire. Le tableau 1.2 présente un extrait du rapport de fiabilité brut généré automatiquement par le démon. On peut facilement en extraire les durées de bon fonctionnement et de non fonctionnement successives.

On constate que les durées de non fonctionnement de la machine sont, d'une part négligeables par rapport aux durées de bon fonctionnement, et d'autre part difficilement exploitables : on a une valeur aberrante (7168) due à une défaillance survenue pendant un pont du mois de juin, puis des valeurs quasiment constantes (à 4 mn). C'est souvent le cas, aussi on se contente en général d'analyser les durées de bon fonctionnement ou durées inter-défaillances.

Dans l'exemple, on a observé $n = 24$ durées inter-défaillances, qu'on peut noter $x_1, \ldots, x_n$ et dont les valeurs sont :

0.63	125.71	89.92	19.24	150.64	458.31	143.90	330.51
101.15	73.15	580.32	521.90	596.87	338.80	222.11	197.25
144.31	850.12	470.21	232.46	170.93	186.76	512.54	545.38

```
                    CUMULATIVE RELIABILITY REPORT
                    -----------------------------

Machine
Observation From:   06/14/95 16:14
To:                 06/11/96 00:00

Territory:          FR
Machine Usage:      DataBase
Observation Phase:  Development

Interruption:
```

Start	Stop	Up Time (h)	Stop Reason	Stop Error	Down Time (mn)
06/14/95 16:14	06/14/95 16:52	1	Unknown	Boot Down	25
06/14/95 17:17	06/19/95 23:00	126	Unknown	Boot Down	7168
06/24/95 19:28	06/28/95 13:24	90	Others	Program Int	98
06/28/95 15:02	06/29/95 10:16	19	Normal	Reboot Id	4
06/29/95 10:21	07/05/95 17:00	151	Time Out	Graphics	0
09/14/95 15:16	10/03/95 17:34	458	Normal	Reboot Id	4
10/03/95 17:39	10/09/95 17:33	144	Normal	Reboot Id	4
10/09/95 17:37	10/23/95 12:08	331	Envir. Fault	Bump Int	4
10/23/95 12:13	--------------	---	------------	----------	----

TABLE 1.2 – Extrait du rapport de fiabilité brut

Le graphe des durées inter-défaillances successives en fonction du temps (figure 1.2) semble montrer une tendance à la croissance de ces données. C'est ce qu'on s'attend à observer : une correction efficace doit augmenter les durées entre défaillances successives.

On souhaite maintenant savoir ce qui va se passer au-delà du 11 juin 1996 :

- quelle est la fiabilité du logiciel à cette date ?
- peut-on prévoir quand se produira la prochaine défaillance ?
- à quelle fréquence de défaillances peut-on s'attendre à l'avenir ?

Le but de ce cours est de fournir les moyens de répondre à ces questions. Comme la fiabilité est une probabilité, on se basera pour l'évaluer sur :

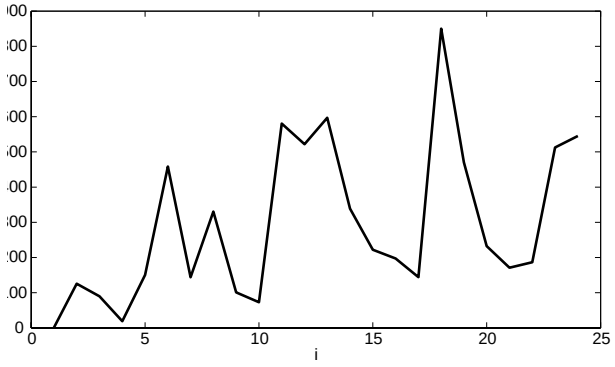


FIGURE 1.2 – Graphe des durées inter-défaillances successives

- une **modélisation probabiliste** du processus des défaillances et corrections successives du logiciel
- une **analyse statistique** de la suite de ces défaillances et corrections.

Le chapitre 2 présente les concepts de base et définit les principales mesures de fiabilité. Le chapitre 3 décrit les lois de probabilité usuelles en fiabilité, en insistant sur les lois exponentielle et de Weibull. Le chapitre 4 est consacré aux calculs de fiabilité par structure, permettant de calculer la fiabilité d'un système non réparable complexe à partir de la fiabilité de ses composants. Le chapitre 5 traite de la modélisation et de l'évaluation de fiabilité d'un logiciel que l'on ne corrige pas. Enfin, les deux derniers chapitres présentent les deux principales classes de modèles de fiabilité des logiciels, les modèles à durées inter-défaillances exponentielles (ETBF) dans le chapitre 6 et les processus de Poisson non homogènes (NHPP) dans le chapitre 7.

Chapitre 2

Les mesures de fiabilité

Les concepts généraux de la fiabilité ont été introduits pour les matériels. On les présente ici dans ce contexte en signalant à l'occasion les spécificités des logiciels. Les mesures de fiabilité sont différentes suivant que les systèmes concernés sont réparables ou non réparables.

2.1 Mesures pour les systèmes non réparables

Comme on l'a vu, un système non réparable est un système qui est mis au rebut dès qu'il tombe en panne. Les considérations sur les réparations ou corrections n'ont donc pas lieu d'être ici. Le seul point important est la **date de panne**, appelée aussi **instant de défaillance**, **durée de vie** ou **durée de bon fonctionnement** du système. Comme celle-ci n'est pas prévisible avec certitude à l'avance, on la modélise par une variable aléatoire, que l'on note X. Une durée étant un réel positif, cette variable aléatoire est à valeurs dans $\mathbb{R}^+$.

Si on s'intéressait à des systèmes pour lesquels le temps est exprimé par un nombre entier, comme un nombre de transactions, X serait à valeurs dans $\mathbb{N}$. On pourrait aussi prendre en compte la possibilité que le système soit en panne à l'instant initial, ce qui voudrait dire que $\mathbb{P}(X = 0) \neq 0$. Nous ne nous placerons ici dans aucun de ces deux cas. Par conséquent, X sera une variable aléatoire continue à valeurs dans $\mathbb{R}^+$. Sa loi de probabilité est définie par :

- sa **fonction de répartition** $F(x) = \mathbb{P}(X \leq x)$,
- sa **densité** $f(x) = F'(x)$.

Plus la durée de fonctionnement est grande, meilleure est la fiabilité du système. Donc on choisit de définir la fiabilité du système à l'instant x comme la probabilité que le système ne soit pas encore tombé en panne à l'instant x ou encore comme la probabilité que le système fonctionne sans défaillance entre 0 et x.

Définition 1 *La* **fiabilité** *d'un système non réparable est la fonction du temps R (R pour* **reliability***) définie par :*

$$\forall x \geq 0, \quad R(x) = \mathbb{P}(X > x) \tag{2.1}$$

On a évidemment $R(x) = 1 - F(x)$ et $R'(x) = -f(x)$. R est donc une fonction décroissante. Cela traduit le fait naturel que l'aptitude au bon fonctionnement d'un système non réparable diminue avec le temps. Mais la monotonie de cette fonction fait que la fiabilité n'est pas suffisamment souple pour pouvoir clairement prendre en compte la diversité des types d'usure. Aussi la principale mesure de fiabilité n'est pas la fonction de fiabilité mais le taux de défaillance.

Définition 2 *Le* **taux de défaillance** *ou* **taux de panne** *ou* **taux de hasard** *d'un système non réparable est la fonction du temps h définie par :*

$$\forall x \geq 0, \quad h(x) = \lim_{\Delta x \to 0} \frac{1}{\Delta x} \mathbb{P}(x < X \leq x + \Delta x \mid X > x) \tag{2.2}$$

Dans cette expression, la probabilité considérée est la probabilité que le système tombe en panne entre x et $x + \Delta x$ sachant qu'il a bien fonctionné entre 0 et x. Notons que la fiabilité est une probabilité mais que le taux de défaillance n'en est pas une : $h(x)$ peut être supérieur à 1.

L'interprétation du taux de défaillance est liée à celle de la densité de la façon suivante. On sait que :

$$\begin{aligned} f(x) &= F'(x) = \lim_{\Delta x \to 0} \frac{F(x + \Delta x) - F(x)}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{1}{\Delta x} \big[\mathbb{P}(X \leq x + \Delta x) - \mathbb{P}(X \leq x)\big] \\ &= \lim_{\Delta x \to 0} \frac{1}{\Delta x} \mathbb{P}(x < X \leq x + \Delta x) \end{aligned}$$

On a donc, pour Δx « petit » :

$$f(x)\,\Delta x \approx \mathbb{P}(x < X \leq x + \Delta x)$$

et :

$$h(x)\,\Delta x \approx \mathbb{P}(x < X \leq x + \Delta x \mid X > x)$$

La quantité $f(x)\,\Delta x$ peut donc être considérée comme la probabilité de défaillance juste après l'instant x alors que $h(x)\,\Delta x$ peut être considérée comme la probabilité de défaillance juste après l'instant x *sachant que* le système n'est pas tombé en panne avant x. Il y a donc une notion d'instantanéité dans $f(x)$ et une notion de durée dans $h(x)$ (comme dans $R(x)$).

On peut illustrer cette différence en comparant :

- la probabilité qu'un homme meure entre 100 et 101 ans ;
- la probabilité qu'un homme meure entre 100 et 101 ans sachant qu'il a vécu jusqu'à 100 ans.

La première (liée à la densité) est très faible : on a de très fortes chances de mourir avant 100 ans. La seconde (liée au taux de défaillance) est évidemment très forte.

On conçoit donc que le taux de défaillance est une mesure pratique de l'usure ou du vieillissement. Un taux de défaillance croissant correspond à un système qui se dégrade,

tandis qu'un taux de défaillance décroissant correspond à un système qui s'améliore avec le temps.

Il est facile d'établir les liens entre le taux de défaillance et la fiabilité :

$$\begin{aligned} h(x) &= \lim_{\Delta x \to 0} \frac{1}{\Delta x} \mathbb{P}(x < X \leq x + \Delta x \mid X > x) \\ &= \lim_{\Delta x \to 0} \frac{1}{\Delta x} \frac{\mathbb{P}(x < X \leq x + \Delta x \cap X > x)}{\mathbb{P}(X > x)} = \lim_{\Delta x \to 0} \frac{1}{\Delta x} \frac{\mathbb{P}(x < X \leq x + \Delta x)}{\mathbb{P}(X > x)} \\ &= \frac{1}{R(x)} \lim_{\Delta x \to 0} \frac{1}{\Delta x} \left[F(x + \Delta x) - F(x)\right] \\ &= \frac{f(x)}{R(x)} = \frac{f(x)}{1 - F(x)} = -\frac{R'(x)}{R(x)} = -\frac{d}{dx} \ln R(x) \end{aligned}$$

En intégrant et en prenant comme condition initiale $R(0) = 1$, car on a supposé que le système fonctionne à l'instant initial, on obtient la *formule d'exponentiation* :

$$R(x) = \exp\left(-\int_0^x h(u)\, du\right) \tag{2.3}$$

Définition 3 *Le* **taux de défaillance cumulé** *ou* **taux de hasard cumulé** *d'un système non réparable est la fonction du temps H définie par :*

$$\forall x \geq 0, \quad H(x) = \int_0^x h(u)\, du = -\ln R(x) \tag{2.4}$$

La formule d'exponentiation s'écrit donc aussi $R(x) = \exp(-H(x))$.

Enfin, puisque $f(x) = R'(x)$, la densité de X s'exprime à l'aide du taux de défaillance sous la forme :

$$f(x) = h(x)\, \exp\left(-\int_0^x h(u)\, du\right) \tag{2.5}$$

Toutes les grandeurs caractéristiques de la loi de probabilité de X s'expriment à l'aide de la fonction h. Le taux de défaillance caractérise donc la loi d'une durée de vie. C'est pourquoi, en pratique, **construire un modèle de fiabilité de systèmes non réparables revient à se donner une forme particulière pour le taux de défaillance**.

Le choix de cette forme est basé sur des considérations de modélisation ou des constatations expérimentales. De nombreuses études pratiques ont montré que le graphe du taux de défaillance d'un système non réparable simple a très souvent une **forme de baignoire**, comme dans la figure 2.1. En effet, h se décompose dans ce cas en 3 parties :

- la **période de jeunesse** : quand un système est neuf, on observe souvent des défaillances précoces, dues à des défauts intrinsèques ou des fautes de conception. Le risque de défaillance est donc assez fort au tout début de la vie du système. Ensuite il diminue car, s'il y a des défauts initiaux, ils vont se manifester tôt. h est donc d'abord décroissant. C'est le **rodage** pour les matériels mécaniques et le **déverminage** pour les matériels électroniques ;

- la **vie utile** : pendant cette période, le taux de défaillance est constant et les défaillances sont purement accidentelles ;
- le **vieillissement** : h se remet à croître car le risque de défaillance va finir par augmenter à cause de l'usure du système.

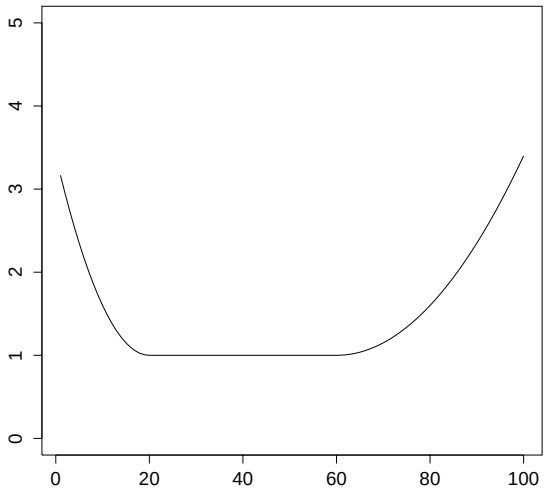


FIGURE 2.1 – Taux de défaillance en forme de baignoire

Du point de vue du consommateur cherchant à s'assurer contre les pannes du système, il est impératif d'avoir une garantie à court terme pour se prémunir contre les défauts de jeunesse. On peut souhaiter avoir une garantie à long terme contre le vieillissement, mais cela va coûter cher et les contrats ne garantissent en général pas les problèmes d'usure. En revanche, une garantie à moyen terme n'est pas forcément utile car, si le système a passé la période de jeunesse, il subira en général peu de défaillances en période de vie utile. Naturellement, pour pouvoir fixer de façon optimale les durées de garantie, il faut connaître ou estimer les dates de transition entre les différentes périodes, ce qui est généralement difficile.

La dernière mesure fondamentale de fiabilité est le MTTF.

Définition 4 *Le* **MTTF** *(***Mean Time To Failure***) d'un système non réparable est la durée moyenne de bon fonctionnement avant sa défaillance :*

$$\text{MTTF} = \mathbb{E}[X] = \int_0^{+\infty} x\, f(x)\, dx \tag{2.6}$$

Une intégration par parties aboutit alors à :

$$\text{MTTF} = \big[-x\, R(x) \big]_0^{+\infty} + \int_0^{+\infty} R(x)\, dx$$

En supposant que $R(x)$ tend vers 0 plus vite que $\frac{1}{x}$, ce qui sera toujours le cas, on obtient une formule plus usuelle pour le MTTF :

$$\text{MTTF} = \int_0^{+\infty} R(x)\,dx \tag{2.7}$$

Remarque : La transformée de Laplace de R est $\bar{R}(s) = \int_0^{+\infty} R(x)\,\exp(-sx)\,dx$. Par conséquent, $\text{MTTF} = \bar{R}(0)$.

2.2 Mesures pour les systèmes réparables

Quand les systèmes sont réparables, deux cas de figure sont possibles, selon que l'on prend en compte ou pas les durées de réparation.

2.2.1 Durées de réparation comptabilisées

Le fonctionnement du système est une succession de **durées de bon fonctionnement** et de **durées de non fonctionnement** ou de **réparation**. On note traditionnellement $\{X_i\}_{i\geq 1}$ les durées de bon fonctionnement successives et $\{Y_i\}_{i\geq 1}$ les durées de réparation successives.

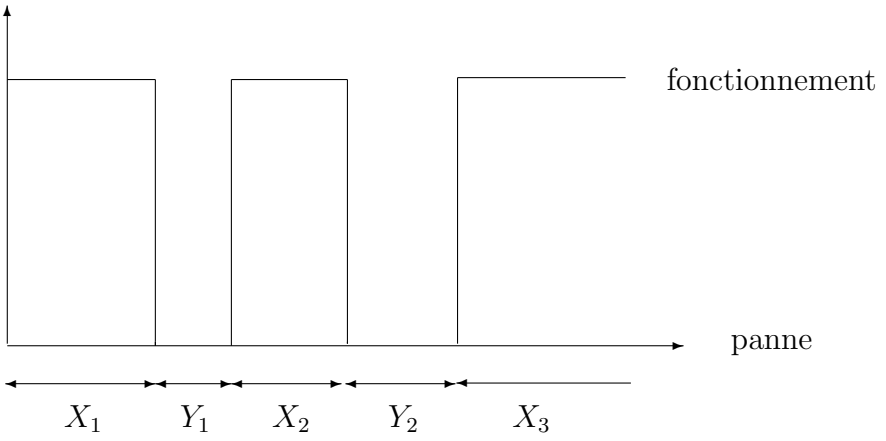


FIGURE 2.2 – Durées de bon fonctionnement et de réparation

La "durée de réparation" Y comprend en fait une durée de détection de la panne, une durée de réparation proprement dite et une durée de remise en service. Pour les logiciels, on peut relancer immédiatement le système après une défaillance, donc il n'y aura pas forcément de durée de réparation en tant que telle, mais il y aura les durées de détection de la panne et de remise en service.

Pour une durée de réparation Y, on définit des quantités similaires à celles qui ont été définies pour une durée de bon fonctionnement X :

- La **maintenabilité** est la fonction de répartition de Y. La maintenabilité en y est la probabilité qu'un système en panne à l'instant 0 soit réparé avant l'instant y :

$$\forall y \geq 0, \quad M(y) = \mathbb{P}(Y \leq y)$$

- Le **taux de réparation** est défini par

$$\forall y \geq 0, \quad \mu(y) = \lim_{\Delta y \to 0} \frac{1}{\Delta y} \mathbb{P}(y < Y \leq y + \Delta y \mid Y > y)$$

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

$$\text{MTTR} = \mathbb{E}[Y] = \int_0^{+\infty} [1 - M(y)]\, dy$$

Dans ce contexte, on peut toujours définir la fiabilité à l'instant x comme la probabilité que le système ne tombe pas en panne entre 0 et x. Cela signifie que l'instant de la première panne doit être supérieur à x, donc

$$\forall x \geq 0, \quad R(x) = \mathbb{P}(X_1 > x).$$

Mais on peut s'intéresser à une autre quantité particulièrement intéressante pour les systèmes réparables, la disponibilité.

Définition 5 *La* **disponibilité** *d'un système réparable est la fonction du temps A (A pour* **availability***) telle que :*

$$\forall t \geq 0, \quad A(t) = \textit{Probabilité que le système fonctionne à l'instant } t.$$

Donner une expression mathématique générale est beaucoup plus complexe pour la disponibilité que pour la fiabilité. Heureusement, il est souvent possible de donner des expressions simples de la **disponibilité asymptotique** :

$$A(\infty) = \lim_{t \to +\infty} A(t).$$

Remarque : La fiabilité implique une notion de durée (fonctionnement pendant une certaine durée) tandis que la disponibilité implique une notion d'instantanéité (fonctionnement à un instant donné).

Contrairement à ceux de $R(x)$ et $M(y)$, le sens de variation de $A(t)$ n'est pas déterminé. On a des systèmes à disponibilité croissante, d'autres à disponibilité décroissante et tous les sens de variation imaginables sont possibles.

Quand le système est remis à neuf après réparation, il est logique de supposer que X_2 a la même loi de probabilité que X_1. Plus généralement, on suppose couramment que les X_i sont indépendants et de même loi, et que les Y_i sont aussi indépendants et de même loi (mais pas la même que celle des X_i). Cela facilite grandement le calcul de la disponibilité. Mais dans la pratique, la réparation ne remet pas souvent le système à neuf, ce qui complexifie les calculs.

On parle parfois de MTBF = MTTF+MTTR (Mean Time Between Failures). Le MTBF est la durée moyenne entre deux défaillances successives, comprenant la durée moyenne de bon fonctionnement et la durée moyenne de réparation. On a alors souvent :

$$\lim_{t \to +\infty} A(t) = \frac{MTTF}{MTTF + MTTR},$$

ce qui se conçoit bien intuitivement.

2.2.2 Durées de réparation non comptabilisées

Dans l'exemple de données du premier chapitre, on a vu que les durées de réparation étaient sujettes à caution et difficiles à interpréter. On a simplement étudié les durées de bon fonctionnement successives. En pratique, il est fréquent également que les durées de réparation soient négligeables par rapport aux durées de bon fonctionnement. Il est donc intéressant de modéliser la situation où les durées de réparation sont négligeables ou non comptabilisées. Dans ce cas, la notion de disponibilité n'a plus aucun sens.

Dans ces conditions, on considère que l'on observe le fonctionnement d'un système réparable à partir d'un instant $T_0 = 0$. Des défaillances se produisent à des instants que l'on note $T_1, T_2, \ldots$. Après chaque défaillance, le système est réparé ou corrigé puis relancé. Le **processus des défaillances** d'un tel système réparable est défini de manière équivalente par l'un des 3 processus aléatoires suivants.

- la suite des instants de défaillance $\{T_i\}_{i\geq 1}$, avec $T_0 = 0$.
- la suite des durées inter-défaillances $\{X_i\}_{i\geq 1}$ où $\forall i \geq 1, X_i = T_i - T_{i-1}$ est la durée entre la $(i-1)^{\text{ème}}$ et la $i^{\text{ème}}$ défaillance. $T_i = \sum_{j=1}^{i} X_j$.
- le processus de comptage des défaillances $\{N_t\}_{t\geq 0}$, où N_t est le nombre cumulé de défaillances survenues entre 0 et t.

La figure 2.3 illustre les quantités aléatoires ainsi définies en présentant une trajectoire quelconque du processus des défaillances. La réalisation de ce processus pour les données de l'exemple de la section 1.8 est présentée dans la figure 2.4.

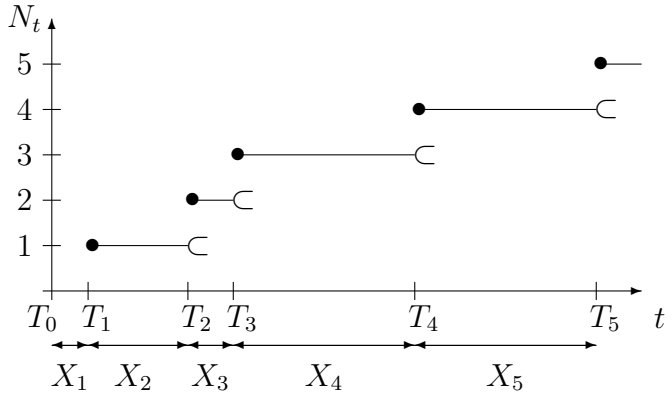


FIGURE 2.3 – Trajectoire quelconque du processus des défaillances

Remarque : L'hypothèse que les durées de correction sont négligeables peut sembler contradictoire avec les chiffres cités au chapitre 1 comme quoi il faudrait en moyenne 12 heures pour détecter et corriger une faute. Dans la pratique, le logiciel est souvent relancé sans que la correction soit faite. Il fonctionne pendant que l'équipe de correcteurs travaille (alors que, rappelons-le, pour le matériel, la réparation empêche le fonctionnement). La correction est introduite un peu plus tard. Les premiers travaux sur la fiabilité des logiciels ont supposé que les durées de correction étaient négligeables (ou non comptabilisées) et

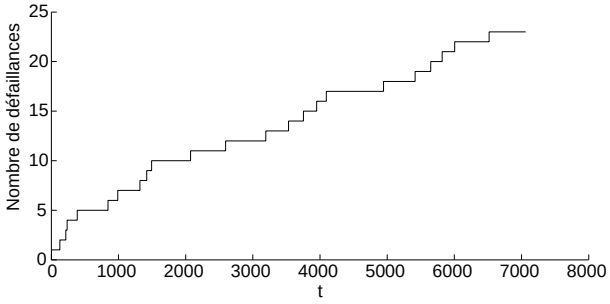


FIGURE 2.4 – Processus des défaillances pour les données de l'exemple

qu'une correction était effectuée à chaque défaillance. Nous conserverons ces hypothèses dans le cadre de ce cours, mais il est possible de prendre en compte des hypothèses plus réalistes.

Si on définit la fiabilité comme précédemment, seul l'instant de la première défaillance est en jeu :

$$\forall x \geq 0, \quad R(x) = \mathbb{P}(X_1 > x) = \mathbb{P}(T_1 > x).$$

Or, si on prend le cas de l'exemple, on souhaite être capables d'évaluer la probabilité que le logiciel fonctionne correctement pendant une durée quelconque à partir de la fin du test le 11 juin 1996. Aussi, on va modifier la définition de la fiabilité en considérant que la fiabilité du système à l'instant t exprime la probabilité qu'il fonctionne correctement pendant une certaine durée à partir de t. Pour être tout à fait complet, on va considérer que cette probabilité peut éventuellement dépendre de tout ce qui s'est passé depuis la mise en service du système. Mathématiquement, cela signifie que c'est une probabilité conditionnelle au nombre et aux instants des défaillances ayant précédé l'instant présent t. D'où la définition suivante.

Définition 6 *La* **fiabilité** *d'un système réparable à l'instant t est la fonction R_t définie par :*

$$\begin{aligned} \forall \tau \geq 0, \quad R_t(\tau; n, t_1, \ldots, t_n) &= \mathbb{P}(T_{n+1} > t + \tau | N_t = n, T_1 = t_1, \ldots, T_n = t_n) \quad (2.8) \\ &= \mathbb{P}(N_{t+\tau} - N_t = 0 | N_t = n, T_1 = t_1, \ldots, T_n = t_n) \end{aligned}$$

Autrement dit, $R_t(\tau; n, t_1, \ldots, t_n)$ est la probabilité que le système fonctionne sans défaillances pendant une durée au moins égale à τ après t, sachant qu'il y a eu exactement n défaillances entre 0 et t, aux instants $t_1, \ldots, t_n$. La première écriture exprime que la prochaine défaillance aura lieu après $t + \tau$ et la seconde exprime qu'il n'y aura aucune défaillance entre t et $t + \tau$. On conçoit bien la définition de la fiabilité à l'aide de la figure 2.5, pour laquelle n défaillances ont eu lieu entre 0 et t.

Quand on se place à l'instant t_n de la dernière défaillance, on est intéressé par la prévision de la durée X_{n+1} à attendre avant la prochaine défaillance. Sa loi de probabilité peut être influencée par le passé du processus de défaillance, donc on s'intéressera plutôt à la loi de X_{n+1} sachant $[T_1 = t_1, \ldots, T_n = t_n]$. Cette loi a pour taux de défaillance $h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(x)$.

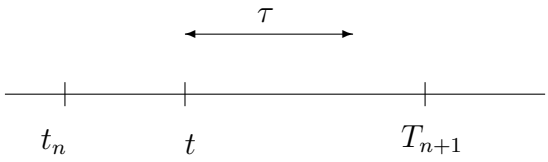


FIGURE 2.5 – Fiabilité pour une durée τ à partir de t, avec n défaillances observées

Pour un système non réparable, la propension de défaillance à l'instant t est exprimée par le taux de défaillance $h(t)$. Pour un système réparable, il est logique d'exprimer la propension de défaillance à l'instant t par le taux de défaillance de la durée inter-défaillance courante à cet instant, autrement dit $h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(t-t_n)$. C'est ce qu'on appelle l'intensité de défaillance à l'instant t.

Définition 7 *L'***intensité de défaillance** *d'un système réparable à l'instant t est la fonction λ_t définie par :*

$$\lambda_t(n;t_1,\ldots,t_n) = h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(t-t_n) \tag{2.9}$$

Suivre l'intensité de défaillance au cours du temps revient donc à étudier les taux de défaillance conditionnels successifs des X_i sachant le passé. On parle alors de concaténation ou d'amalgame de taux de défaillance.

On montre que l'intensité de défaillance s'écrit aussi :

$$\begin{aligned}\lambda_t(n;t_1,\ldots,t_n) &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}\,\mathbb{P}(t < T_{n+1} \le t+\Delta t \mid N_t = n, T_1 = t_1,\ldots,T_n = t_n)\\ &= \lim_{\Delta t\to 0}\frac{1}{\Delta t}\,\mathbb{P}(N_{t+\Delta t} - N_t = 0 \mid N_t = n, T_1 = t_1,\ldots,T_n = t_n)\end{aligned} \tag{2.10}$$

La probabilité dans cette écriture est la probabilité que le système tombe en panne entre t et $t+\Delta t$ sachant tout le passé du processus des défaillances à l'instant t.

L'intensité de défaillance est aux systèmes réparables ce que le taux de défaillance est aux systèmes non réparables. Une intensité de défaillance croissante correspond à une fréquence de défaillance qui augmente avec le temps, donc à un système qui s'use malgré les réparations. Une intensité de défaillance décroissante correspond à un système qui s'améliore avec le temps. A priori, les matériels rentrent dans la première catégorie et les logiciels dans la seconde.

Définition 8 *Le* **MTTF** *d'un système réparable à l'instant t est la durée moyenne d'attente de la prochaine défaillance à l'instant t, sachant tout le passé du processus des défaillances à cet instant :*

$$\mathrm{MTTF}_t(n;t_1,\ldots,t_n) = \mathbb{E}\left[T_{n+1} - t \mid N_t = n, T_1 = t_1,\ldots,T_n = t_n\right] \tag{2.11}$$

Les résultats suivants sont les équivalents pour les systèmes réparables des formules (2.3) et (2.7).

$$R_t(\tau; n, t_1, \ldots, t_n) = \exp\left(-\int_t^{t+\tau} \lambda_u(n; t_1, \ldots, t_n)\, du\right) = \exp\left(-\int_0^{\tau} \lambda_{t+u}(n; t_1, \ldots, t_n)\, du\right) \tag{2.12}$$

$$\mathrm{MTTF}_t(n; t_1, \ldots, t_n) = \int_0^{+\infty} R_t(\tau; n, t_1, \ldots, t_n)\, d\tau \tag{2.13}$$

La formule (2.12) avec $t = t_n$ et $\tau = t - t_n$ donne

$$\mathbb{P}(T_{n+1} > t \mid T_1 = t_1, \ldots, T_n = t_n) = \exp\left(-\int_{t_n}^{t} \lambda_u(n; t_1, \ldots, t_n)\, du\right)$$

qui n'est rien d'autre qu'une réécriture de

$$\mathbb{P}(X_{n+1} > t - t_n \mid T_1 = t_1, \ldots, T_n = t_n) = \exp\left(-\int_0^{t-t_n} h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(u)\, du\right)$$

Une autre mesure importante de fiabilité des systèmes réparables est le nombre moyen de défaillances survenues à chaque instant. C'est ce qu'on appelle la fonction moyenne.

Définition 9 *La* **fonction moyenne** *(en anglais* mean value function*) du processus des défaillances est la fonction m définie par :*

$$\forall t \geq 0, \quad m(t) = \mathbb{E}[N_t] \tag{2.14}$$

Pour le cas où on ne prend pas en compte les durées de réparation, toutes les mesures de fiabilité des systèmes réparables s'expriment à l'aide de l'intensité de défaillance. Par conséquent, **construire un modéle de fiabilité des systèmes réparables (et donc des logiciels) revient à proposer une forme particulière pour l'intensité de défaillance**.

Dans le cadre de ce cours, nous ne verrons que les 3 classes de modèles les plus simples, caractérisées par les formes d'intensité suivantes :

Processus de Poisson homogènes (HPP)	$\lambda_t(n; t_1, \ldots, t_n) = \lambda$
Modèles à durées inter-défaillances exponentielles (ETBF)	$\lambda_t(n; t_1, \ldots, t_n) = \lambda_{n+1}$
Processus de Poisson non homogènes (NHPP)	$\lambda_t(n; t_1, \ldots, t_n) = \lambda(t)$

2.3 Evaluation des mesures de fiabilité

Pour évaluer la fiabilité, le taux de défaillance, la disponibilité ou l'intensité de défaillance d'un système, il faut passer par 2 étapes.

1. **Modélisation probabiliste**. A partir d'hypothèses sur le fonctionnement du système, l'effet des réparations et la nature du phénomène aléatoire ayant engendré les défaillances, il faut proposer des modèles réalistes pour les variables aléatoires impliquées. Par exemple, il faut proposer une loi de probabilité vraisemblable pour la durée de bon fonctionnement X d'un système non réparable ou pour la suite $\{X_i\}_{i\geq 1}$ des durées inter-défaillances d'un système réparable.

2. **Analyse statistique**. Les modèles proposés ont des paramètres inconnus qu'il va falloir estimer. Pour cela, il faut observer le fonctionnement des systèmes, relever les instants des défaillances et des réparations, et effectuer une analyse statistique de ces données (qu'on appelle le **retour d'expériences**). On va pouvoir ainsi estimer les caractéristiques de fiabilité des systèmes et utiliser ces estimations pour prendre des décisions qui peuvent être cruciales en termes de sûreté de fonctionnement. Par exemple, il faut décider si un produit est suffisamment fiable pour que l'on puisse le mettre en vente sans risque, ou bien décider de l'arrêt des tests. Pour effectuer les estimations, plusieurs méthodes sont possibles, mais on utilise la plupart du temps la méthode du **maximum de vraisemblance**.

Chapitre 3

Les lois de probabilité usuelles en fiabilité

3.1 Introduction

On a dit que la fiabilité d'un système non réparable est caractérisée par la loi de probabilité de sa durée de bon fonctionnement X. Quand on s'intéresse, dans une approche boîte blanche, à un système complexe constitué de composants interconnectés, on va chercher à déterminer la loi de X en fonction des lois des durées de bon fonctionnement des composants élémentaires. Dans une approche boîte noire, on s'intéressera directement à la loi de X, sans chercher à décomposer le système en composants.

Il est donc capital d'avoir des modèles de base pour les lois des durées de bon fonctionnement de systèmes non réparables simples. Dans ce chapitre, on présente les plus utilisées de ces lois, essentiellement la loi exponentielle et la loi de Weibull. Pour chaque loi, on donnera, quand c'est possible, l'expression de la fiabilité, du taux de défaillance et du MTTF. Dans tout ce chapitre, on supposera que les durées de bon fonctionnement sont à valeurs dans $\mathbb{R}^+$, donc x sera implicitement supposé être un réel positif.

3.2 La loi exponentielle $\exp(\lambda)$

Une variable aléatoire X est de loi exponentielle de paramètre $\lambda > 0$, notée $\exp(\lambda)$, si et seulement si sa fonction de répartition est :

$$F(x) = 1 - \exp(-\lambda x)$$

- La fiabilité est $R(x) = 1 - F(x)$ d'où :

$$R(x) = \exp(-\lambda x) \tag{3.1}$$

- La densité est $f(x) = F'(x) = \lambda \exp(-\lambda x)$.
- La durée de vie moyenne est :

$$\text{MTTF} = \mathbb{E}[X] = \int_0^{+\infty} R(x)\,dx = \int_0^{+\infty} \exp(-\lambda x)\,dx = \frac{1}{\lambda} \tag{3.2}$$

- Le taux de défaillance est :

$$h(x) = \frac{f(x)}{R(x)} = \frac{\lambda \exp(-\lambda x)}{\exp(-\lambda x)} = \lambda \tag{3.3}$$

Le taux de défaillance est donc constant, ce qui signifie que la loi exponentielle modélise les durées de vie de systèmes qui ne s'usent pas et qui ne s'améliorent pas.

On dit aussi que la loi exponentielle est **sans mémoire**, ce qu'on exprime de la façon suivante : si le système n'est pas encore tombé en panne à l'instant t, c'est comme s'il était neuf à cet instant. Mathématiquement, cela s'écrit :

$$\forall x \geq 0, \quad \mathbb{P}(X > t + x \mid X > t) = \mathbb{P}(X > x).$$

On a :

$$\mathbb{P}(X > t + x \mid X > t) = \frac{\mathbb{P}(X > t + x \cap X > t)}{\mathbb{P}(X > t)} = \frac{\mathbb{P}(X > t + x)}{\mathbb{P}(X > t)} = \frac{R(t + x)}{R(t)}$$

Si X est de loi $\exp(\lambda)$, alors :

$$\mathbb{P}(X > t + x \mid X > t) = \frac{\exp(-\lambda(t + x))}{\exp(-\lambda t)} = \exp(-\lambda x) = \mathbb{P}(X > x)$$

donc la loi exponentielle est sans mémoire.

Réciproquement, si X est une variable aléatoire sans mémoire, alors on a :

$$\mathbb{P}(X > t + x \mid X > t) = \frac{R(t + x)}{R(t)} = \mathbb{P}(X > x) = R(x)$$

Donc la propriété d'absence de mémoire implique que $R(t + x) = R(t)R(x)$ pour tout $x \geq 0$ et $t \geq 0$. On en déduit que pour tout $x \geq 0$:

$$\begin{aligned} R'(x) &= \lim_{\Delta x \to 0} \frac{R(x + \Delta x) - R(x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{R(x)R(\Delta x) - R(x)}{\Delta x} \\ &= R(x) \lim_{\Delta x \to 0} \frac{R(\Delta x) - 1}{\Delta x} = R(x)\, R'(0) \end{aligned}$$

Comme R est décroissante, $R'(0)$ est une constante strictement négative que l'on note $-\lambda$, avec $\lambda > 0$. Ainsi, R est l'unique solution de l'équation différentielle $R'(x) = -\lambda R(x)$ avec comme condition initiale $R(0) = 1$. Autrement dit, on a $R(x) = \exp(-\lambda x)$. Ainsi la seule loi de probabilité à densité continue vérifiant la propriété d'absence de mémoire est la loi exponentielle. Dans le contexte de la fiabilité, cette absence de mémoire s'interprète comme une absence de vieillissement et une absence de rajeunissement.

Dans la pratique, on dit souvent que l'on peut modéliser par une loi exponentielle la durée de vie de systèmes qui sont dans leur période de vie utile, c'est-à-dire qui, dans la courbe en baignoire, ont dépassé la période de jeunesse et ne sont pas encore entrés en période d'usure. Mais c'est une erreur méthodologique car la loi de probabilité de X doit pouvoir modéliser l'ensemble de la durée de vie du système.

Par conséquent, la loi exponentielle ne devrait être utilisée que pour des systèmes qui ne s'usent pas et ne s'améliorent pas. Or tous les systèmes matériels sont soumis à l'usure, donc leur durée de vie devrait avoir a priori un taux de défaillance croissant, au moins en fin de vie. Par contre, un logiciel ne s'use pas. Tant qu'il n'est pas modifié, sa propension à subir une défaillance reste constante. Aussi la loi exponentielle a-t-elle un rôle prépondérant en fiabilité des logiciels.

Remarque 1 : Le MTTF est exprimé par une unité de temps, par exemple l'heure. La relation MTTF $= 1/\lambda$ implique donc qu'en pratique, on donne pour unité de λ l'inverse d'une unité de temps. Cela explique qu'un objectif de fiabilité soit souvent exprimé en terme de taux de panne « par heure », comme dans l'exemple de Meteor vu en section 1.6. : on sous-entend un taux de défaillance constant et on se fixe comme objectif par rame $\lambda < 10^{-9}\, h^{-1}$ pour le matériel et $\lambda < 10^{-11}\, h^{-1}$ pour le logiciel.

Remarque 2 : Si l'on admet que la durée de vie d'un système est de loi exponentielle, toute maintenance préventive est inutile puisque le système est comme neuf à chaque instant tant qu'il n'est pas tombé en panne.

Si la loi exponentielle est de loin la loi de durée de vie la plus utilisée en raison de sa simplicité, elle ne permet de modéliser ni l'usure, ni le rajeunissement. Il est donc nécessaire de disposer de lois plus sophistiquées. En fiabilité, la loi de Weibull est la plus populaire d'entre elles.

3.3 La loi de Weibull $\mathcal{W}(\eta, \beta)$

Une variable aléatoire X est de loi de Weibull de paramètre d'échelle $\eta > 0$ et de paramètre de forme $\beta > 0$, notée $\mathcal{W}(\eta, \beta)$, si et seulement si sa fonction de répartition est :

$$F(x) = 1 - \exp\left(- \left(\frac{x}{\eta}\right)^{\beta} \right)$$

- La fiabilité est :

$$R(x) = \exp\left(- \left(\frac{x}{\eta}\right)^{\beta} \right) \tag{3.4}$$

- La densité est :

$$f(x) = F'(x) = \frac{\beta}{\eta} \left(\frac{x}{\eta}\right)^{\beta-1} \exp\left(- \left(\frac{x}{\eta}\right)^{\beta} \right)$$

- La durée de vie moyenne est MTTF $= \int_0^{+\infty} \exp\left(-(x/\eta)^{\beta}\right)\, dx$. Un changement de variables $u = (x/\eta)^{\beta}$ permet d'obtenir :

$$\text{MTTF} = \eta\, \Gamma\left(\frac{1}{\beta} + 1\right) \tag{3.5}$$

  où Γ est la fonction gamma d'Euler définie par :

$$\Gamma(a) = \int_0^{+\infty} x^{a-1} \exp(-x)\, dx \tag{3.6}$$

En particulier, $\Gamma(n) = (n-1)!$ pour tout $n \in \mathbb{N}^*$.

- Le taux de défaillance est :

$$h(x) = \frac{f(x)}{R(x)} = \frac{\beta}{\eta}\left(\frac{x}{\eta}\right)^{\beta-1} \tag{3.7}$$

Le taux de défaillance de la loi de Weibull est donc une puissance du temps, ce qui permet de modéliser de nombreuses situations. En particulier :

- si $\beta < 1$, h est décroissant donc le système s'améliore ;
- si $\beta > 1$, h est croissant donc le système s'use ;
- si $\beta = 1$, h est constant et on retrouve la loi exponentielle comme cas particulier de la loi de Weibull.

La figure 3.1 donne les graphes des taux de défaillance de la loi de Weibull pour $\beta \in \{0.5, 1, 1.5, 3\}$.

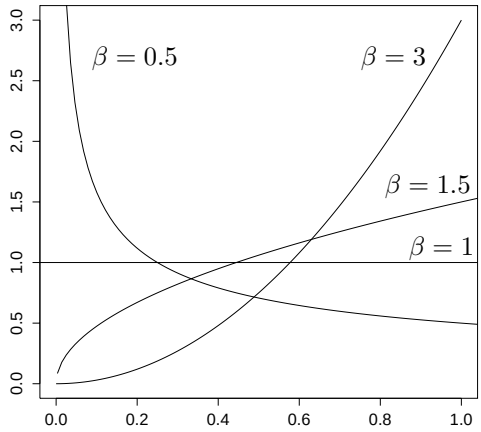


FIGURE 3.1 – Taux de défaillance de la loi de Weibull

Remarquons que pour $\beta \in]1, 2[$, h est concave, donc le système s'use, mais de moins en moins vite. L'interprétation de ce type d'usure est difficile et fait l'objet de controverses. Pour $\beta > 2$, h est convexe, ce qui correspond à une accélération de l'usure. Cela se conçoit plus facilement.

On dit parfois que la loi de Weibull permet de modéliser la période de jeunesse (pour $\beta < 1$), la vie utile (pour $\beta = 1$) et la période de vieillissement (pour $\beta > 1$). Là encore, c'est une erreur méthodologique car on doit représenter l'ensemble de la durée de vie par une seule loi de probabilité.

La loi de Weibull est très liée à la loi exponentielle, d'une part parce que la loi exponentielle est une loi de Weibull particulière et d'autre part par la propriété suivante.

Proposition 1 *Si X est de loi de Weibull $\mathcal{W}(\eta, \beta)$, alors X^β est de loi* $\exp(1/\eta^\beta)$.

Démonstration.

$$\mathbb{P}(X^\beta > x) = \mathbb{P}(X > x^{1/\beta}) = \exp\left(-\left(\frac{x^{1/\beta}}{\eta}\right)^\beta\right) = \exp\left(-\left(\frac{x}{\eta^\beta}\right)\right)$$

d'où le résultat. ■

Une propriété remarquable de la loi de Weibull est que c'est l'une des lois des valeurs extrêmes : pour n variables aléatoires $X_1, \ldots, X_n$ indépendantes et de même loi, la loi limite de variables aléatoires s'écrivant $a_n \min_{i=1}^n X_i + b_n$ quand on fait tendre n vers l'infini, est de trois types possibles. La seule dont le support soit $\mathbb{R}_+$ est la loi de Weibull. Concrètement, cela signifie que la loi de Weibull est un modèle naturel pour des systèmes constitués d'un très grand nombre de composants et dont la panne survient dès qu'un composant est défaillant (système série, voir chapitre 4).

3.4 Autres lois usuelles

3.4.1 La loi gamma $\mathcal{G}(\alpha, \lambda)$

X est de loi gamma de paramètre de forme $\alpha > 0$ et de paramètre d'échelle $\lambda > 0$, notée $\mathcal{G}(\alpha, \lambda)$, si et seulement si sa densité est :

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} \exp(-\lambda x)\ x^{\alpha-1}$$

où Γ est la fonction gamma définie en (3.6).

La fonction de répartition de la loi gamma n'a pas d'expression explicite, donc la fiabilité et le taux de défaillance non plus. En revanche, on dispose du MTTF et d'éléments qualitatifs sur le taux de défaillance :

- La durée de vie moyenne est : $\text{MTTF} = \dfrac{\alpha}{\lambda}$.
- On peut montrer que :
  - si $\alpha < 1$, h est décroissant donc le système s'améliore ;
  - si $\alpha > 1$, h est croissant donc le système s'use ;
  - si $\alpha = 1$, h est constant et on retrouve la loi exponentielle.

  Ces 3 cas sont représentés dans la figure 3.2.

Pour n entier, $\mathcal{G}\left(\dfrac{n}{2}, \dfrac{1}{2}\right)$ est la **loi du chi-2** à n degrés de liberté, notée χ^2_n.

Proposition 2 *Si $X_1, \ldots, X_n$ sont indépendantes et de même loi $\exp(\lambda)$, alors $\sum_{i=1}^n X_i$ est de loi gamma $\mathcal{G}(n, \lambda)$.*

Proposition 3 *Si X est de loi $\mathcal{G}(\alpha, \lambda)$ et a est un réel strictement positif, alors aX est de loi $\mathcal{G}(\alpha, \lambda/a)$.*

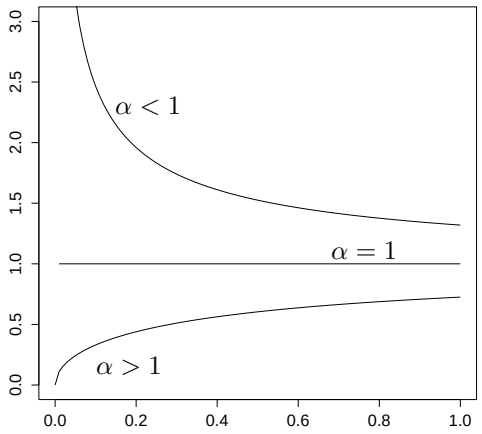


FIGURE 3.2 – Taux de défaillance de la loi gamma

3.4.2 La loi lognormale $\mathcal{LN}(m, \sigma^2)$

X est de loi lognormale de paramètres $m \in \mathbb{R}$ et $\sigma^2 > 0$, notée $\mathcal{LN}(m, \sigma^2)$, si et seulement si $\ln X$ est de loi normale $\mathcal{N}(m, \sigma^2)$.

- La densité est :
$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(\ln x - m)^2\right)$$

- Le MTTF vaut : $\text{MTTF} = \exp\left(m + \frac{\sigma^2}{2}\right)$.

Là encore, la fonction de répartition, la fiabilité et le taux de défaillance de la loi lognormale n'ont pas d'expression explicite. En revanche, on peut vérifier que le taux de défaillance croît puis décroît en tendant vers 0 (voir la figure 3.3). Ceci peut modéliser des situations réelles : un système qui se détériore puis se met à s'améliorer au bout d'un moment. En fait l'expérience montre que la loi lognormale est plus à même de modéliser des durées de réparation que des durées de bon fonctionnement.

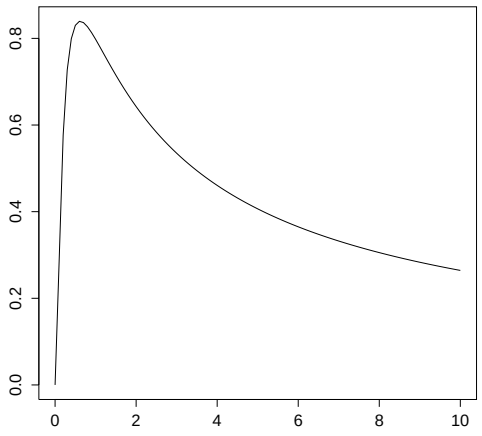


FIGURE 3.3 – Taux de défaillance de la loi lognormale

3.4.3 Lois avec taux de défaillance en baignoire

Il est étonnant de constater que, bien qu'il soit communément admis qu'en pratique le taux de défaillance d'un système non réparable a souvent une forme de baignoire, il existe peu de lois de probabilité de durées de vie possédant cette propriété. Par exemple, aucune des lois citées jusqu'à maintenant ne rentre dans ce cadre. La façon la plus simple de construire un taux de défaillance en baignoire est de « raccorder »trois taux de type Weibull respectivement décroissant, constant et croissant, en faisant en sorte que le taux résultant soit continu et à dérivée continue. Par exemple, la figure 2.1 a été obtenue à partir d'un taux de la forme :

$$h(x) = \begin{cases} \lambda + \dfrac{\beta_1}{\eta_1}\left(\dfrac{\tau_1 - x}{\eta_1}\right)^{\beta_1 - 1} & \text{si} \quad x \in [0, \tau_1[\\ \lambda & \text{si} \quad x \in [\tau_1, \tau_2] \\ \lambda + \dfrac{\beta_2}{\eta_2}\left(\dfrac{x - \tau_2}{\eta_2}\right)^{\beta_2 - 1} & \text{si} \quad x \in]\tau_2, +\infty[\end{cases}$$

Dans cette expression, la période de vie utile est l'intervalle $[\tau_1, \tau_2]$. D'autres lois de probabilité possèdent des taux de défaillance dont la forme se rapproche d'une baignoire, sans avoir pour autant une période de vie utile aussi bien délimitée. Notons par exemple :

$$\begin{aligned} h(x) &= \alpha\beta(\alpha x)^{\beta-1} + \frac{\alpha}{\beta}(\alpha x)^{1/\beta - 1} \\ h(x) &= \alpha(\beta + \lambda x)x^{\beta-1}\exp(\lambda x) \end{aligned}$$

La figure 3.4 donne les graphes des 3 taux de défaillance ci-dessus.

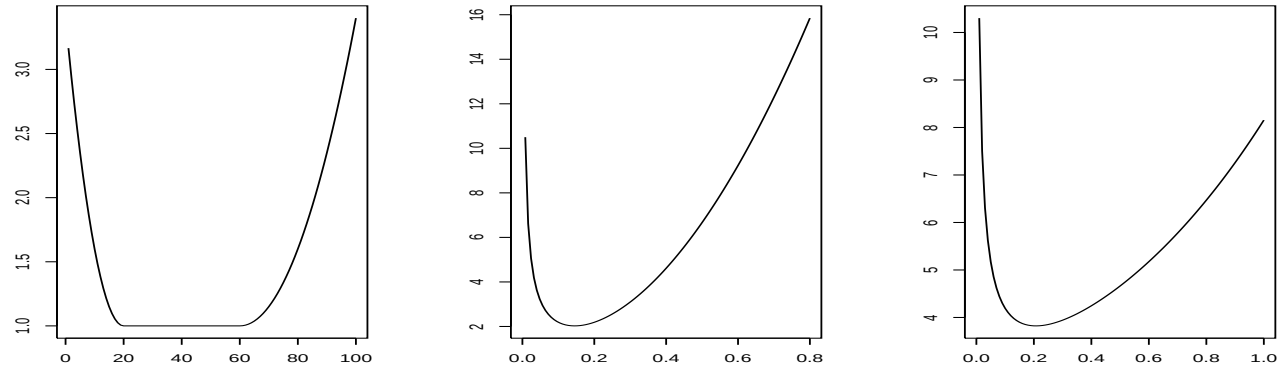


FIGURE 3.4 – Taux de défaillance en baignoire

Chapitre 4

Calculs de fiabilité par structure

4.1 Principes

Le principe des calculs de fiabilité par structure (ou architecture) est de considérer qu'un système est constitué de composants élémentaires, et que sa fiabilité dépend à la fois de la fiabilité de ses composants et de la façon dont le bon fonctionnement ou la panne de chaque composant influe sur le bon fonctionnement ou la panne du système tout entier. Il est donc nécessaire de représenter la **logique de fonctionnement** du système.

Plusieurs types de représentations sont possibles : diagrammes de fiabilité, arbres de défaillance, graphes de Markov, réseaux de Petri, diagrammes de décision binaires, réseaux bayésiens, etc... On ne s'intéressera ici qu'à des systèmes non réparables et on représentera leur fonctionnement par un diagramme de fiabilité.

Le **diagramme de fiabilité** d'un système est un graphe sans circuit admettant une entrée E et une sortie S, dont :

- les sommets, appelés **blocs**, représentent les composants du système,
- les arcs traduisent les relations entre les différents composants, au sens où le système fonctionne si et seulement si il existe un chemin allant de E à S qui ne passe que par des composants en fonctionnement.

On peut faire l'analogie avec un réseau de distribution d'eau : l'eau n'est pas coupée tant qu'il existe un chemin dans le réseau qui lui permet d'aller de son point d'entrée à son point de sortie.

Remarque : le diagramme de fiabilité est une représentation *logique* du fonctionnement du système, qui n'a rien à voir avec une représentation *physique* des liaisons entre les différents composants. De même, il n'y a aucune contrainte de précédence dans ces diagrammes.

Exemple : une chaîne hi-fi comprend une platine CD (1), un tuner FM (2), un amplificateur (3) et deux enceintes (4 et 5). Le fonctionnement normal de la chaîne implique que tous ces éléments fonctionnent. Le diagramme de fiabilité est alors donné dans la figure 4.1. En effet, si un seul de ces éléments ne fonctionne pas, la chaîne ne fonctionne pas correctement.

Mais on peut admettre un fonctionnement dégradé dans lequel il est suffisant d'entendre au moins une des deux sources sonores sur au moins une des deux enceintes. Le diagramme

FIGURE 4.1 – Diagramme de fiabilité de la chaîne hi-fi en fonctionnement normal

de fiabilité est alors donné dans la figure 4.2.

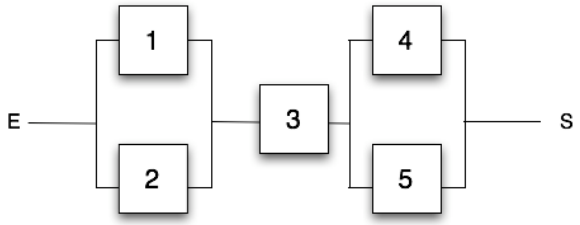


FIGURE 4.2 – Diagramme de fiabilité de la chaîne hi-fi en fonctionnement dégradé

Si on note $X_1, \ldots, X_5$ les durées de bon fonctionnement des 5 composants, il est facile de voir que, dans le premier cas, la durée de bon fonctionnement du système est $X = \min_{i=1}^{5} X_i$. Dans le deuxième cas, c'est moins évident mais on obtient que $X = \min\left(\max(X_1, X_2), X_3, \max(X_4, X_5)\right)$.

Quand le nombre de composants augmente, la structure du système peut se complexifier. Dans ce chapitre, nous allons étudier les structures de base les plus simples et donner une méthode permettant de calculer la fiabilité d'un système pour une structure complexe quelconque. Les critères de fiabilité sont un élément à prendre en compte dans le choix d'une architecture pour un système complexe.

Dans la suite, on considèrera des systèmes à n composants. Sauf mention contraire, les fonctionnements des n composants seront supposés indépendants. Pour le composant i, on note :

- X_i sa durée de bon fonctionnement,
- $r_i(x) = \mathbb{P}(X_i > x)$ sa fiabilité,
- $h_i(x)$ son taux de défaillance. $r_i(x) = \exp\left(-\int_0^x h_i(u)\,du\right)$.

Pour le système, on note X sa durée de bon fonctionnement, $R(x)$ sa fiabilité et $h(x)$ son taux de défaillance.

4.2 Systèmes série

Définition 10 *Un* **système série** *est un système qui ne fonctionne que si tous ses composants fonctionnent.*

C'est le cas de la chaine hi-fi en fonctionnement normal. Le diagramme de fiabilité est

similaire à celui de la figure 4.1, avec n composants au lieu de 5.

Un système série tombe en panne dès qu'un de ses composants tombe en panne. On a donc :

$$X = \min_{i=1}^{n} X_i$$

La fiabilité du système est alors :

$$R(x) = \mathbb{P}(X > x) = \mathbb{P}(\min_{i=1}^{n} X_i > x) = \mathbb{P}(\forall i, X_i > x) = \mathbb{P}\left(\bigcap_{i=1}^{n} [X_i > x]\right)$$

Comme on a supposé les composants indépendants, la probabilité ci-dessus est la probabilité d'une intersection d'évènements indépendants. Elle est donc égale au produit des probabilités de ces évènements :

$$R(x) = \prod_{i=1}^{n} \mathbb{P}(X_i > x) = \prod_{i=1}^{n} r_i(x)$$

On a donc :

$$R(x) = \prod_{i=1}^{n} \exp\left(-\int_0^x h_i(u)\,du\right) = \exp\left(-\sum_{i=1}^{n}\int_0^x h_i(u)\,du\right) = \exp\left(-\int_0^x \sum_{i=1}^{n} h_i(u)\,du\right)$$

Et comme $R(x) = \exp\left(-\int_0^x h(u)\,du\right)$, on en déduit que :

$$h(x) = \sum_{i=1}^{n} h_i(x)$$

Autrement dit, le taux de défaillance d'un système série à composants indépendants est égal à la somme des taux de défaillance de ses composants.

Il n'y a pas de résultat simple pour le MTTF :

$$MTTF = \int_0^{+\infty} R(x)\,dx = \int_0^{+\infty} \prod_{i=1}^{n} r_i(x)\,dx = \int_0^{+\infty} \exp\left(-\int_0^x \sum_{i=1}^{n} h_i(u)\,du\right)\,dx$$

Si tous les composants ont un taux de défaillance constant, $\forall i, \forall x, h_i(x) = \lambda_i$, donc X_i est de loi $\exp(\lambda_i)$ et $r_i(x) = \exp(-\lambda_i x)$. Alors $R(x) = \prod_{i=1}^{n} \exp(-\lambda_i x) = \exp\left(-[\sum_{i=1}^{n} \lambda_i]x\right)$ et $h(x) = \sum_{i=1}^{n} \lambda_i$ est encore constant.

On met donc là en évidence une propriété remarquable de la loi exponentielle : si $X_1, \ldots, X_n$ sont indépendantes et de lois respectives $\exp(\lambda_i)$, alors $X = \min_{i=1}^{n} X_i$ est de loi $\exp(\sum_{i=1}^{n} \lambda_i)$. Dans ce cas, on a un résultat simple pour le MTTF :

$$MTTF = \frac{1}{\sum_{i=1}^{n} \lambda_i}$$

De même, un système série constitué de composants indépendants et de durées de vie de lois de Weibull avec le même paramètre β a une durée de vie qui est encore de loi de Weibull. Enfin, on a aussi vu que la durée de vie d'un système série dont le nombre de composants tend vers l'infini a une loi qui tend vers une loi de Weibull.

4.3 Systèmes parallèles

4.3.1 Définition et propriétés

Définition 11 *Un* **système parallèle** *est un système tel qu'il suffit qu'un seul de ses composants fonctionne pour qu'il fonctionne.*

Autrement dit, la défaillance du système survient quand tous ses composants sont en panne.

Dans les systèmes parallèles, on distingue deux cas :

- La **redondance passive** ou **stand-by** : un seul composant fonctionne à la fois. Quand le composant qui fonctionne tombe en panne, il est instantanément remplacé par un des composants en attente. Dans ce cas, $X = \sum_{i=1}^{n} X_i$. La proposition 2 montre que si tous les composants sont indépendants et de même loi $\exp(\lambda)$, la durée de vie du système en redondance passive correspondant est de loi gamma $G(n, \lambda)$.
- La **redondance active** : les n composants fonctionnent en même temps.

On se place dans la suite de cette section dans le cas de la redondance active. Le diagramme de fiabilité est donné dans la figure 4.3.

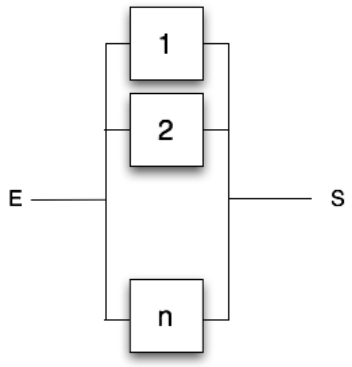


FIGURE 4.3 – Diagramme de fiabilité pour un système parallèle.

On a évidemment :

$$X = \max_{i=1}^{n} X_i$$

La fiabilité du système est alors :

$$R(x) = \mathbb{P}(X > x) = \mathbb{P}(\max_{i=1}^{n} X_i > x) = 1 - \mathbb{P}(\max_{i=1}^{n} X_i \leq x) = 1 - \mathbb{P}(\forall i, X_i \leq x)$$

Avec des composants indépendants, on obtient :

$$R(x) = 1 - \prod_{i=1}^{n} \mathbb{P}(X_i \leq x) = 1 - \prod_{i=1}^{n} (1 - \mathbb{P}(X_i > x))$$

d'où finalement :

$$R(x) = 1 - \prod_{i=1}^{n} (1 - r_i(x))$$

En écrivant $R(x) = 1 - \prod_{i=1}^{n} \left(1 - \exp\left(-\int_0^x h_i(u)\,du\right)\right)$ puis $h(x) = -\dfrac{R'(x)}{R(x)}$, on obtient que le taux de défaillance du système est :

$$h(x) = \frac{\displaystyle\sum_{i=1}^{n} h_i(x) \exp\left(-\int_0^x h_i(u)\,du\right) \prod_{j\neq i} \left(1 - \exp\left(-\int_0^x h_j(u)\,du\right)\right)}{\displaystyle 1 - \prod_{i=1}^{n} \left(1 - \exp\left(-\int_0^x h_i(u)\,du\right)\right)}$$

Donc, contrairement au cas d'un système série, le taux de défaillance d'un système parallèle ne s'exprime pas facilement en fonction du taux de défaillance de ses composants.

Il n'y a pas non plus d'expression simple du MTTF.

4.3.2 Cas où tous les composants ont un taux de défaillance constant

On a :

- $\forall i, h_i(x) = \lambda_i$.
- $R(x) = 1 - \prod_{i=1}^{n} (1 - \exp(-\lambda_i x))$.
- $h(x) = \dfrac{\displaystyle\sum_{i=1}^{n} \lambda_i \exp(-\lambda_i x) \prod_{j\neq i} (1 - \exp(-\lambda_j x))}{\displaystyle 1 - \prod_{i=1}^{n} (1 - \exp(-\lambda_i x))}$. Donc un système parallèle dont tous les composants ont un taux de défaillance constant, n'a pas un taux de défaillance constant !

En développant la fiabilité, on obtient :

$$\begin{aligned} R(x) \;=\; & \sum_{i=1}^{n} \exp(-\lambda_i x) - \sum_{i=1}^{n} \sum_{j\neq i} \exp(-(\lambda_i + \lambda_j)x) + \sum_{i,j,k\ distincts} \exp(-(\lambda_i + \lambda_j + \lambda_k)x) \\ & - \ldots + (-1)^{n+1} \exp(-[\sum_{i=1}^{n} \lambda_i]x) \end{aligned}$$

d'où on déduit le MTTF :

$$MTTF = \sum_{i=1}^{n} \frac{1}{\lambda_i} - \sum_{i=1}^{n} \sum_{j \neq i} \frac{1}{\lambda_i + \lambda_j} + \sum_{i,j,k\ distincts} \frac{1}{\lambda_i + \lambda_j + \lambda_k} - \ldots + (-1)^{n+1} \frac{1}{\sum_{i=1}^{n} \lambda_i}$$

On montre que $\lim_{t \to +\infty} h(x) = \min_{i=1}^{n} \lambda_i$. C'est logique : c'est le composant le plus fiable qui a tendance à tomber en panne le dernier, donc à provoquer la panne du système.

4.3.3 Cas où tous les composants sont identiques

On a $\forall i, r_i(x) = r(x)$. Alors la fiabilité du système est :

$$R(x) = 1 - [1 - r(x)]^n$$

Comme $r(x) \in [0, 1]$, on a $[1 - r(x)]^n \geq [1 - r(x)]^{n+1}$, donc $1 - [1 - r(x)]^n \leq 1 - [1 - r(x)]^{n+1}$. Par conséquent, quand on augmente le nombre de composants en redondance dans un système parallèle, on augmente la fiabilité du système.

Notons que c'est l'inverse pour les systèmes série puisque $[r(x)]^n \geq [r(x)]^{n+1}$.

4.3.4 Gain de fiabilité par les redondances

Le principe ci-dessus est valable à plus grande échelle. En pratique, le moyen le plus simple pour augmenter la sûreté de fonctionnement d'un système est d'ajouter des redondances, c'est-à-dire de faire fonctionner plusieurs systèmes identiques en parallèle. Par exemple, on met deux phares aux voitures au lieu d'un. Evidemment, cela a un coût et il faut donc trouver un compromis entre le coût des redondances et le gain de fiabilité qu'elles entrainent. Pour cela, il faut quantifier ce gain de fiabilité.

Prenons l'exemple d'un composant de taux de défaillance constant λ. Admettons que l'on mesure sa fiabilité par le MTTF, qui vaut ici $1/\lambda$. De combien la mise en parallèle de plusieurs composants identiques et indépendants va-t-elle augmenter le MTTF ?

Si on met en parallèle n composants, la fiabilité du système est $R(x) = 1 - [1 - r(x)]^n = 1 - [1 - \exp(-\lambda x)]^n$.

Donc le MTTF est $MTTF = \int_0^{+\infty} [1 - [1 - \exp(-\lambda x)]^n]\ dx$.

Le changement de variables $u = 1 - \exp(-\lambda x)$ permet d'écrire

$$\begin{aligned} MTTF &= \int_0^1 \frac{1 - u^n}{\lambda(1 - u)}\, du = \frac{1}{\lambda} \int_0^1 \sum_{k=0}^{n-1} u^k\, du = \frac{1}{\lambda} \sum_{k=0}^{n-1} \int_0^1 u^k\, du \\ &= \frac{1}{\lambda} \sum_{k=0}^{n-1} \left[\frac{u^{k+1}}{k+1} \right]_0^1 = \frac{1}{\lambda} \sum_{k=0}^{n-1} \frac{1}{k+1} = \frac{1}{\lambda} \left(1 + \frac{1}{2} + \ldots + \frac{1}{n} \right) \end{aligned}$$

Pour $n = 1$, on retrouve bien $MTTF = \frac{1}{\lambda}$. Pour $n = 2$, on obtient $MTTF = \frac{1}{\lambda} \left(1 + \frac{1}{2} \right) = \frac{3}{2\lambda}$. Donc mettre 2 composants au lieu d'un (cas des phares) augmente

de moitié le MTTF. Ca ne le double pas, contrairement à ce qu'on pourrait peut-être croire (c'est la redondance *passive* de 2 composants qui permet de doubler le MTTF).

Pour doubler le MTTF, il faut 4 composants car $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 2.083$. Pour multiplier le MTTF par 10, il faudrait mettre en parallèle 12369 composants !

Cette évolution lente est due au fait que la série $\sum_{i=1}^{n} \frac{1}{i}$ diverge de manière logarithmique : $\lim_{n \to +\infty} \left(1 + \frac{1}{2} + \ldots + \frac{1}{n} - \ln n \right) = 0.577215$ (constante d'Euler).

Cela signifie qu'en pratique, on ne peut pas augmenter aussi facilement la sûreté de fonctionnement d'un système en rajoutant des redondances. Par ailleurs, il faut aussi prendre en compte les contraintes de coût.

4.3.5 La redondance logicielle

Ce qui fait bien fonctionner la redondance pour les systèmes matériels, c'est l'indépendance entre les durées de bon fonctionnement des composants. Si on a deux composants de durées de vie continues et indépendantes X_1 et X_2, on a $\mathbb{P}(X_1 = X_2) = 0$. Donc quand un des composants tombe en panne, l'autre fonctionne toujours et fait fonctionner le système. Ainsi, si on a un problème avec un phare, on peut toujours rouler en utilisant le deuxième. La probabilité que les deux phares tombent en panne en même temps est nulle (sauf si un évènement externe perturbe l'ensemble de la voiture, comme une panne électrique générale ou un accident, qui provoque ce qu'on appelle une *défaillance de cause commune*).

Si on veut appliquer le principe de la redondance aux logiciels, on ne va évidemment pas faire fonctionner en parallèle deux copies du même logiciel, puisque leur fonctionnement sera identique. Il faut faire développer deux programmes ayant les mêmes spécifications par deux équipes différentes. Si on fait fonctionner les deux programmes en même temps avec les mêmes données d'entrée, on peut espérer que quand l'un aura une défaillance, l'autre fonctionnera correctement. Mais l'expérience montre que ce n'est pas aussi simple : les deux équipes butent sur les mêmes difficultés et auront tendance à faire des fautes aux mêmes endroits. Apparemment, c'est ce qui s'est passé pour la panne géante du réseau mobile de Bouygues Télécom en novembre 2004 : deux serveurs en parallèle sont tombés en panne en même temps.

En conclusion, l'indépendance des durées de bon fonctionnement, valable pour la plupart des matériels, ne l'est plus pour les logiciels. Donc la redondance logicielle est loin d'être aussi efficace que la redondance matérielle. Il reste que la redondance logicielle augmente quand même la fiabilité des logiciels, même si on ne sait pas quantifier cette augmentation. C'est donc une méthode souvent utilisée. Par exemple, Airbus utilise une redondance logicielle sur la chaîne de commande et surveillance des avions. A noter également que la redondance logicielle est très chère puisqu'il faut deux équipes de développeurs.

4.4 Systèmes k/n

Définition 12 *Un* **système** k/n *est un système qui ne fonctionne que si au moins* k *composants parmi* n *fonctionnent.*

Par exemple, le système de contrôle-commande de la température d'un réacteur chimique ou nucléaire est conçu selon une architecture 2/3.

- $k = 1$ correspond à un système parallèle.
- $k = n$ correspond à un système série.

On ne peut pas représenter ce mode de fonctionnement par un diagramme de fiabilité usuel.

La fiabilité $R(x)$ est la probabilité que k composants au moins parmi n fonctionnent encore à l'instant x. Si on note N_x le nombre de composants qui fonctionnent à l'instant x, on a :

$$R(x) = \mathbb{P}(N_x \geq k)$$

Dans le cas général, on ne peut rien dire de plus. Mais si on suppose que tous les composants sont identiques et indépendants, de même fiabilité $r(x)$, alors la variable aléatoire N_x est de loi binomiale $\mathcal{B}(n, r(x))$, ce qui permet de calculer :

$$R(x) = \sum_{j=k}^{n} C_n^j \, r(x)^j \, [1 - r(x)]^{n-j}$$

- Pour $k = n$, on obtient $R(x) = r(x)^n$. C'est bien la fiabilité d'un système série.
- Pour $k = 1$, on obtient :

$$\begin{aligned} R(x) &= \sum_{j=1}^{n} C_n^j \, r(x)^j \, [1 - r(x)]^{n-j} = \sum_{j=0}^{n} C_n^j \, r(x)^j \, [1 - r(x)]^{n-j} - C_n^0 \, r(x)^0 \, [1 - r(x)]^{n-0} \\ &= [r(x) + 1 - r(x)]^n - [1 - r(x)]^n = 1 - [1 - r(x)]^n \end{aligned}$$

C'est bien la fiabilité d'un système parallèle.

4.5 Systèmes mixtes

Les systèmes mixtes sont obtenus en combinant les systèmes série et les systèmes parallèles.

4.5.1 Systèmes série-parallèle

Définition 13 *Un* **système série-parallèle** *résulte de la mise en parallèle de sous-systèmes série.*

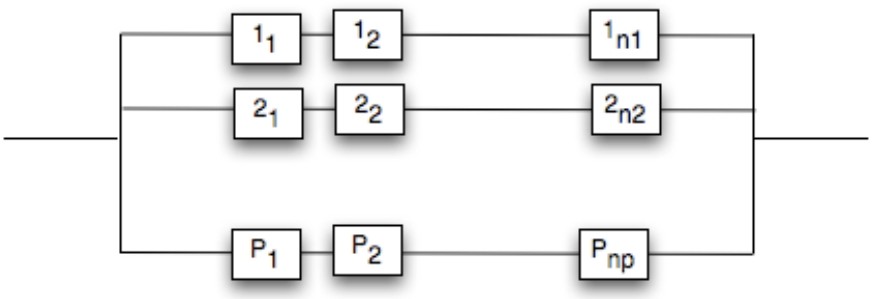


FIGURE 4.4 – Diagramme de fiabilité pour un système série-parallèle

Le diagramme de fiabilité est donné dans la figure 4.4.

Si on note $r_{ij}(x)$ la fiabilité du $j^{\text{ème}}$ composant de la $i^{\text{ème}}$ branche, les résultats précédents montrent que la fiabilité est :

$$R(x) = 1 - \prod_{i=1}^{p} \left[1 - \prod_{j=1}^{n_i} r_{ij}(x) \right]$$

4.5.2 Systèmes parallèle-série

Définition 14 *Un* **système parallèle-série** *résulte de la mise en série de sous-systèmes parallèles.*

Le diagramme de fiabilité est donné dans la figure 4.5.

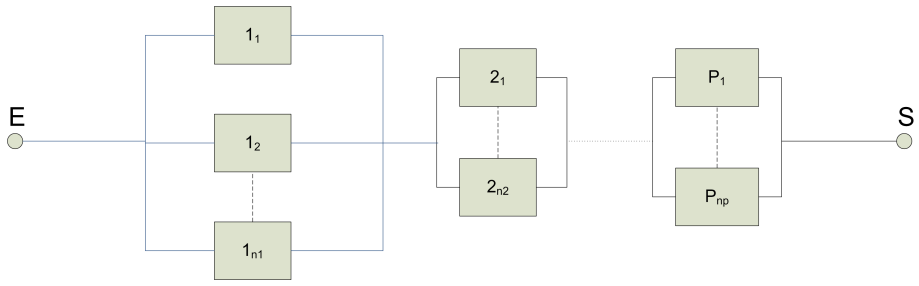


FIGURE 4.5 – Diagramme de fiabilité pour un système parallèle-série

Avec les mêmes notations que précédemment, on obtient que la fiabilité est :

$$R(x) = \prod_{i=1}^{p} \left[1 - \prod_{j=1}^{n_i} [1 - r_{ij}(x)] \right]$$

La chaîne hi-fi avec fonctionnement dégradé est un système parallèle-série. Sa fiabilité est :

$$R(x) = [1 - (1 - r_1(x))(1 - r_2(x))]\ r_3(x)\ [1 - (1 - r_4(x))(1 - r_5(x))]$$

4.6 La méthode de factorisation

De nombreux systèmes ne sont pas des systèmes série, parallèles, k/n ou mixtes. C'est le cas du système dit **en pont**, dont le diagramme de fiabilité est donné dans la figure 4.6.

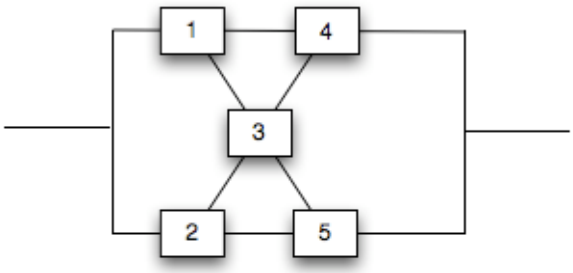


FIGURE 4.6 – Diagramme de fiabilité pour un système en pont

Pour calculer sa fiabilité, on va utiliser la **méthode de factorisation**. Celle-ci consiste à effectuer des conditionnements successifs qui vont permettre de se ramener à des systèmes mixtes.

On note $B_i(x)$ l'évènement $[X_i > x]$, signifiant que le composant i fonctionne entre 0 et x. De même, on note $B(x)$ l'évènement $[X > x]$, signifiant que le système fonctionne entre 0 et x. La fiabilité du composant i est $r_i(x) = \mathbb{P}(B_i(x))$ et la fiabilité du système est $R(x) = \mathbb{P}(B(x))$.

Le théorème des probabilités totales permet d'écrire :

$$\begin{aligned} R(x) &= \mathbb{P}(B(x)) = \mathbb{P}(B(x)|B_3(x))\,\mathbb{P}(B_3(x)) + \mathbb{P}(B(x)|\overline{B}_3(x))\,\mathbb{P}(\overline{B}_3(x)) \\ &= R_A(x)\,r_3(x) + R_B(x)\,(1 - r_3(x)) \end{aligned}$$

où $R_A(x)$ est la fiabilité du système quand on sait que le composant 3 fonctionne, c'est-à-dire la fiabilité du système A donné par la figure 4.7, et $R_B(x)$ est la fiabilité du système quand on sait que le composant 3 ne fonctionne pas, c'est-à-dire la fiabilité du système B donné par la figure 4.8.

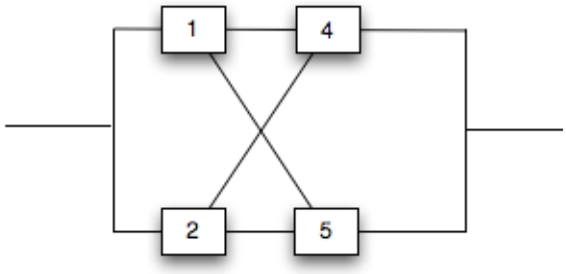


FIGURE 4.7 – Système en pont, 3 fonctionne

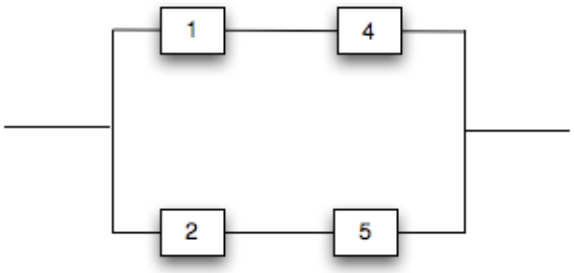


FIGURE 4.8 – Système en pont, 3 en panne

Il est clair que le système A est équivalent à un système parallèle-série, dont la fiabilité est :

$$R_A(x) = [1 - (1 - r_1(x))(1 - r_2(x))]\ [1 - (1 - r_4(x))(1 - r_5(x))]$$

De même, le système B est un système série-parallèle, dont la fiabilité est :

$$R_B(x) = 1 - [1 - r_1(x)r_4(x)]\ [1 - r_2(x)r_5(x)]$$

Finalement, la fiabilité du système en pont est :

$$\begin{aligned} R(x) &= r_3(x)\ [1 - (1 - r_1(x))(1 - r_2(x))]\ [1 - (1 - r_4(x))(1 - r_5(x))] \\ &\quad + (1 - r_3(x))\ [1 - [1 - r_1(x)r_4(x)]\ [1 - r_2(x)r_5(x)]] \end{aligned}$$

Si tous les composants sont identiques, on obtient :

$$\begin{aligned} R(x) &= r(x)\ \left[1 - (1 - r(x))^2\right]^2 + (1 - r(x))\ \left[1 - (1 - r^2(x))^2\right] \\ &= r^2(x)\ \left[2 + 2r(x) - 5r^2(x) + 2r^3(x)\right] \end{aligned}$$

Si les composants ont un taux de défaillance constant λ, $r(x) = \exp(-\lambda x)$, d'où :

$$R(x) = 2\exp(-2\lambda x) + 2\exp(-3\lambda x) - 5\exp(-4\lambda x) + 2\exp(-5\lambda x)$$

On en déduit facilement la durée de vie moyenne du système :

$$MTTF = \int_0^{+\infty} R(x)\,dx = \frac{2}{2\lambda} + \frac{2}{3\lambda} - \frac{5}{4\lambda} + \frac{2}{5\lambda} = \frac{49}{60\lambda} = 0.82\,\frac{1}{\lambda}$$

Le MTTF du système vaut donc 82 % du MTTF de ses composants.

Le taux de défaillance est :

$$h(x) = -\frac{R'(x)}{R(x)} = \frac{4\lambda\exp(-2\lambda x) + 6\lambda\exp(-3\lambda x) - 20\lambda\exp(-4\lambda x) + 10\lambda\exp(-5\lambda x)}{2\exp(-2\lambda x) + 2\exp(-3\lambda x) - 5\exp(-4\lambda x) + 2\exp(-5\lambda x)}$$

On a $\lim_{x\to+\infty} h(x) = 2\lambda$. Cela signifie qu'au bout d'un moment, le système se comporte comme deux composants en série, qui est la configuration minimale avant la panne définitive. La forme du taux de défaillance est donnée dans la figure 4.9.

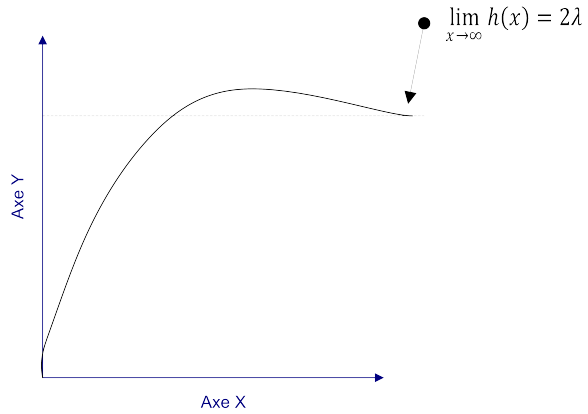


FIGURE 4.9 – Taux de défaillance du système en pont

Chapitre 5

Fiabilité d'un logiciel non corrigé : le processus de Poisson homogène

5.1 Rappels et définitions

On revient maintenant dans le cadre correspondant aux données présentées dans le chapitre 1. A un instant initial, on a mis en fonctionnement un système informatique et on a relevé ses instants de défaillance et de redémarrage successifs. Comme on l'a vu sur l'exemple, les durées de non fonctionnement (entre une panne et un redémarrage) sont souvent d'une part négligeables par rapport aux durées de bon fonctionnement et d'autre part difficilement exploitables. On va donc considérer ici que les durées de non fonctionnement ne sont pas comptabilisées et ne prendre en compte que les durées de bon fonctionnement successives. Après chaque défaillance, le logiciel peut être corrigé ou pas.

On est donc dans le cas, vu au chapitre 2, où les durées de réparation ne sont pas comptabilisées. Rappelons alors que le processus des défaillances, dont une trajectoire est représentée dans la figure 5.1, est défini indifféremment par :

- la suite des instants de défaillance $\{T_i\}_{i\geq 1}$, avec $T_0 = 0$.
- la suite des durées inter-défaillances $\{X_i\}_{i\geq 1}$ où $\forall i \geq 1, X_i = T_i - T_{i-1}$ est la durée entre la $(i-1)^{\text{ème}}$ et la $i^{\text{ème}}$ défaillance. $T_i = \sum_{j=1}^{i} X_j$.
- le processus de comptage des défaillances $\{N_t\}_{t\geq 0}$, où N_t est le nombre cumulé de défaillances survenues entre 0 et t.

Alors on a vu que :

- La fiabilité à l'instant t est une fonction qui donne la probabilité que le système fonctionne sans défaillance pendant au moins une certaine durée τ après t, sachant tout le passé du processus à l'instant t :

$$\begin{aligned} R_t(\tau; n, t_1, \ldots, t_n) &= \mathbb{P}(T_{n+1} > t + \tau | N_t = n, T_1 = t_1, \ldots, T_n = t_n) \\ &= \mathbb{P}(N_{t+\tau} - N_t = 0 | N_t = n, T_1 = t_1, \ldots, T_n = t_n) \end{aligned}$$

- Le processus des défaillances est entièrement déterminé par les taux de défaillance conditionnels des durées inter-défaillance sachant le passé $h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(x)$ ou

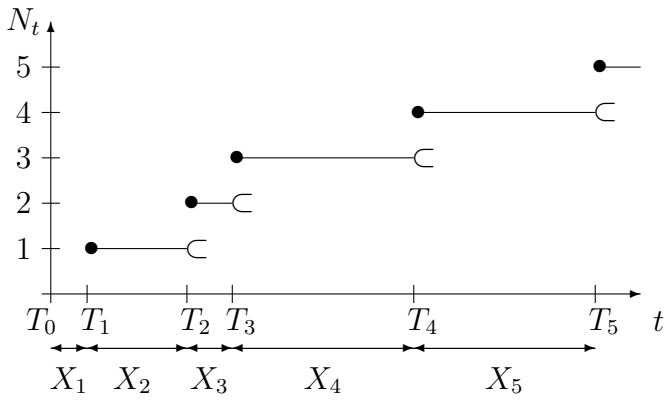


FIGURE 5.1 – Trajectoire du processus des défaillances

par son intensité de défaillance :

$$\lambda_t(n; t_1, \ldots, t_n) = h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(t - t_n)$$

- La fonction moyenne du processus est $m(t) = \mathbb{E}[N_t]$.

Nous avons vu dans le chapitre 2 les liens entre R_t et λ_t, ainsi que des expressions générales pour le MTTF et les lois de probabilité de T_{n+1} et N_t. Dans la suite du cours, nous allons réutiliser ces résultats dans des cas particuliers de modèles construits à partir de formes particulières de l'intensité λ_t.

Par ailleurs, nous avons vu que la loi exponentielle possède la propriété d'absence de mémoire, ce qui fait qu'on l'utilise pour modéliser les durées de bon fonctionnement de systèmes non réparables qui ne s'usent pas et ne s'améliorent pas. Or un logiciel possède naturellement la propriété d'absence d'usure : tant qu'un logiciel n'est pas modifié, sa propension à subir une défaillance reste constante. Par conséquent, entre deux corrections (ou entre deux défaillances puisqu'instants de corrections et de défaillances sont confondus), l'intensité de défaillance d'un logiciel doit rester constante. Cela signifie que les durées entre défaillances X_i doivent être des variables aléatoires de loi exponentielle.

Nous allons nous placer dans ce chapitre dans le cas le plus simple, celui où le logiciel n'est jamais corrigé et est relancé en l'état après chaque défaillance. C'est le cas d'un logiciel en clientèle ou entre deux mises à jours, maintenances ou versions. On dit que l'on est en situation de **fiabilité stabilisée**.

Alors, après chaque défaillance, on redémarre le système toujours dans la même situation. Il est donc normal de considérer que les durées inter-défaillances seront indépendantes et de même loi. La conjonction de ces deux hypothèses fait que le modèle à utiliser est le suivant :

Définition 15 *Pour un logiciel non corrigé, les durées inter-défaillances X_i sont indépendantes et de même loi exponentielle* $\exp(\lambda)$. *On dit que le processus des défaillances est un* **processus de Poisson homogène** *(HPP pour Homogeneous Poisson Process) de paramètre λ.*

La signification du nom HPP apparaîtra ultérieurement.

L'indépendance des durées inter-défaillances fait que la loi de X_{n+1} sachant $[T_1 = t_1, \ldots, T_n = t_n]$ ne dépend pas de $t_1, \ldots, t_n$. C'est la loi de X_{n+1}, donc la loi $\exp(\lambda)$, dont le taux de défaillance est constant et vaut λ. Par conséquent, on a :

$$\lambda_t(n; t_1, \ldots, t_n) = h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(t - t_n) = h_{X_{n+1}}(t - t_n) = \lambda$$

Par conséquent, l'intensité de défaillance du système est constante, ce qui exprime que la propension du système à tomber en panne est toujours la même au cours du temps. C'est logique compte-tenu des hypothèses effectuées.

Nous allons étudier en détail ce cas particulier pour illustrer la démarche d'évaluation de fiabilité, depuis la construction du modèle jusqu'au calcul opérationnel des prévisions de fiabilité à partir des données.

5.2 Propriétés des processus de Poisson homogènes

5.2.1 Lois des durées inter-défaillances

Par définition du modèle, les durées inter-défaillances X_i sont indépendantes et de même loi exponentielle $\exp(\lambda)$. On a donc $\forall i \geq 1, \forall x \in \mathbb{R}^+$:

- densité : $f_{X_i}(x) = \lambda \exp(-\lambda x)$,
- fonction de répartition : $F_{X_i}(x) = 1 - \exp(-\lambda x)$,
- espérance : $\mathbb{E}[X_i] = \dfrac{1}{\lambda}$. Le paramètre λ s'interprète donc comme l'inverse de la durée moyenne entre deux défaillances.

5.2.2 Lois des instants de défaillances

D'après la proposition 2 du chapitre 3, si $X_1, \ldots, X_n$ sont indépendantes et de même loi $\exp(\lambda)$, alors $T_n = \sum_{i=1}^{n} X_i$ est de loi gamma $\mathcal{G}(n, \lambda)$, dont la densité est :

$$f_{T_n}(t) = \frac{\lambda^n}{(n-1)!} \exp(-\lambda t)\, t^{n-1}$$

La durée moyenne d'attente de la $n^{\text{ème}}$ défaillance est donc $\mathbb{E}[T_n] = \dfrac{n}{\lambda}$.

5.2.3 Loi du nombre de défaillances survenues à chaque instant

Pour déterminer la loi de N_t, on peut remarquer que s'il y a eu plus de n défaillances à l'instant t, c'est que l'instant de la $n^{\text{ème}}$ défaillance est inférieur à t. Par conséquent :

$$\mathbb{P}(N_t \geq n) = \mathbb{P}(T_n \leq t) = \int_0^t f_{T_n}(x)dx = \int_0^t \frac{\lambda^n}{(n-1)!} \exp(-\lambda x)\, x^{n-1} dx = \frac{\lambda^n}{(n-1)!}\, I_n$$

avec $I_n = \int_0^t \exp(-\lambda x)\, x^{n-1} dx$. En intégrant par parties, on obtient :

$$I_n = \frac{t^n}{n}\ \exp(-\lambda t) + \frac{\lambda}{n}\ I_{n+1}$$

Alors :

$$\begin{aligned}\mathbb{P}(N_t \geq n) &= \frac{\lambda^n}{(n-1)!}\ \left[\frac{t^n}{n}\ \exp(-\lambda t) + \frac{\lambda}{n}\ I_{n+1}\right] = \frac{(\lambda t)^n}{n!}\ \exp(-\lambda t) + \frac{\lambda^{n+1}}{n!}\ I_{n+1} \\ &= \frac{(\lambda t)^n}{n!}\ \exp(-\lambda t) + \mathbb{P}(N_t \geq n+1)\end{aligned}$$

D'où :

$$\mathbb{P}(N_t = n) = \mathbb{P}(N_t \geq n) - \mathbb{P}(N_t \geq n+1) = \frac{(\lambda t)^n}{n!}\ \exp(-\lambda t) \tag{5.1}$$

ce qui prouve que N_t est de loi de Poisson $\mathcal{P}(\lambda t)$. Ce résultat classique a donné son nom au processus de Poisson homogène.

On peut montrer en fait que le processus aléatoire $\{N_t\}_{t\geq 0}$ est à accroissements indépendants, c'est à dire que, pour $0 < s < t$, $N_t - N_s$ est indépendant de ce qui s'est passé avant s, et que $N_t - N_s$ est de loi $\mathcal{P}(\lambda(t-s))$.

La fonction moyenne donne le nombre moyen de défaillances survenues entre 0 et t :

$$m(t) = \mathbb{E}[N_t] = \lambda t$$

Elle est proportionnelle à t, ce qui est logique puisque l'intensité de défaillance est constante.

5.2.4 Fiabilité

La fiabilité est :

$$R_t(\tau; n, t_1, \ldots, t_n) = \mathbb{P}(N_{t+\tau} - N_t = 0 | N_t = n, T_1 = t_1, \ldots, T_n = t_n)$$

La propriété d'accroissements indépendants entraîne que

$$R_t(\tau; n, t_1, \ldots, t_n) = \mathbb{P}(N_{t+\tau} - N_t = 0)$$

Et comme $N_{t+\tau} - N_t$ est de loi $\mathcal{P}(\lambda(t+\tau-t)) = \mathcal{P}(\lambda\tau)$, on a :

$$\forall t \geq 0, \forall n \geq 1, \forall t_1 \leq \ldots, \leq t_n \leq t, \quad R_t(\tau; n, t_1, \ldots, t_n) = \exp(-\lambda\tau)$$

La fiabilité est donc indépendante de l'instant auquel on la calcule. On la note alors simplement $R(\tau) = \exp(-\lambda\tau)$. C'est une autre façon de considérer que la fiabilité est stabilisée. C'est la propriété d'absence de mémoire de la loi exponentielle qui explique ce phénomène.

5.2.5 MTTF

$$\begin{aligned}\mathrm{MTTF}_t(n;t_1,\ldots,t_n) &= \mathbb{E}\left[T_{n+1}-t \mid N_t=n, T_1=t_1,\ldots,T_n=t_n\right]\\ &= \int_0^{+\infty} R_t(\tau;n,t_1,\ldots,t_n)\,d\tau = \int_0^{+\infty}\exp(-\lambda\tau)\,d\tau\\ &= \frac{1}{\lambda}\end{aligned}$$

Donc le MTTF est indépendant de l'instant auquel on le calcule, pour les mêmes raisons que précédemment. Par conséquent, quel que soit l'instant auquel on se place, la durée moyenne d'attente de la prochaine défaillance est $\frac{1}{\lambda}$.

Ce phénomène est parfois connu sous le nom de **paradoxe du bus** : si les instants de passage des bus à un arrêt se font selon un HPP d'intensité λ, alors la durée moyenne d'attente du bus pour un passager arrivant à cet arrêt sera la même, quel que soit le temps écoulé depuis le passage du bus précédent. Bien sûr, dans la pratique, les passages des bus ne se font pas selon un HPP.

5.3 Estimation de la fiabilité

Les résultats probabilistes ci-dessus permettent de calculer toutes les mesures intéressantes de fiabilité des logiciels. Elles s'expriment toutes à l'aide du paramètre inconnu λ. Pour avoir une évaluation de ces mesures, il faut donner une valeur à λ, c'est-à-dire estimer ce paramètre.

Pour cela, on se place à l'instant de la $n^{\text{ème}}$ défaillance t_n et on va estimer λ au vu de la suite des durées inter-défaillances successives $x_1,\ldots,x_n$ par la méthode du maximum de vraisemblance.

La fonction de vraisemblance associée à l'observation de variables aléatoires $X_1,\ldots X_n$ indépendantes et de même loi est :

$$\mathcal{L}(\lambda;x_1,\ldots,x_n) = f_{(X_1,\ldots,X_n)}(x_1,\ldots,x_n) = \prod_{i=1}^{n} f_{X_i}(x_i)$$

On obtient donc ici :

$$\mathcal{L}(\lambda;x_1,\ldots,x_n) = \prod_{i=1}^{n}\lambda\exp(-\lambda x_i) = \lambda^n\exp(-\lambda\sum_{i=1}^{n}x_i)$$

L'estimateur de maximum de vraisemblance est la valeur de λ qui maximise cette fonction, ou, de manière équivalente, son logarithme :

$$\ln\mathcal{L}(\lambda;x_1,\ldots,x_n) = n\ln\lambda - \lambda\sum_{i=1}^{n}x_i$$

Pour maximiser ce logarithme, on annule sa dérivée par rapport à λ :

$$\frac{\partial}{\partial\lambda}\ln\mathcal{L}(\lambda;x_1,\ldots,x_n) = \frac{n}{\lambda} - \sum_{i=1}^{n}x_i$$

L'estimateur de maximum de vraisemblance de λ est donc :

$$\hat{\lambda}_n = \frac{n}{\sum_{i=1}^{n} X_i} = \frac{n}{T_n} = \frac{1}{\overline{X}_n}$$

Remarquons que, puisque la durée moyenne entre deux défaillances successives est $\mathbb{E}[X_i] = 1/\lambda$, il semble naturel d'estimer λ par l'inverse de la moyenne des durées inter-défaillances observées. C'est le principe de la méthode d'estimation des moments.

Comme dans toute étude statistique, il faut se poser la question de la qualité de cet estimateur : est-il sans biais, convergent, efficace ?

$\hat{\lambda}_n$ est sans biais si et seulement si $\mathbb{E}(\hat{\lambda}_n) = \lambda$. Or :

$$\begin{aligned}
\mathbb{E}(\hat{\lambda}_n) &= \mathbb{E}\left(\frac{n}{T_n}\right) = \int \frac{n}{u}\, f_{T_n}(u)\, du = n \int_0^{+\infty} \frac{1}{u}\, \frac{\lambda^n}{(n-1)!}\, \exp(-\lambda u)\, u^{n-1}\, du \\
&= \frac{n\lambda}{n-1} \int_0^{+\infty} \frac{\lambda^{n-1}}{(n-2)!}\, \exp(-\lambda u)\, u^{n-2}\, du = \frac{n\lambda}{n-1} \int_0^{+\infty} f_{T_{n-1}}(u)\, du \\
&= \frac{n\lambda}{n-1}
\end{aligned}$$

$\mathbb{E}(\hat{\lambda}_n) \neq \lambda$, donc $\hat{\lambda}_n$ est biaisé. Mais

$$\hat{\lambda'}_n = \frac{n-1}{n}\, \hat{\lambda}_n = \frac{n-1}{T_n} \tag{5.2}$$

est un estimateur sans biais de λ.

On peut montrer que cet estimateur est convergent au sens où $\lim_{n \to +\infty} Var(\hat{\lambda'}_n) = 0$. Il est asymptotiquement efficace, au sens où sa variance est asymptotiquement égale à la borne de Cramer-Rao, borne inférieure de toutes les variances possibles d'estimateurs sans biais. En fait, on peut montrer que cet estimateur est sans biais et de variance minimale (ESBVM), donc c'est l'estimateur optimal de λ. On peut alors s'en servir pour faire des prévisions sur le futur du processus. Par exemple :

- La fiabilité $R(\tau) = \exp(-\lambda\tau)$ peut être estimée par $\hat{R}_n(\tau) = \exp(-\hat{\lambda}_n\tau)$ ou par $\hat{R'}_n(\tau) = \exp(-\hat{\lambda'}_n\tau)$.
- Le $MTTF = \dfrac{1}{\lambda}$ peut être estimé par $\dfrac{1}{\hat{\lambda}_n} = \dfrac{T_n}{n} = \overline{X}_n$ ou par $\dfrac{1}{\hat{\lambda'}_n} = \dfrac{T_n}{n-1}$.

Mais ce n'est pas parce que $\hat{\lambda'}_n$ est l'estimateur optimal de λ que $\exp(-\hat{\lambda'}_n\tau)$ sera l'estimateur optimal de $\exp(-\lambda\tau)$ ni que $\dfrac{1}{\hat{\lambda'}_n}$ sera l'estimateur optimal de $\dfrac{1}{\lambda}$. En fait, on montre que :

- L'estimateur optimal de $R(\tau) = \exp(-\lambda\tau)$ est $\hat{R}(\tau) = \left(1 - \dfrac{\tau}{T_n}\right)^{n-1}$.

- L'estimateur optimal de $MTTF = \frac{1}{\lambda}$ est $\widehat{MTTF}_n = \frac{1}{\hat{\lambda}_n} = \overline{X}_n$.

Remarque : quand il n'y a pas de correction, la fiabilité est stable donc ça n'a aucun sens de faire des calculs du type "temps de test nécessaire pour atteindre un objectif fixé de fiabilité".

5.4 Application aux données de l'exemple

Pour notre exemple de référence, on a $n = 24$, $x_1 = 0.63, \ldots, x_{24} = 545.38$, d'où $t_n = \sum_{i=1}^{n} x_i = 7063.12$. Alors, si on admet que ces données sont issues d'un HPP, on a :

- $\hat{\lambda'}_n = \frac{n-1}{t_n} = 3.256\ 10^{-3}$.
- $\widehat{MTTF}_n = \frac{t_n}{n} = 294.3$, donc on prévoit une durée moyenne d'attente entre défaillances de 294.3 h.
- La prévision de fiabilité à 100 heures est $\hat{R}(100) = \left(1 - \frac{100}{7063.12}\right)^{23} = 0.72$. On estime donc qu'il y a 72% de chances (seulement) que la prochaine défaillance ne survienne pas dans les 100 prochaines heures.

Evidemment, ces valeurs numériques n'ont de sens que si le modèle proposé est bien adapté au jeu de données. Or, d'une part des corrections ont été apportées au logiciel au fur et à mesure de l'apparition des défaillances, et d'autre part les données semblent bien exhiber une croissance de fiabilité. Donc il n'est pas logique d'appliquer un modèle de processus de Poisson homogène à ces données.

Il paraît donc indispensable d'utiliser une procédure de validation de modèles. Dans le cas traité ici, puisque les X_i sont indépendantes et de même loi exponentielle, il suffit d'appliquer un test d'adéquation à la loi exponentielle. Quand on le fait, l'hypothèse d'exponentialité est rejetée, ce qui prouve qu'il faut proposer d'autres modèles pour ce jeu de données.

5.5 Validation des logiciels

On peut appliquer le cadre de modélisation présenté ici au problème de validation des logiciels. Si on veut baser la validation sur une mesure de fiabilité, un critère usuel est de considérer que le logiciel est validé si son intensité de défaillance est inférieure à un seuil critique λ_0 fixé à l'avance. Par exemple, pour le cas de Météor cité dans le chapitre 1, $\lambda_0 = 10^{-11}$ par rame et par heure. Il revient au même de considérer que le logiciel est validé si sa fiabilité dépasse un certain seuil puisque $\lambda < \lambda_0 \Longleftrightarrow \exp(-\lambda\tau) > \exp(-\lambda_0\tau)$.

Il s'agit donc, au vu des observations, de trancher entre les deux hypothèses "$\lambda < \lambda_0$" et "$\lambda \geq \lambda_0$". Cela peut se faire à l'aide d'un test statistique d'hypothèses.

Dans un test, l'hypothèse que l'on cherche à montrer (ici la validation du logiciel) doit être l'hypothèse alternative. On va donc effectuer un test de H_0 : "$\lambda \geq \lambda_0$" contre H_1 :

"$\lambda < \lambda_0$".

5.5.1 Validation en présence de défaillances

Un test est déterminé par sa région critique, ensemble des valeurs des observations pour lesquelles on rejettera H_0 au profit de H_1. Le seuil α du test est la probabilité maximale de rejeter à tort H_0. Ici, il est naturel de conclure que $\lambda < \lambda_0$ si $\hat{\lambda'}_n$ est significativement inférieur à λ_0. On propose donc une région critique de la forme $W = \left\{\hat{\lambda'}_n < l_\alpha\right\}$. Autrement dit, si l'intensité estimée est suffisamment petite, on va conclure que l'objectif de fiabilité est atteint, avec une faible probabilité de se tromper.

La valeur de l_α est déterminée en écrivant que le seuil du test α est la probabilité maximale de rejeter H_0 alors que H_0 est vraie :

$$\alpha = \sup_{H_0} \mathbb{P}(\hat{\lambda'}_n < l_\alpha) = \sup_{\lambda \geq \lambda_0} \mathbb{P}\left(\frac{n-1}{T_n} < l_\alpha\right) = \sup_{\lambda \geq \lambda_0} \mathbb{P}\left(T_n > \frac{n-1}{l_\alpha}\right)$$

T_n est de loi gamma $\mathcal{G}(n, \lambda)$, qui est peu maniable. On préfère utiliser la loi du chi-deux, que l'on obtient facilement en écrivant que :

$$2\lambda T_n \rightsquigarrow \mathcal{G}\left(n, \frac{\lambda}{2\lambda}\right) = \mathcal{G}\left(n, \frac{1}{2}\right) = \mathcal{G}\left(\frac{2n}{2}, \frac{1}{2}\right) = \chi^2_{2n}$$

On dit que $2\lambda T_n$ est une fonction pivotale. Alors,

$$\alpha = \sup_{\lambda \geq \lambda_0} \mathbb{P}\left(2\lambda T_n > 2\lambda \frac{n-1}{l_\alpha}\right) = \sup_{\lambda \geq \lambda_0} \left[1 - F_{\chi^2_{2n}}\left(2\lambda \frac{n-1}{l_\alpha}\right)\right] \tag{5.3}$$

Une fonction de répartition est croissante, donc le terme entre crochets est une fonction décroissante de λ. Son maximum pour $\lambda \geq \lambda_0$ est donc atteint en $\lambda = \lambda_0$. Par conséquent :

$$\alpha = 1 - F_{\chi^2_{2n}}\left(2\lambda_0 \frac{n-1}{l_\alpha}\right) \Longrightarrow 2\lambda_0 \frac{n-1}{l_\alpha} = F^{-1}_{\chi^2_{2n}}(1-\alpha) \Longrightarrow l_\alpha = \frac{2\lambda_0(n-1)}{F^{-1}_{\chi^2_{2n}}(1-\alpha)} \tag{5.4}$$

Et finalement la région critique est :

$$W = \left\{\hat{\lambda'}_n < \frac{2\lambda_0(n-1)}{F^{-1}_{\chi^2_{2n}}(1-\alpha)}\right\} \tag{5.5}$$

α est une probabilité d'erreur, que l'on se fixe. Prenons par exemple $\alpha = 5\%$. Dans l'exemple, $n = 24$. La table de la loi du χ^2 permet d'établir que $F^{-1}_{\chi^2_{48}}(0.95) \approx 65.2$, d'où $W = \left\{\hat{\lambda'}_n < 0.71\ \lambda_0\right\}$.

On pourra donc conclure avec moins de 5% de chances de se tromper que $\lambda < \lambda_0$ si $\hat{\lambda'}_n < 0.71\ \lambda_0$, c'est-à-dire si l'intensité de défaillance estimée est à peu près inférieure aux trois quarts de l'intensité de défaillance objectif.

5.5.2 Validation en l'absence de défaillances

Quand la procédure de développement du logiciel a été de bonne qualité, il est possible qu'aucune défaillance ne survienne sur la période d'observation. On ne peut alors pas estimer l'intensité de défaillance λ autrement que par 0 et la procédure ci-dessus est inapplicable ($n = 0$).

Pour valider le logiciel, on peut alors utiliser le critère suivant : le logiciel sera validé si sa durée de bon fonctionnement sans défaillances est supérieure à un certain seuil. Autrement dit, si l'on n'a pas observé de défaillance au bout d'un temps assez long, on en conclut que le logiciel est suffisamment fiable.

Mathématiquement, cela signifie que l'on prend une région critique de la forme $W = \{T_1 > l_\alpha\}$. Alors, pour α fixé, l_α est choisi de sorte que :

$$\alpha = \sup_{H_0} \mathbb{P}(T_1 > l_\alpha) = \sup_{\lambda \geq \lambda_0} \exp(-\lambda l_\alpha) = \exp(-\lambda_0 l_\alpha)$$

puisque T_1 est de loi $\exp(\lambda)$. D'où $-\lambda_0 l_\alpha = \ln \alpha$ et :

$$l_\alpha = -\frac{\ln \alpha}{\lambda_0} = -\ln \alpha \, MTTF_0$$

où $MTTF_0 = \dfrac{1}{\lambda_0}$ est le MTTF objectif. Alors la région critique est

$$W = \{T_1 > -\ln \alpha \, MTTF_0\}$$

Pour $\alpha = 5\%$, $-\ln \alpha = 2.996 \approx 3$. Donc, pour valider un logiciel avec moins de 5% de chances de se tromper, il faut le faire fonctionner sans défaillances pendant une durée supérieure à 3 fois le MTTF objectif.

Pour des logiciels à haute exigence de sécurité, cela fait une durée de tests prohibitive. Dans l'exemple de Météor, l'intensité de défaillance objectif était de 10^{-11} par rame et par heure, ce qui fait une durée de test sans défaillance de $3\ 10^{11}$ heures, c'est-à-dire 30 millions d'années... On peut évidemment réduire cette durée en parallèlisant les tests, mais il est clair qu'on n'arrivera jamais à valider un objectif aussi fort de fiabilité en un temps raisonnable.

Il n'est donc pas possible d'évaluer avec précision la fiabilité de systèmes à très haut niveau de sûreté. Mais les méthodes présentées ici permettent d'obtenir des bornes inférieures pour la fiabilité, ce qui est quand même utile pour quantifier la confiance dans la sûreté du logiciel.

L'application aux données de l'exemple a montré la nécessité d'utiliser des modèles prenant en compte la qualité des corrections effectuées. C'est ce que nous allons faire dans les deux derniers chapitres du cours, avec les modèles ETBF et NHPP.

Chapitre 6

Les modèles à durées inter-défaillances exponentielles (ETBF)

6.1 Définition et propriétés

On se place comme dans le chapitre précédent dans le cas où on observe les durées inter-défaillances successives X_i d'un logiciel et où les durées de réparation ne sont pas comptabilisées. La propriété d'absence d'usure des logiciels fait qu'on supposera encore que les X_i sont de lois exponentielles. Mais cette fois, on va considérer que l'on corrige le logiciel après chaque défaillance, ce qui fait que les X_i ne seront pas de même loi. Enfin, il n'y a pas de raison particulière de supposer une dépendance entre les X_i. On aboutit alors à la définition suivante :

Définition 16 *Les* **modèles à durées inter-défaillances exponentielles** *(ETBF Pour Exponential Times Between Failures) sont les modèles de fiabilité des logiciels tels que les durées inter-défaillances X_i sont indépendantes et de lois exponentielles de paramètres respectifs λ_i,* $\exp(\lambda_i)$.

L'indépendance des durées inter-défaillances fait que la loi de X_{n+1} sachant $[T_1 = t_1, \ldots, T_n = t_n]$ ne dépend pas de $t_1, \ldots, t_n$. C'est la loi de X_{n+1}, qui est une loi exponentielle, de taux de défaillance constant λ_{n+1}. Par conséquent, l'intensité de défaillance des modèles ETBF vérifie :

$$\lambda_t(n; t_1, \ldots, t_n) = h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(t - t_n) = h_{X_{n+1}}(t - t_n) = \lambda_{n+1}$$

C'est une fonction en escalier, dont une trajectoire est représentée sur la figure 6.1.

La correction a pour effet de modifier l'intensité de défaillance : une bonne correction diminuera l'intensité ($\lambda_{n+1} < \lambda_n$, cas des 2 premières corrections sur la figure 6.1), une mauvaise correction l'augmentera ($\lambda_{n+1} > \lambda_n$, cas de la troisième correction sur cette figure).

En appliquant des résultats généraux sur les processus de défaillance à une intensité de cette forme, on obtient les principales propriétés de ces modèles, données ci-dessous.

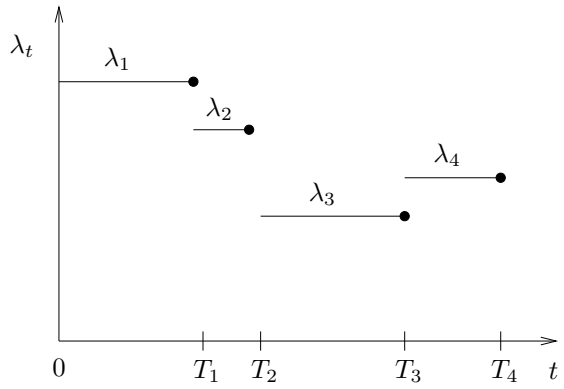


FIGURE 6.1 – Intensité d'un modèle ETBF

6.1.1 Loi des instants de défaillance

La loi de $T_n = \sum_{i=1}^{n} X_i$ est la loi de la somme de variables aléatoires indépendantes et de lois exponentielles. Si les λ_i sont tous distincts, on montre que cette variable aléatoire est de **loi hypoexponentielle**, définie par sa densité :

$$f_{T_n}(t) = \sum_{i=1}^{n} \frac{\prod_{j=1}^{n} \lambda_j}{\prod_{\substack{j=1 \\ j\neq i}}^{n} (\lambda_j - \lambda_i)} \exp(-\lambda_i t) \tag{6.1}$$

On a bien entendu $\mathbb{E}(T_n) = \sum_{i=1}^{n} \mathbb{E}(X_i) = \sum_{i=1}^{n} \frac{1}{\lambda_i}$.

6.1.2 Loi du nombre de défaillances survenues à chaque instant

En partant de $\mathbb{P}(N_t = n) = \mathbb{P}(T_n \leq t < T_{n+1}) = \mathbb{P}(T_n \leq t) - \mathbb{P}(T_{n+1} \leq t)$ et en utilisant (6.1), on obtient :

$$\mathbb{P}(N_t = n) = \sum_{i=1}^{n+1} \frac{\prod_{j=1}^{n} \lambda_j}{\prod_{\substack{j=1 \\ j\neq i}}^{n+1} (\lambda_j - \lambda_i)} \exp(-\lambda_i t) \tag{6.2}$$

Ce n'est pas une loi de probabilité usuelle et son espérance n'a pas d'expression simple.

6.1.3 Fiabilité et MTTF

L'hypothèse d'absence de mémoire de la loi exponentielle fait que la fiabilité et le MTTF à l'instant t ne dépendent que du nombre de défaillances survenues :

$$R_t(\tau; n, t_1, \ldots, t_n) = \exp(-\lambda_{n+1}\tau) \tag{6.3}$$

$$MTTF_t(n, t_1, \ldots, t_n) = \frac{1}{\lambda_{n+1}} \tag{6.4}$$

6.1.4 Fonction de vraisemblance

Dans la mesure où les X_i sont indépendantes et de lois exponentielles, pour un modèle ayant pour paramètre θ, la fonction de vraisemblance associée à l'observation de $x_1, \ldots, x_n$ est :

$$\mathcal{L}(\theta; x_1, \ldots, x_n) = \prod_{i=1}^{n} f_{X_i}(x_i) = \prod_{i=1}^{n} \lambda_i \exp(-\lambda_i x_i) = \left[\prod_{i=1}^{n} \lambda_i\right] \exp\left(-\sum_{i=1}^{n} \lambda_i x_i\right) \tag{6.5}$$

Pour aller plus loin, il faut faire des hypothèses sur la forme de λ_i. On va voir dans la suite de ce chapitre les deux plus connus des modèles ETBF, le modèle de Jelinski-Moranda et le modèle géométrique de Moranda.

6.2 Le modèle de Jelinski-Moranda

Le modèle de Jelinski-Moranda [9] est historiquement très important car c'est le tout premier modèle de fiabilité des logiciels, proposé en 1972. Il repose sur les hypothèses suivantes :

- Le logiciel contient à l'instant initial un nombre inconnu N de fautes.
- Quand une défaillance survient, la faute qui l'a provoquée est éliminée parfaitement.
- Aucune nouvelle faute n'est introduite.
- L'intensité de défaillance est proportionnelle au nombre de fautes résiduelles.

Soit ϕ le coefficient de proportionnalité. Au départ, le logiciel contient N fautes, donc $\lambda_1 = N\phi$. A la première défaillance, on corrige parfaitement la faute responsable, donc il reste $N-1$ fautes et $\lambda_2 = (N-1)\phi$. En continuant ainsi, on aboutit à $\lambda_i = (N-i+1)\ \phi$ pour tout $i \leq N$, d'où la définition suivante :

Définition 17 *Le* **modèle de Jelinski-Moranda** *(JM) est défini de manière équivalente par :*

- *Pour* $t < t_N,\ \lambda_t(n; t_1, \ldots, t_n) = (N-n)\ \phi,\ \phi \in \mathbb{R}^+, N \in \mathbb{N}$. *Pour* $t \geq t_N,\ \lambda_t = 0$.
- *Pour* $i \leq N$, *les* X_i *sont indépendantes et de lois respectives* $\exp\left((N-i+1)\ \phi\right)$.

N est le nombre de fautes initiales, donc c'est un paramètre entier. ϕ est un paramètre réel positif qui représente la qualité des corrections successives. Cette qualité est la même à chaque correction, comme on peut le voir sur la figure 6.2, qui représente une trajectoire de l'intensité de défaillance d'un modèle JM.

Pour obtenir les propriétés du modèle JM, il suffit de remplacer λ_i par $(N-i+1)\ \phi$ dans les expressions fournies dans la section précédente. Le résultat le plus remarquable est la simplicité de la loi du nombre de défaillances survenues à chaque instant, N_t.

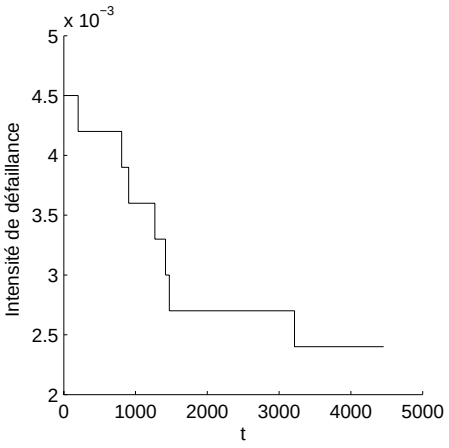


FIGURE 6.2 – Intensité de défaillance du modèle de Jelinski-Moranda

Proposition 4 *N_t est de loi binomiale $\mathcal{B}(N, 1-\exp(-\phi t))$.*

Démonstration : à faire en exercice.

Une variable aléatoire de loi $\mathcal{B}(N,p)$ modélise le nombre de fois où un évènement de probabilité p se produit en N expériences identiques et indépendantes. Par conséquent, tout se passe comme si les N fautes initiales étaient indépendantes et qu'elles avaient toutes une probabilité $1-\exp(-\phi t)$ de se manifester entre 0 et t. Autrement dit, la durée d'apparition (aussi appelée latence) d'une faute est une variable aléatoire de loi $\exp(\phi)$. Les fautes ont toutes le même taux de manifestation (elles ont la même probabilité de se produire sur un intervalle de temps donné) et celui-ci est constant.

La fonction moyenne est :

$$m(t) = \mathbb{E}(N_t) = N(1-\exp(-\phi t)) \tag{6.6}$$

Le nombre cumulé moyen de défaillances survenues croit donc comme la fonction de répartition d'une loi exponentielle. Une façon plus pratique d'exprimer cette propriété est d'écrire que :

$$\frac{d}{dt}\,\mathbb{E}(N_t) = N\phi\exp(-\phi t) \tag{6.7}$$

Ainsi, on peut considérer que le taux moyen d'occurrence des défaillances est une fonction exponentielle décroissante du temps.

En utilisant les propriétés générales des modèles ETBF, on montre également que :

- $\forall n \leq N,\ f_{T_n}(t) = n\ \phi\ C_N^n\ \exp(-N\phi t)\ (\exp(\phi t)-1)^{n-1}$.
- $\forall t < T_N,\ R_t(\tau; n, t_1, \ldots, t_n) = \exp(-(N-n)\phi\tau)$.
- $\forall t < T_N,\ MTTF_t(n, t_1, \ldots, t_n) = \dfrac{1}{(N-n)\phi}$.

La fonction de vraisemblance associée à l'observation de $x_1, \ldots, x_n$ est :

$$\begin{aligned}\mathcal{L}(N, \phi; x_1, \ldots, x_n) &= \left[\prod_{i=1}^{n} \lambda_i\right] \exp\left(-\sum_{i=1}^{n} \lambda_i x_i\right) \\ &= \left[\prod_{i=1}^{n} (N-i+1)\phi\right] \exp\left(-\sum_{i=1}^{n} (N-i+1)\phi x_i\right) \\ &= \phi^n \left[\prod_{i=1}^{n} (N-i+1)\right] \exp\left(-\phi \sum_{i=1}^{n} (N-i+1) x_i\right)\end{aligned}$$

Son logarithme est :

$$\ln \mathcal{L}(N, \phi; x_1, \ldots, x_n) = n \ln \phi + \sum_{i=1}^{n} \ln(N-i+1) - \phi \sum_{i=1}^{n} (N-i+1) x_i \tag{6.8}$$

d'où :

$$\frac{\partial}{\partial \phi} \ln \mathcal{L}(N, \phi; x_1, \ldots, x_n) = \frac{n}{\phi} - \sum_{i=1}^{n} (N-i+1) x_i \tag{6.9}$$

On en déduit que les estimateurs de maximum de vraisemblance de N et ϕ, $\hat{N}_n$ et $\hat{\phi}_n$, sont liés par la relation :

$$\hat{\phi}_n = \frac{n}{\sum_{i=1}^{n} (\hat{N}_n - i + 1) X_i} \tag{6.10}$$

Par ailleurs :

$$\frac{\partial}{\partial N} \ln \mathcal{L}(N, \phi; x_1, \ldots, x_n) = \sum_{i=1}^{n} \frac{1}{N-i+1} - \phi \sum_{i=1}^{n} x_i \tag{6.11}$$

Et on obtient que $\hat{N}_n$ est solution de l'équation implicite :

$$\sum_{i=1}^{n} \frac{1}{N-i+1} - \frac{n \sum_{i=1}^{n} X_i}{\sum_{i=1}^{n} (N-i+1) X_i} = 0 \tag{6.12}$$

qui se résoud par des méthodes numériques, comme celle de Newton-Raphson.

Pour les données de l'exemple, on obtient $\hat{\phi}_n = 1.71\ 10^{-4}$ et $\hat{N}_n = 34$. Comme $n = 24$, cela signifie qu'on estime qu'il reste encore 10 fautes dans le logiciel au moment de l'étude.

Les estimations de MTTF et de fiabilité à 100 heures sont $\widehat{MTTF}_{t_n} = 602.4$ h et $\hat{R}_{t_n}(100) = 84.7\%$, à comparer aux estimations équivalentes obtenues sous l'hypothèse de processus de Poisson homogène : $\widehat{MTTF} = 294.3$ h et $\hat{R}(100) = 72.0\%$. Cette différence très importante est logique : contrairement au HPP, le modèle JM fait l'hypothèse que la fiabilité croit, donc il est normal de trouver des estimations de fiabilité et MTTF nettement supérieures.

Le modèle de Jelinski-Moranda a été le premier modèle de fiabilité des logiciels et a été beaucoup utilisé. Cependant, il présente un certain nombre de défauts :

- Quand on a corrigé N défaillances, on a éliminé les N fautes initiales, donc le logiciel est théoriquement parfait, ce qui est peu plausible pour les logiciels complexes.
- Le problème principal est l'hypothèse que les fautes ont même taux de manifestation. En fait, certaines fautes se manifestent très rapidement et d'autres seulement après une longue période d'utilisation du logiciel : une faute peut survenir à chaque exécution et une autre une fois sur un million d'exécutions.
- Le choix d'un nombre de fautes initiales comme paramètre du modèle est contestable. Ce qui compte pour l'utilisateur d'un logiciel, ce n'est pas le nombre total de fautes, mais la fréquence d'occurrence des défaillances.
- Toute faute activée est supposée être complètement repérée et parfaitement corrigée. En fait ce n'est pas forcément le cas. Cela explique que le modèle de Jelinski-Moranda aura tendance à sous-estimer le nombre de fautes, donc à surestimer la fiabilité : le modèle JM est un modèle optimiste. La vérité doit donc se trouver quelque part entre les résultats fournis par les modèles HPP et JM.

6.3 Le modèle géométrique de Moranda

Le principal défaut du modèle JM est que les fautes sont supposées avoir toutes le même taux de manifestation, ce qui se traduit par le fait que la décroissance de l'intensité est toujours la même à chaque correction. En fait, les fautes ont des sévérités différentes. Les fautes les plus graves ont tendance à se produire très tôt, et leur correction va grandement diminuer l'intensité de défaillance. L'amélioration due aux corrections doit donc logiquement être forte au début de la période d'observation, puis de plus en plus faible. C'est le phénomène bien connu de perte d'impact des corrections au cours du temps.

Une façon de modéliser ce phénomène dans les modèles ETBF est de faire en sorte que les taux de défaillance successifs décroissent de manière géométrique et non plus arithmétique comme dans le modèle JM. On avait dans le modèle JM $\lambda_i = \lambda_{i-1} - \phi$. Pour obtenir une décroissance géométrique, il suffit de poser $\lambda_i = c\lambda_{i-1}$.

La croissance de fiabilité se traduira par $c < 1$. Remarquons que ce modèle pourrait être un modèle de décroissance de fiabilité en prenant $c > 1$, et qu'on retouve le processus de Poisson homogène ($\lambda_i = \lambda \ \forall i$) pour $c = 1$.

Alors $\lambda_i = c\lambda_{i-1} = c^2\lambda_{i-2} = \ldots = c^{i-1}\lambda_1$. On pose $\lambda_1 = \lambda$ et on obtient le modèle proposé par Moranda [14] sous le nom de "geometric de-eutrophication model".

Définition 18 *Le* **modèle géométrique de Moranda** *(GM) est défini de manière équivalente par :*

- $\lambda_t(n; t_1, \ldots, t_n) = \lambda\, c^n,\ \lambda \in \mathbb{R}^{+*}, c \in]0, 1]$.
- *Les* X_i *sont indépendantes et de lois respectives* $\exp(\lambda\, c^{i-1})$.

λ s'interprète comme l'intensité de défaillance initiale et c représente la qualité de la correction : plus c est petit, meilleure est la correction. $c = 1$ correspond à l'absence de correction.

La perte d'efficacité des corrections au cours du temps se visualise sur l'intensité du modèle GM (figure 6.3) par des sauts entre les paliers qui diminuent géométriquement.

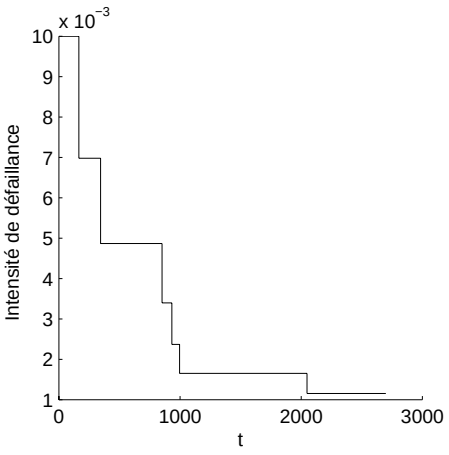


FIGURE 6.3 – Intensité de défaillance du modèle géométrique de Moranda

Les lois de probabilité de N_t et de T_n n'ont pas de forme simple. En particulier, on n'a pas d'expression utilisable pour leurs espérances. En revanche, la fiabilité et le MMTF sont très simples :

$$R_t(\tau; n, t_1, \ldots, t_n) = \exp(-\lambda c^n \tau)$$

$$MTTF_t(n, t_1, \ldots, t_n) = \frac{1}{\lambda c^n}$$

La fonction de vraisemblance est :

$$\begin{aligned}\mathcal{L}(\lambda, c; x_1, \ldots, x_n) &= \left[\prod_{i=1}^{n} \lambda_i\right] \exp\left(-\sum_{i=1}^{n} \lambda_i x_i\right) = \left[\prod_{i=1}^{n} \lambda\, c^{i-1}\right] \exp\left(-\sum_{i=1}^{n} \lambda\, c^{i-1} x_i\right) \\ &= \lambda^n c^{\sum_{i=1}^{n}(i-1)} \exp\left(-\lambda \sum_{i=1}^{n} c^{i-1} x_i\right) = \lambda^n c^{\frac{n(n-1)}{2}} \exp\left(-\lambda \sum_{i=1}^{n} c^{i-1} x_i\right)\end{aligned}$$

$$\ln \mathcal{L}(\lambda, c; x_1, \ldots, x_n) = n \ln \lambda + \frac{n(n-1)}{2} \ln c - \lambda \sum_{i=1}^{n} c^{i-1} x_i \tag{6.13}$$

$$\frac{\partial}{\partial \lambda} \ln \mathcal{L}(\lambda, c; x_1, \ldots, x_n) = \frac{n}{\lambda} - \sum_{i=1}^{n} c^{i-1} x_i \quad \text{d'où} \quad \hat{\lambda}_n = \frac{n}{\sum_{i=1}^{n} \hat{c}_n^{i-1} X_i} \tag{6.14}$$

$$\frac{\partial}{\partial c} \ln \mathcal{L}(\lambda, c; x_1, \ldots, x_n) = \frac{n(n-1)}{2c} - \lambda \sum_{i=1}^{n} (i-1) c^{i-2} x_i \tag{6.15}$$

On annule cette dérivée, on remplace $\hat{\lambda}_n$ par son expression en fonction de $\hat{c}_n$, et on obtient finalement que $\hat{c}_n$ est solution de l'équation implicite :

$$\sum_{i=1}^{n} (n - 2i + 1) \hat{c}_n^{i-1} X_i = 0 \tag{6.16}$$

Cette équation polynomiale se résoud facilement par la méthode de Newton-Raphson.

Pour les données de l'exemple, on obtient $\hat{\lambda}_n = 7.83\ 10^{-3}$ et $\hat{c}_n = 0.936$. Une valeur légèrement inférieure à 1 pour $\hat{c}_n$ traduit une légère croissance de fiabilité.

Les estimations de MTTF et de fiabilité à 100 heures sont $\widehat{MTTF}_{t_n} = 620.0$ h et $\hat{R}_{t_n}(100) = 85.1\%$. Ces valeurs sont proches de celles obtenues pour le modèle JM.

Chapitre 7

Les processus de Poisson non homogènes (NHPP)

7.1 Définition et propriétés

Définition 19 *Les* **processus de Poisson non homogènes** *(***NHPP** *pour Non Homogeneous Poisson Processes) sont les modèles de fiabilité des systèmes réparables pour lesquels l'intensité de défaillances ne dépend que du temps :*

$$\lambda_t(n; t_1, \ldots, t_n) = \lambda(t) \tag{7.1}$$

On supposera que $\lambda(t)$ est continue. Cela signifie qu'après la correction, l'intensité de défaillance est la même qu'avant la défaillance (voir figure 7.1).

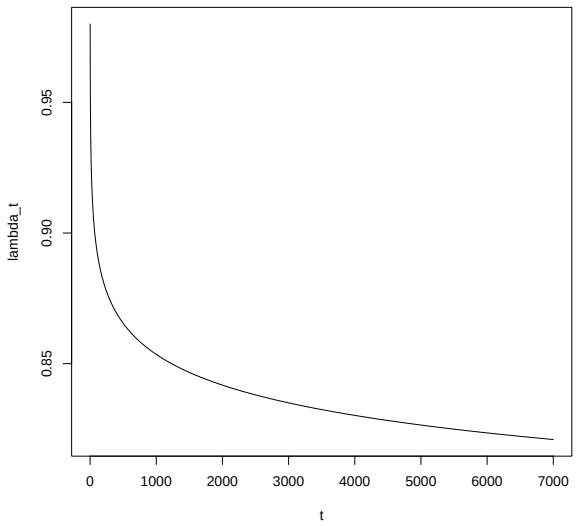


FIGURE 7.1 – Intensité d'un modèle NHPP

Par conséquent, la correction ne provoque pas de discontinuité dans l'intensité de défaillance, donc c'est comme si elle n'avait aucun effet.

C'est ce qu'on appelle le **principe de réparation minimale**. Si un système matériel est constitué d'un très grand nombre de composants et que l'un d'entre eux tombe en panne, son remplacement ou sa réparation ne va pas changer grand chose à l'intensité de défaillance (penser par exemple au remplacement d'un joint de robinet dans une centrale nucléaire). C'est aussi ce qui se passe quand une réparation se fait dans l'urgence suite à une défaillance, dans le but de remettre en fonctionnement le système mais sans le rendre plus fiable.

Pour le logiciel, cela revient à supposer que le programme contient un très grand nombre de petites fautes et que chaque correction n'en enlève qu'une toute petite partie.

On a dit qu'il était logique que la correction introduise une discontinuité dans l'intensité de défaillance. Donc a priori, les modèles NHPP ne devraient pas convenir. En fait, on peut considérer que ce sont des approximations qui peuvent se révéler très satisfaisantes sur des jeux de données.

L'intensité d'un modèle ETBF est une fonction en escalier et une fonction continue est une limite de fonctions en escalier. Par conséquent, on peut considérer un NHPP comme un cas limite ou une approximation d'un modèle ETBF. On reviendra plus tard sur les liens entre les différents types de modèles.

Le processus de Poisson homogène correspond à $\lambda_t(n; t_1, \ldots, t_n) = \lambda$, $\forall t > 0$. C'est donc à la fois un cas particulier de modèle ETBF ($\lambda_t(n; t_1, \ldots, t_n) = \lambda_{n+1}$) et de NHPP ($\lambda_t(n; t_1, \ldots, t_n) = \lambda(t)$).

Les propriétés générales des processus de défaillance permettent d'établir celles des NHPP.

7.1.1 Loi du nombre de défaillances survenues à chaque instant

Proposition 5 *N_t est de loi de Poisson $\mathcal{P}(m(t))$, où $m(t) = \int_0^t \lambda(u)du$.*

Ce résultat a donné son nom au processus de Poisson. Le processus est homogène quand $\lambda(t)$ est une fonction constante et non homogène sinon. On retrouve bien le résultat du HPP en prenant $\lambda(t) = \lambda$: N_t est de loi $\mathcal{P}(\lambda t)$.

Comme pour le HPP, on peut montrer en fait que le processus aléatoire $\{N_t\}_{t\geq 0}$ est à accroissements indépendants, c'est à dire que, pour $0 < s < t$, $N_t - N_s$ est indépendant de ce qui s'est passé avant s, et que $N_t - N_s$ est de loi $\mathcal{P}(m(t) - m(s))$.

Le nombre moyen de défaillances survenues entre 0 et t est $\mathbb{E}[N_t] = m(t)$, donc $\dfrac{d}{dt}\,\mathbb{E}[N_t] = m'(t) = \lambda(t)$. Par conséquent, l'intensité de défaillance d'un NHPP peut s'interpréter comme un taux moyen d'occurrence de défaillances.

Remarque. On a vu que pour le modèle JM, $\dfrac{d}{dt}\,\mathbb{E}[N_t] = N\phi\exp(-\phi t)$. Par conséquent, le NHPP d'intensité $\lambda(t) = N\phi\exp(-\phi t)$ (modèle GO, voir plus loin) peut être considéré comme une approximation du modèle JM au sens où une fonction continue est une approximation d'une fonction en escalier. On peut donc s'attendre en particulier à ce que les deux modèles donnent des estimations de fiabilité du même ordre.

De la même façon, on peut approcher n'importe quel modèle ETBF par un NHPP. L'intérêt est que le NHPP est beaucoup plus facile à utiliser que le modèle ETBF correspondant.

7.1.2 Loi des durées inter-défaillances et instants de défaillance

Par définition de l'intensité de défaillance, on a :

$$\lambda_t(n; t_1, \ldots, t_n) = h_{X_{n+1}|T_1=t_1,\ldots,T_n=t_n}(t - t_n) = \lambda(t) = h_{X_{n+1}|T_n=t_n}(t - t_n) \tag{7.2}$$

D'où $h_{X_{n+1}|T_n=t_n}(x) = \lambda(t_n + x)$, ce qui signifie que le taux de défaillance de la loi conditionnelle de X_{n+1} sachant $[T_n = t_n]$ est $\lambda(t_n + x)$. Tout se passe donc comme si le système avait un taux de défaillance initial $\lambda(t)$ et que les corrections n'avaient aucun effet. On retrouve ici l'idée de réparation minimale.

D'après (2.5), ce résultat peut s'écrire aussi :

$$\begin{aligned} f_{X_{n+1}|T_n=t_n}(x) &= \lambda(t_n + x) \exp\left(-\int_0^x \lambda(t_n + u)du\right) \\ &= \lambda(t_n + x) \exp\left(-\int_{t_n}^{t_n+x} \lambda(u)du\right) \end{aligned} \tag{7.3}$$

ou :

$$\mathbb{P}(X_{n+1} > x | \, T_n = t_n) = \exp\left(-\int_{t_n}^{t_n+x} \lambda(u)du\right) \tag{7.4}$$

On remarque que, contrairement au cas des modèles ETBF, les durées inter-défaillances X_i ne sont pas indépendantes.

En utilisant les résultats précédents, on montre facilement que la loi conditionnelle de T_{n+1} sachant $[T_n = t_n]$ est donnée par :

$$f_{T_{n+1}|T_n=t_n}(t) = \lambda(t) \exp\left(-\int_{t_n}^{t} \lambda(u)du\right) \tag{7.5}$$

$$\mathbb{P}(T_{n+1} > t | \, T_n = t_n) = \exp\left(-\int_{t_n}^{t} \lambda(u)du\right) \tag{7.6}$$

7.1.3 Fiabilité et MTTF

La fiabilité est donnée par :

$$R_t(\tau; n, t_1, \ldots, t_n) = \exp\left(-\int_t^{t+\tau} \lambda_u(n; t_1, \ldots, t_n)\, du\right)$$

Comme l'intensité ne dépend que du temps, on va noter la fiabilité $R_t(\tau)$ et on obtient :

$$R_t(\tau) = \exp\left(-\int_t^{t+\tau} \lambda(u)du\right) = \exp(-[m(t + \tau) - m(t)]) \tag{7.7}$$

Il n'y a pas d'expression simple pour $MTTF_t$.

7.1.4 Fonction de vraisemblance

Pour un NHPP, il est plus simple de calculer la fonction de vraisemblance à l'aide des instants de défaillance plutôt que des durées inter-défaillances. Pour un modèle ayant pour paramètre θ, la fonction de vraisemblance associée à l'observation de $t_1, \ldots, t_n$ est :

$$\begin{aligned}
\mathcal{L}(\theta; t_1, \ldots, t_n) &= f_{(T_1,\ldots,T_n)}(t_1, \ldots, t_n) = \prod_{i=1}^{n} f_{T_i|T_{i-1}=t_{i-1},\ldots,T_1=t_1}(t_i) \\
&= \prod_{i=1}^{n} f_{T_i|T_{i-1}=t_{i-1}}(t_i) = \prod_{i=1}^{n} \lambda(t_i)\ \exp\big(-\int_{t_{i-1}}^{t_i} \lambda(u)\,du\big) \\
&= \left[\prod_{i=1}^{n} \lambda(t_i)\right]\ \exp\big(-\int_{0}^{t_n} \lambda(u)\,du\big) \qquad (7.8)
\end{aligned}$$

Cette fonction est très simple, ce qui permettra d'estimer très simplement les paramètres des modèles NHPP. C'est en grande partie ce qui a fait leur succès auprès des praticiens, malgré le fait qu'ils soient moins réalistes que les modèles ETBF.

Pour aller plus loin, il faut faire des hypothèses sur la forme de $\lambda(t)$. On va voir dans la suite de ce chapitre les deux plus connus des modèles NHPP, le modèle de Duane et le modèle de Goel-Okumoto.

7.2 Le modèle de Duane

C'est le plus connu et le plus utilisé des NHPP. Il porte le nom de Duane [5] qui a observé empiriquement que ce modèle était bien adapté à des matériels électriques, et il a été étudié par Crow [4]. Il est connu en anglais sous le nom de Power-Law Process (PLP).

Définition 20 *Le* **modèle de Duane** *ou* **Power-Law Process (PLP)** *est le NHPP dont l'intensité est une puissance du temps :*

$$\lambda(t) = \alpha\beta t^{\beta-1}, \quad \alpha \in \mathbb{R}^+, \beta \in \mathbb{R}^+ \qquad (7.9)$$

Ce modèle est très souple au sens où il peut modéliser plusieurs types de tendance dans la fiabilité (voir figure 7.2) :

- $\beta > 1 \Longrightarrow \lambda(t)$ est croissante $\Longrightarrow$ décroissance de fiabilité
- $\beta < 1 \Longrightarrow \lambda(t)$ est décroissante $\Longrightarrow$ croissance de fiabilité
- $\beta = 1 \Longrightarrow \lambda(t)$ est constante $\Longrightarrow$ fiabilité stabilisée : processus de Poisson homogène

Le paramètre β s'interprète facilement comme le degré de dégradation ou d'amélioration du système. En fiabilité des logiciels, on s'attend à une croissance de fiabilité, donc à un paramètre β inférieur à 1. α est un paramètre d'échelle.

Il est clair que ce modèle est aux systèmes réparables ce que la loi de Weibull est aux systèmes non réparables. En particulier, X_1 est de loi de Weibull $\mathcal{W}(\alpha^{-1/\beta}, \beta)$.

Les propriétés du PLP se déduisent facilement des propriétés générales des NHPP.

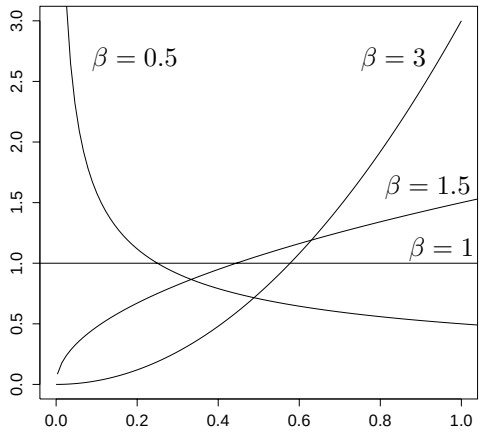


FIGURE 7.2 – Intensité du modèle de Duane pour différentes valeurs de β

- N_t est de loi $\mathcal{P}\left(\alpha t^{\beta}\right)$.
- $\mathbb{E}[N_t] = m(t) = \int_0^t \lambda(u)du = \alpha t^{\beta}$.
- $R_t(\tau) = \exp(-\alpha[(t+\tau)^{\beta} - t^{\beta}])$
- Il n'y a pas d'expression simple pour le MTTF, mais on peut le calculer numériquement à partir de $MTTF_t = \int_0^{+\infty} R_t(\tau)d\tau$.

La fonction de vraisemblance associée à l'observation de $t_1, \ldots, t_n$ est :

$$\begin{aligned} \mathcal{L}(\alpha, \beta; t_1, \ldots, t_n) &= \left[\prod_{i=1}^{n} \lambda(t_i)\right] \exp\Big(-\int_0^{t_n} \lambda(u)\, du\Big) \\ &= \left[\prod_{i=1}^{n} \alpha\beta t_i^{\beta-1}\right] \exp(-\alpha t_n^{\beta}) = \alpha^n \beta^n \left[\prod_{i=1}^{n} t_i^{\beta-1}\right] \exp(-\alpha t_n^{\beta}) \end{aligned} \tag{7.10}$$

$$\ln \mathcal{L}(\alpha, \beta; t_1, \ldots, t_n) = n \ln \alpha + n \ln \beta + (\beta - 1) \sum_{i=1}^{n} \ln t_i - \alpha t_n^{\beta} \tag{7.11}$$

$\dfrac{\partial \ln \mathcal{L}}{\partial \alpha} = \dfrac{n}{\alpha} - t_n^{\beta}$, qui vaut 0 pour $\alpha = \dfrac{n}{t_n^{\beta}}$.

$$\frac{\partial \ln \mathcal{L}}{\partial \beta} = \frac{n}{\beta} + \sum_{i=1}^{n} \ln t_i - \alpha t_n^{\beta} \ln t_n \tag{7.12}$$

En annulant cette quantité et en remplaçant α par $\dfrac{n}{t_n^{\beta}}$, on obtient :

$$\frac{n}{\beta} + \sum_{i=1}^{n} \ln t_i - n \ln t_n = \frac{n}{\beta} + \sum_{i=1}^{n} \ln \frac{t_i}{t_n} = 0 \tag{7.13}$$

$$\Longrightarrow \beta = \frac{n}{-\displaystyle\sum_{i=1}^{n} \ln \frac{t_i}{t_n}} = \frac{n}{\displaystyle\sum_{i=1}^{n-1} \ln \frac{t_n}{t_i}} \tag{7.14}$$

On obtient finalement :

Proposition 6 *Les estimateurs de maximum de vraisemblance des paramètres du modèle de Duane sont :*

$$\hat{\beta}_n = \frac{n}{\sum_{i=1}^{n-1} \ln \frac{T_n}{T_i}} \qquad et \qquad \hat{\alpha}_n = \frac{n}{T_n^{\hat{\beta}_n}} \tag{7.15}$$

Le fait que ces estimateurs aient une forme explicite est très rare en fiabilité des logiciels et explique en grande partie la popularité de ce modèle.

En remplaçant α et β par $\hat{\alpha}_n$ et $\hat{\beta}_n$ dans les expressions de $R_t(\tau)$ et $MTTF_t$, on peut estimer la fiabilité et le MTTF à n'importe quel instant.

Pour les données de l'exemple, on obtient $\hat{\alpha}_n = 0.100$ et $\hat{\beta}_n = 0.618$. Une valeur inférieure à 1 pour $\hat{\beta}_n$ confirme la croissance de fiabilité.

Les estimations de MTTF et de fiabilité à 100 heures sont $\widehat{MTTF}_{t_n} = 487.7$ h et $\hat{R}_{t_n}(100) = 81.2\%$. Ces valeurs sont intermédiaires entre celles obtenues pour les modèles JM/GM et celles obtenues sous l'hypothèse HPP.

Etant donné que le HPP est un cas particulier du modèle de Duane, si ce modèle avait été adapté aux données, l'estimation de β aurait été proche de 1. Comme ce n'est pas le cas, cela signifie que, conformément à l'intuition, le HPP n'est pas un bon modèle pour ces données. Mathématiquement, traiter ce problème c'est tester H_0 : "$\beta = 1$" contre H_1 : "$\beta \neq 1$" quand on suppose que le modèle de Duane est adéquat.

Il reste à déterminer si on peut considérer que les données proviennent du modèle de Duane. Pour cela, il faut utiliser des tests d'adéquation à ce modèle. Il en existe, mais leur étude dépasse le cadre de ce cours.

7.3 Le modèle de Goel-Okumoto

Ce modèle est basé sur les hypothèses suivantes [7] :

- Le logiciel contient à l'instant initial un nombre aléatoire N de fautes, dont l'espérance est a.
- Quand une défaillance survient, la faute incriminée est parfaitement corrigée et aucune nouvelle faute n'est introduite.
- L'intensité de défaillance est proportionnelle au nombre moyen de fautes résiduelles. Soit b le facteur de proportionnalité.

On constate que ces hypothèses sont similaires à celles du modèle de Jelinski-Moranda, sauf sur deux points. Tout d'abord, le nombre de fautes initial est une variable aléatoire et non plus une constante inconnue. Ensuite, l'intensité est supposée proportionnelle au nombre moyen de fautes résiduelles $\mathbb{E}[N - N_t]$ et non plus au nombre absolu (aléatoire) de fautes résiduelles $N - N_t$.

Pour le modèle JM (avec des notations différentes), les hypothèses aboutissaient à l'écriture $\lambda_t(n; t_1, \ldots, t_n) = \lambda_{n+1} = \phi(N - n)$. Ici, la troisième hypothèse implique que

l'intensité ne dépend que du temps. Donc le modèle est un NHPP et on a :

$$\lambda(t) = b\,\mathbb{E}[N - N_t] = b\ [a - \mathbb{E}[N_t]] \tag{7.16}$$

Comme $\mathbb{E}[N_t] = m(t) = \int_0^t \lambda(u)du$, on obtient en fait une équation différentielle :

$$m'(t) = b\ [a - m(t)] \tag{7.17}$$

Pour résoudre cette équation, on pose $g(t) = a - m(t)$. On a :

$$g'(t) = -m'(t) = -b\ [a - m(t)] = -b\ g(t) \tag{7.18}$$

D'où :

$$\frac{g'(t)}{g(t)} = -b \Longrightarrow \ln g(t) = -bt + cste \Longrightarrow g(t) = \exp(-bt + cste) = K\ \exp(-bt) \tag{7.19}$$

Puisqu'il n'y a pas de défaillance à l'instant initial, $g(0) = a - m(0) = a - \mathbb{E}[N_0] = a$. Donc $K = a$. Alors :

$$g(t) = a\ \exp(-bt) = a - m(t) \Longrightarrow m(t) = a\,[1 - \exp(-bt)] \tag{7.20}$$

et $m'(t) = \lambda(t) = ab\exp(-bt)$. Finalement :

Définition 21 *Le* **modèle de Goel-Okumoto** *(GO) est le NHPP dont l'intensité est une fonction exponentielle du temps :*

$$\lambda(t) = a\,b\,\exp(-bt),\quad a \in \mathbb{R}^{+*}, b \in \mathbb{R}^{+*} \tag{7.21}$$

L'intensité décroit donc comme une exponentielle alors que pour le PLP, elle décroit comme une puissance. A priori, par construction du modèle, b est un réel positif. On pourrait autoriser b à être négatif pour pouvoir modéliser une décroissance de fiabilité, mais il faudrait alors réécrire le modèle, par exemple sous la forme $\lambda(t) = \lambda\,\exp(-ct)$ avec $c \in \mathbb{R}$.

Comme on l'a dit plus haut, l'intensité du modèle GO a la même forme que le taux d'occurrence de défaillance du modèle JM, donc le modèle GO est l'équivalent NHPP du modèle JM.

Remarque : $\lim\limits_{t\to+\infty} \mathbb{E}[N_t] = \lim\limits_{t\to+\infty} m(t) = a$. Cela signifie que le nombre maximal de défaillances observables si on poursuit indéfiniment le test est a. C'est normal puisque a est le nombre moyen de fautes initiales et qu'on élimine une et une seule faute à chaque défaillance. On n'observera donc qu'un nombre fini de défaillances et le logiciel sera parfait en moyenne après la $a^{\text{ème}}$ correction. C'est le même phénomène que pour le modèle JM, avec les mêmes défauts.

La fiabilité est :

$$R_t(\tau) = \exp(-[m(t+\tau) - m(t)]) = \exp\left(-a\exp(-bt)[1 - \exp(-b\tau)]\right)$$

Le MTTF est :

$$MTTF_t = \int_0^{+\infty} R_t(\tau)d\tau = \int_0^{+\infty} \exp(-a\exp(-bt)[1-\exp(-b\tau)])d\tau$$

En $+\infty$, la fonction à intégrer vaut $\exp(-a\exp(-bt))$. Comme ce nombre est différent de 0, l'intégrale diverge et on a :

$$\forall t \geq 0, \quad MTTF_t = +\infty \tag{7.22}$$

Ce résultat signifie qu'à chaque instant, l'espérance du temps d'attente de la prochaine défaillance est infinie ! Ce phénomène apparemment paradoxal est en fait lié au résultat précédent. Etant donné qu'il n'y a qu'un nombre fini de défaillances possibles, la probabilité qu'à un instant t quelconque toutes les fautes aient été corrigées est non nulle. Or, si toutes les fautes ont été corrigées, le temps d'attente de la prochaine défaillance est bien infini. Si l'infini est possible, même avec une faible probabilité, alors le MTTF sera forcément infini. Par conséquent, ça n'a pas de sens d'estimer le MTTF dans ce contexte, ce qui est un handicap important à l'utilisation de ce modèle.

La fonction de vraisemblance associée à l'observation de $t_1, \ldots, t_n$ est :

$$\begin{aligned}
\mathcal{L}(a,b;t_1,\ldots,t_n) &= \left[\prod_{i=1}^{n}\lambda(t_i)\right]\ \exp\Big(-\int_0^{t_n}\lambda(u)\,du\Big) \\
&= \left[\prod_{i=1}^{n} ab\,\exp(-bt_i)\right]\ \exp(-a[1-\exp(-bt_n)]) \\
&= a^n b^n\ \exp(-b\sum_{i=1}^{n} t_i - a[1-\exp(-bt_n)])
\end{aligned} \tag{7.23}$$

$$\ln\mathcal{L}(a,b;t_1,\ldots,t_n) = n\ln a + n\ln b - b\sum_{i=1}^{n} t_i - a[1-\exp(-bt_n)] \tag{7.24}$$

$\dfrac{\partial \ln\mathcal{L}}{\partial a} = \dfrac{n}{a} - 1 + \exp(-bt_n)$, qui vaut 0 pour $a = \dfrac{n}{1-\exp(-bt_n)}$.

$$\frac{\partial \ln\mathcal{L}}{\partial b} = \frac{n}{b} - \sum_{i=1}^{n} t_i - at_n\exp(-bt_n) \tag{7.25}$$

On en déduit que :

Proposition 7 *Les estimateurs de maximum de vraisemblance des paramètres du modèle de Goel-Okumoto sont tels que $\hat{b}_n$ est solution de l'équation implicite :*

$$\frac{n}{\hat{b}_n} - \sum_{i=1}^{n} T_i - \frac{nT_n\exp(-\hat{b}_nT_n)}{1-\exp(-\hat{b}_nT_n)} = 0 \tag{7.26}$$

et

$$\hat{a}_n = \frac{n}{1-\exp(-\hat{b}_nT_n)} \tag{7.27}$$

Pour les données de l'exemple, on obtient $\hat{b}_n = 1.335\ 10^{-4}$ et $\hat{a}_n = 39.31$. Une valeur positive pour $\hat{b}_n$ confirme la croissance de fiabilité. La valeur de $\hat{a}_n$ indique qu'on estime qu'il y avait autour de 39 fautes initiales dans le logiciel. Donc, puisque $n = 24$ d'entre elles sont apparues, il en reste encore environ 15. Rappelons qu'avec le modèle JM, on avait estimé qu'il y avait 34 fautes initiales.

L'estimation de fiabilité à 100 heures est $\hat{R}_{t_n}(100) = 81.6\%$, qui est pratiquement identique à celle fournie par le modèle de Duane. Quant au MTTF, on ne peut pas l'estimer par autre chose que $+\infty$.

7.4 Autres modèles NHPP

7.4.1 Le modèle de Musa-Okumoto

C'est le modèle NHPP dérivé du modèle géométrique de Moranda. Son intensité [15] est :

$$\lambda(t) = \frac{\lambda}{1 + \lambda\gamma t}, \quad \lambda \in \mathbb{R}^{+*}, \gamma \in \mathbb{R}^{+} \tag{7.28}$$

Pour les données de l'exemple, on obtient $\hat{\lambda}_n = 7.03\ 10^{-3}$ et $\hat{\gamma}_n = 0.0547$. L'estimation de fiabilité à 100 heures est $\hat{R}_{t_n}(100) = 82.8\%$ et celle du MTTF est $\widehat{MTTF}_{t_n} = 559$ h.

7.4.2 Le modèle de Yamada-Ohba-Osaki

Les modèles précédents ont tous une intensité strictement monotone, croissante ou décroissante. Or, dans la pratique, on a souvent constaté que les défaillances étaient espacées au début de la période d'observation, puis de plus en plus rapprochées, pour finir par s'espacer à nouveau. Ce phénomène peut s'expliquer en phase de test par le fait que les testeurs qui découvrent un logiciel ne trouvent pas de défaillances tout de suite. Puis, à mesure que leur connaissance du logiciel s'approfondit, ils parviennent à localiser les fautes potentielles. Enfin, quand la majeure partie des fautes a été éliminée, les défaillances deviennent rares du fait de l'efficacité de la correction et du faible potentiel d'activation des fautes résiduelles.

Mathématiquement, cela revient à supposer que l'intensité de défaillance commence par croître, puis décroît, d'où le nom de modèles **en forme de S** (**S-shaped models**). Le modèle de Yamada-Ohba-Osaki (YOO) [21] est de ce type avec :

$$\lambda(t) = a\, b^2\, t\, \exp(-bt) \quad a \in \mathbb{R}^{+*}, b \in \mathbb{R}^{+*} \tag{7.29}$$

Pour les données de l'exemple, on obtient $\hat{b}_n = 5.58\ 10^{-4}$ et $\hat{a}_n = 26.5$. L'estimation de fiabilité à 100 heures est $\hat{R}_{t_n}(100) = 89.5\%$ et le MTTF est infini, comme pour le modèle GO.

Il existe de nombreuses variantes de ces modèles, pouvant prendre en compte quantités de paramètres comme la correction imparfaite, la correction différée, la dépendance entre les fautes, des paramètres environnementaux, une décomposition modulaire du logiciel, une classification des types de défaillances, le coût du test et de la correction, l'existence de ruptures, etc... [16].

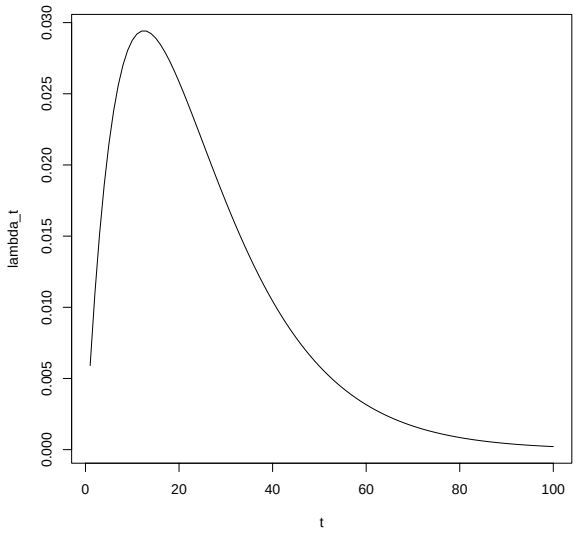


FIGURE 7.3 – Intensité d'un modèle de Yamada-Ohba-Osaki

7.5 Conclusion

Au final, on dispose de nombreux modèles de fiabilité qui permettent, au vu des défaillances et corrections successives d'un logiciel, d'évaluer sa fiabilité, son MTTF, le nombre de fautes résiduelles, etc... et de prévoir son comportement futur. A l'aide de ces estimations, on peut fournir un critère d'arrêt des tests, prévoir la maintenance, fixer des garanties, etc...

Devant la multiplicité des modèles de fiabilité possibles, le principal problème des praticiens est de choisir parmi ceux-ci celui ou ceux qui sont les plus appropriés pour modéliser une application précise et traiter un jeu de données particulier.

Le choix de modèles se déroule en 3 étapes :

- La première étape consiste à juger de la pertinence des modèles par rapport au problème étudié : validité des hypothèses.
- La seconde étape juge de la qualité intrinsèque d'un modèle. Est-il simple ? Les paramètres ont-ils une signification physique ? Peut-on les estimer facilement et avec précision ? Le modèle permet-il d'estimer correctement la fiabilité ?
- Enfin, il faut confronter le modèle aux données : peut-on admettre que les données sont issues du modèle en question ? La réponse à cette dernière question est un **test d'adéquation statistique**. Malheureusement, si les tests d'adéquation sont bien connus quand les observations sont des variables aléatoires indépendantes et de même loi, ce n'est pas le cas quand elles ne sont ni indépendantes, ni de même loi, ce qui se produit pour les données de fiabilité des logiciels. Néanmoins, certaines méthodes sont disponibles et permettent d'effectuer un choix pertinent de modèles.

Bibliographie

[1] Bon J.L., *Fiabilité des systèmes*, Masson, 1995.

[2] Norme CEI 61508, *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité*, Commission Électrotechnique Internationale, 2001.

[3] Cocozza-Thivent C., *Processus stochastiques et fiabilité des systèmes*, Springer, 1997.

[4] Crow L.H., Reliability analysis for complex repairable systems, in *Reliability and biometry - Statistical analysis of lifelength*, SIAM Philadelphia, 379-410, 1974.

[5] Duane J.T., Learning curve approach to reliability monitoring, *IEEE Transactions on Aerospace*, AS-2, **2**, 563-566, 1964.

[6] Gaudoin O. et Ledoux J., *Modélisation aléatoire en fiabilité des logiciels*, Hermès Science Publications, 2007.

[7] Goel A.L. et Okumoto K., Time dependent error detection rate model for software reliability and other performance measures, *IEEE Transactions on Reliability*, R-28, **1**, 206-211, 1979.

[8] Habrias H., *La mesure du logiciel*, Teknea, 1994.

[9] Jelinski Z. et Moranda P.B., Statistical computer performance evaluation, in *Software reliability research*, 465-497, W. Freiberger ed, Academic press, New-York, 1972.

[10] Laprie J.C. ed., *Dependability : basic concepts and terminology*, Springer, 1992.

[11] Laprie J.C. ed., *Guide de la sûreté de fonctionnement*, Cepaduès, 1996.

[12] Lawless J.F., *Statistical models and methods for lifetime data*, Wiley, 2003.

[13] Lyu M.R. ed., *Handbook of software reliability engineering*, IEEE Computer Society Press and Mc Graw-Hill Book Company, 1996.

[14] Moranda P.B., Event altered rate models for general reliability analysis, *IEEE Transactions on Reliability*, R-28, **5**, 376-381, 1979.

[15] Musa J.D. et Okumoto K., A logarithmic Poisson execution time model for software reliability measurement, *Proc. 7th Int. Conf. on Software Engineering*, 230-238, 1984.

[16] Pham H., *Software reliability*, Springer, 2000.

[17] Printz J., *Productivité des programmeurs*, Hermès-Lavoisier, 2001.

[18] Rausand M. and Hoyland A., *System reliability theory : models, statistical methods and applications*, Wiley, 2004.

[19] Vallée F., Fiabilité des logiciels, *Techniques de l'Ingénieur*, SE 2 520, 1-10, 2004.

[20] Vallée F. et Vernos D., Comment utiliser la fiabilité du logiciel comme critère d'arrêt du test, *13ème Colloque Européen de Sûreté de Fonctionnement ($\lambda\mu$13 - ESREL 2002)*, 164-167, 2002.

[21] Yamada S., Ohba M. et Osaki S., S-shaped reliability growth modelling for software error detection, *IEEE Transactions on Reliability*, R-35, **5**, 475-478, 1983.