



Analyse quantitative et prévision des fautes





Contexte

- **1 système avec potentiellement des mécanismes de tolérance aux fautes**
- **Ce que vous savez déjà faire normalement**
Identifier des limites dans le type de fautes et leur nombre telles que le système reste fiable, ou peut revenir dans un état fonctionnel.
- **Ce que l'on voit aujourd'hui :**
 - Modéliser l'incertitude sur l'occurrence des fautes via des probabilités
 - Modéliser le cycle de vie complet du système via un processus stochastique
 - Utiliser des modèles « sur étagère » pour faciliter certaines analyses (Chaînes de Markov, et PTA)



Rappel de probabilité

- Une variable aléatoire est définie pour un domaine D , la variable en elle-même est une fonction de certaines parties de D vers un espace probabilisé.
- Si votre variable a un nombre fini de réalisations, la fonction attribuant une probabilité à chacune est la distribution de masse
- Si votre variable a un nombre infini de réalisations, on parle de distribution de probabilité
- Deux lois sont cruciales de notre point de vue : le théorème de Bayes, et la somme des événements disjoints.



Occurrence d'une défaillance

■ Ce que l'on va modéliser :

- Système S, défaillance f
- 2 cas :
 - Le système peut défaillir de manière asynchrone à l'usage
 - Le système ne peut défaillir que lorsqu'il est sollicité
- 2 caractérisations
 - Probabilité de défaillir par demande
 - Distribution des dates de défaillances



Occurrence de défaillance à la demande

- **Modélisation triviale : Variable aléatoire booléenne indépendante de tout autre phénomène**
 - Caractérisation : 1 paramètre pour la loi de Bernouilli
 - Fiabilité = $1-p$
 - Défaillance = p
- **Modélisation plus avancée : exploiter la structure de sorte que l'occurrence d'une défaillance est caractérisée par $\text{pred}(X_1, \dots, X_n)$ où pred est un prédicat et X_1, \dots, X_n sont des variables aléatoires.**



Occurrence de défaillance « asynchrone »

- **Principe : X variable aléatoire réelle (positive) telle que X caractérise la date d'occurrence de la prochaine défaillance à partir de la date 0.**
- **Rappel : X peut être vue comme la caractérisation d'un processus d'arrivée. Ainsi, on peut définir une séquence X_1, \dots, X_n décrivant les différentes dates successives de défaillance (si cela fait sens)**
- **Le cas sympathique de la loi exponentielle**
 - Modélise des temps d'arrivée « sans mémoire »
 - Pas besoin de se souvenir de la date 0
 - 1 seule paramètre pour la caractérisation.



Rappel sur la loi exponentielle

- Pdf :

$$f(t) = 1/\lambda * \exp(-t/\lambda)$$

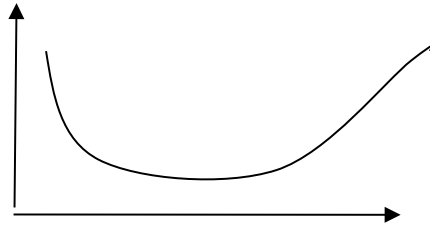
- Espérance :

$$1/\lambda$$

Idée : la probabilité qu'une défaillance se produise instantanément (ie en t et $t+h$ pour h tendant vers 0 est constante et égale à λ).

Le cas des loi à mémoire : Weibull

- Pb : faire l'hypothèse d'un taux constant s'applique assez mal au matériel



- Taux d'occurrence variable → courbe en baignoire
- Weibull : variation du taux de défaillance contrôlée par 3 paramètres... je n'en dirai pas plus car ces modèles sont propres aux matériel...



Prise en compte de mode de fonctionnement / du recouvrement de l'état défaillant

■ Problème :

un système tolérant aux fautes possède différents états de fonctionnement

→ différentes défaillances

→ Différents taux de défaillance

Solution

Modélisation par processus stochastique (vision séquentielle)

Modélisation par vecteur d'états (vision hiérarchique)

Modélisation mixte : vecteur de processus stochastiques

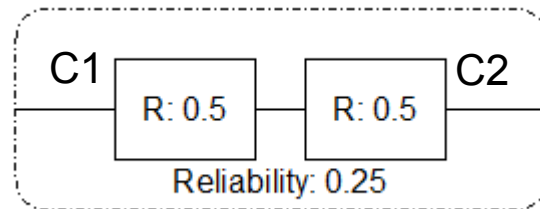


Propositional random formula et reliability block diagram

- **Question : peut on se donner un modèle graphique proche de l'architecture pour modéliser ces situations**
- **Idée : proposer des structure hiérarchisable et composable + une méthode d'évaluation de fiabilité**
- **Solution Reliability Block Diagram (RBD)**
- <http://pages.mtu.edu/~pjbonamy/rbdtool.html>
- **Principe :**
 - 1 block = 1 système avec un état binaire : fiable / non fiable
 - Une loi de probabilité pour la défaillance (Bernouilli paramètre p)
 - Des motifs composites de blocs : série, parallèle, K out of N....

Motifs de base, sémantique et évaluation la série

- Le bloc série (2 à n composants) :



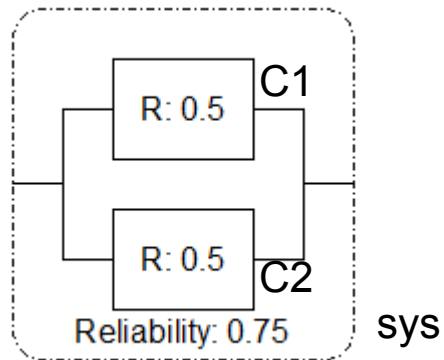
- Signification :
Le système requiert C1,C2 pour être fiable
- Formalisation
si f_x état booléen / vrai \Rightarrow x fiable
 $f_{sys} = f_{c1} \& f_{c2}$

- Interprétation numérique
 $R(C1) = P(f_{c1})$, $R(C2) = P(f_{c2})$
 $R(Sys) = P(f_{c1}) * P(f_{c2})$

- Hypothèse sous jacente
Indépendance des défaillances de C1 vs C2
- Pb possible un même composant ne peut apparaître à deux endroits possibles

Motifs de base, sémantique et évaluation la composition parallèle

- Le bloc parallèle (2 à n composants possibles):



- Signification :

Le système requiert
C1 ou C2 pour être fiable

- Formalisation

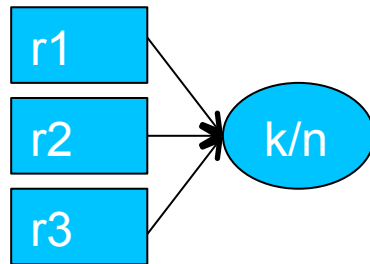
si fx état booléen / vrai \Rightarrow x fiable
 $f_{sys} = fc1$ ou $fc2$

- Interprétation numérique
 $R(C1) = P(fc1)$, $R(C2) = P(fc2)$
 $R(Sys) = P(fc1) + P(fc2) - P(fc1) * P(fc2)$

Motifs de base, sémantique et évaluation

le K parmi N

■ Le bloc série :



■ Signification :

Le système requiert
 $X = \{C_j \mid C_j \text{ fiable}\} / \text{Card}(X) \geq k$

■ Formalisation (peu d'intérêt)

Remarque : il y a une théorie sous-jacente pour automatiser les calculs

Chaque structure est associée à une formule logique à base de booléens !

La fiabilité d'une architecture
== une formule logique



Formalisation logique pure du RBD

■ Une architecture avec des noms ...

- chaque composant élémentaire se voit affecter un état booléen
- Les compositions série / parallèle / k parmi N définissent des formules logiques à partir des formules des blocs ou structures contenues
- Les variables d'état booléen sont vues comme des variables aléatoire 2 à 2 indépendantes

■ Avantages

- Utilisation de noms => un même composant peut être utilisé à deux endroits, ce n'est plus grave !
 - Indépendance des états élémentaires => règles de calcul simples si la formule possède une bonne propriété : forme normale de disjonction en conjonction disjointes
- ⇒ Cela fournit un cadre complet, possibilité d'utiliser des variables continues pour chaque état (date de passe de vrai à faux) le modèle devient « temporisé ».



Utilisation en direct de RBD pour fiabilité réplication passive / active à 3 répliques



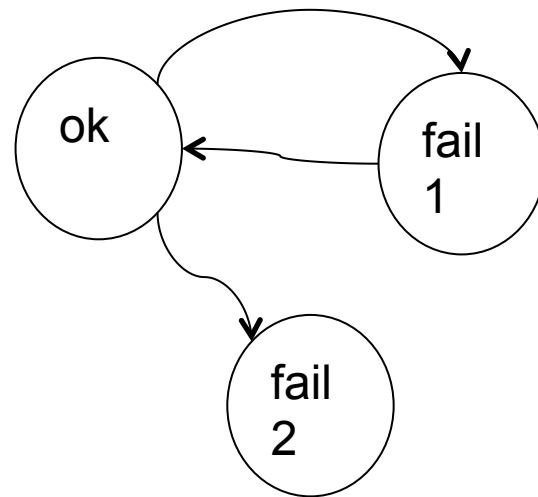
Modélisation fine des modes de défaillance

- **Problèmes :**
 - que faire modéliser un mode de défaillance réparable ?
 - Comment modéliser l'éventualité de deux modes de défaillance distinct à partir d'un même état ?
- **Solutions :**
 - Modéliser les changements d'états explicitement
 - Modèles pour l'analyse : Chaînes de Markov (discrètes et continues), et peut être les probabilistic timed automata

Rappel : automate et système à transition

Modèle Automate à état fini

- **Q** : ensemble des états
- **Δ** : ensemble des transitions (q,q') (changements d'états possibles)
- **I** : états initiaux
- **L** : étiquetage de Δ par des noms /des symboles





Chaîne de Markov en temps discret (pour un domaine fini)

Idée : l'état de votre système est aléatoire car
il y a une incertitude quantifiable à l'initialisation
il y a une incertitude lors des changements d'états

Chaîne de Markov discrète = processus stochastique :
 (X_i) , i dans N tel que

- chaque X_i est une variable aléatoire sur D .
- X_i représente l'état du processus à la date discrète i
- $P(X_i=v_i \mid X_{i-1}=v_{i-1}, X_{i-2}=v_{i-2}, \dots, X_0=v_0) = P(X_i=v_i \mid X_{i-1}=v_{i-1})$

Interprétation L'état i ne dépend que de l'état $i-1$, et de la probabilité conditionnelle $P(X_i=v_i \mid X_{i-1}=v_{i-1})$



Chaîne de Markov en temps discret (pour un domaine fini)

Vocabulaire et notations

$P(X_i=v_i \mid X_{i-1}=v_{i-1})$ est la probabilité de transition de v_{i-1} vers v_i

La définition des probabilité de transition et de la distribution de X_0 caractérise totalement la chaîne

Représentation graphique : système à transition étiqueté par les probabilités de transitions

Modélisation d'un cas pratique

3 états possible : ok failed1, failed2 tel que

A chaque transition

- Soit le système est « utilisé » avec pour effet un comportement normal, ou une défaillance fail1, ou une défaillance fail2.
- Soit, une réparation est tentée, avec pour effet un retour à l'état ok, ou un échec.
- Soit le système est dans l'état failed2 et reste définitivement défaillant

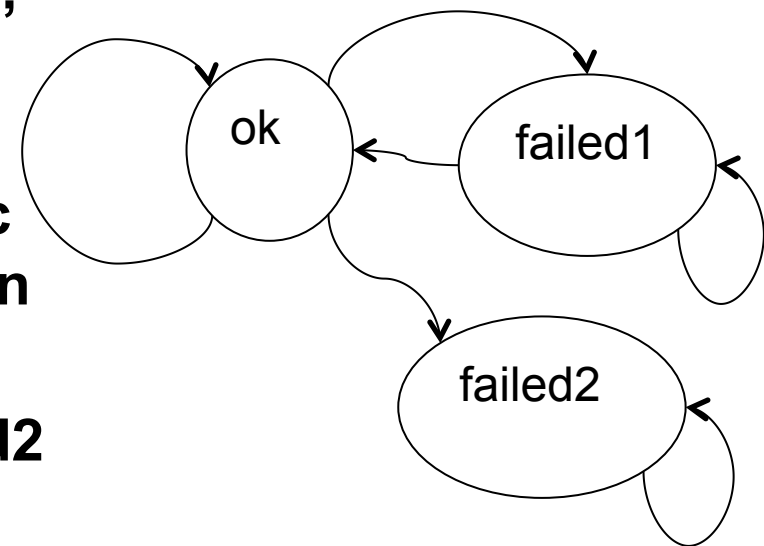
V_{i-1}	V_i	CPD
ok	ok	0.8
ok	failed1	0.15
ok	failed2	0.05
Failed1	ok	0.8
Failed1	Failed1	0.2
Failed2	Failed2	1

Modélisation d'un cas pratique

3 états possible : ok failed1, failed2 tel que

A chaque transition

- Soit le système est « utilisé » avec pour effet un comportement normal, ou une défaillance fail1, ou une défaillance fail2.
- Soit, une réparation est tentée, avec pour effet un retour à l'état ok, ou un échec.
- Soit le système est dans l'état failed2 et reste définitivement défaillant





Ce que l'on peut analyser

Probabilité stationnaire = vecteur π de probabilité

Tel que
$$\pi_i = S(v_i) = \lim_{n \rightarrow +\infty} \frac{\text{Card}(\{X_j = v_i \mid j \leq n\})}{n}$$

~ temps passé en moyenne dans l'état v_i pour chaque v_i

Probabilité transitoire = probabilité que X_i vaille v pour $i =$ une date précise, ou i dans un intervalle

Intérêt (exemple) : calcul la probabilité d'être fiable à la date i .



Evaluation de fiabilité

**Fiabilité à la demande = probabilité de transition
d'un état fiable vers un état fiable**

Que peut on étudier :

- **Probabilité de rester fiable pendant N instants**
- **Probabilité de ne pas subir un type particulier de défaillance sur un intervalle I d'instant.**

**Comment l'exprimer ? C'est une propriété
d'invariance.... On formalisera plus tard.**



Evaluation de la disponibilité

Disponibilité ~ ratio de temps passé dans un état fonctionnel

- ⇒ **Probabilité stationnaire** fournit une mesure de la disponibilité
- ⇒ **Probabilité transitoire** sur un intervalle autre mesure de disponibilité (i.e. probabilité d'être dans un état fonctionnel à un quelconque moment d'un intervalle donné).

PB = les transitions modélisent un temps logique (un nombre d'action), il faut que cela ait du sens !!!

Chaîne de markov en temps continu

Idée : dynamique en temps continue entre état discret =

Une séquence de paire (état discret, durée de séjour)

Chaîne de Markov continue = processus stochastique :

(X_i, T_i) , i dans N tel que

- **(X_i) , i dans N est une chaîne discrète de markov**
- **(T_i) , i dans N est une suite de variables aléatoires continues**
- **On définit $X(t)$ comme le X_k tel que t est dans $[T_k, T_{k+1}]$**
- **La chaîne est caractérisée par**

$$P(X(t+h)=v_i | X(t)=v_j) = q_{ji} h + o(h)$$

Principe : q_{ji} constante désignant le taux de transition de la valeur v_j vers v_i



Conséquence et interprétation

Hypothèses de travail:

- Taux de transition constant (chaine dite stationnaire)
- Domaine de valeur fini

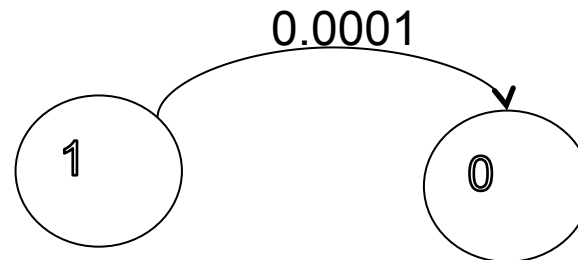
Conséquences :

- Temps de séjour dans l'état v_i suit une loi exponentielle de paramètre : $\sum_{j \neq i} q_{ij}$
- le taux de sortie est la somme des taux individuels de transition

Exemple

Systeme composé de 1 composant avec 2 états de fonctionnement : 1 (tout va bien) et 0 (défaillance par crash).

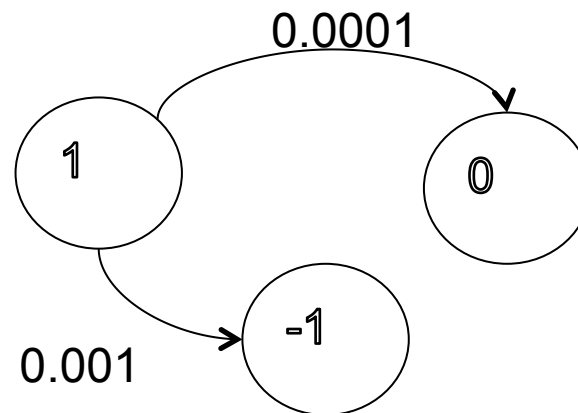
Le taux de transition de l'état 1 vers 0 est supposé constant et égal à 0.0001 h^{-1}



Exemple suite

Systeme composé de 1 composant avec 2 états de fonctionnement : 1 (tout va bien) et 0 (défaillance par crash), -1défaillance en valeur

Le taux de transition de l'état 1 vers 0 est supposé constant et égal à 0.0001 h^{-1} , le taux de transition de l'état 1 vers -1 est de 0.001 h^{-1}



le taux de sortie de 1 est de $0.0001+0.001$



Ce qu'il faut savoir

Analyse symbolique possible de la probabilité transitoire mais potentiellement fastidieuse (résolution pour chaîne stationnaire connue)

Possibilité d'automatiser l'analyse via des outils (non symbolique)

Limitations : les comportements sont non bornés dans le temps.



Modélisation de systèmes complexes par réseau de chaînes

Idée : 1 chaîne 1 état dans $[0, N]$.

\Rightarrow P chaînes P états dans $[0, N]$ évoluant en parallèle

Utilité : étude d'un système structuré complexe.

**Pb : comment modéliser les effets de type
« propagation de faute ? »**

Outil théorique : réseau d'automate / chaînes de markov synchronisées.



Systemes à transitions et produit synchronisés

Soit $A1$ et $A2$ deux systemes à transitions définis respectivement par

- l'ensemble d'état $Q1, (Q2)$
- l'ensemble de transition d'état $\Delta1, (\Delta2)$ sous ensemble de $Q1 \times Q1, (Q2 \times Q2)$
- L'ensemble d'étiquette $L1, (L2)$ tel que chaque transition est étiquetée par un élément de cet ensemble

Le produit synchronisé $A1 \times A2$ est un systeme à transition dont l'ensemble des états est $Q1 \times Q2$, dont les étiquettes sont dans $(L1 \cup L2)$ et dont les transition sont définies comme suit



Systemes à transitions et produit synchronisés (suite)

Pour chaque transition (q, p) de Δ_1 étiqueté l tel que l est dans L_1 / L_2 , alors pour tout état q' de Q_2 , la transition $((q, q'), (p, q'))$ est dans l'ensemble des transitions de $A_1 \times A_2$.

Pour chaque transition (q, p) de Δ_2 étiqueté l tel que l est dans L_2 / L_1 , alors pour tout état q' de Q_1 , la transition $((q', q), (q', p))$ est dans l'ensemble des transitions de $A_1 \times A_2$.

Pour toute étiquette dans l'intersection de L_1 / L_2 $((q', q), (p', p))$ est dans l'ensemble de transitions de $A_1 \times A_2$ ssi (q', p') est dans Δ_1 , et (q, p) est dans Δ_2 , et toutes les deux sont étiquetés l



Logique temporelle et calcul de probabilité

Idée : on peut évaluer la probabilité pour une chaîne

- **De passer par une séquence d'états**
- **D'être dans un état à une date précise**
- **D'être dans un ensemble d'état à une date précise**
- **....**

On peut évaluer la probabilité de n'importe quel événement qui est une contrainte sur « l'exécution de la chaîne »

Logique temporelle = contrainte sur la séquence d'états franchie (LTL), ou sur les transitions possibles en chaque états (CTL)

Logique temporelle et calcul de probabilité

Syntaxe :

Des contraintes instantanées : logique propositionnelle ou prédicat (sans quantification)

Des contraintes sur l'enchaînement d'état (non bornées):

$G \varphi$: φ doit être vraie en chaque instant futur

$F \varphi$: φ doit être vraie en au moins un instant du futur

$\Phi1 \text{ U } \Phi2$: $\Phi2$ doit être vraie au moins un instant dans le futur et $\Phi1$ doit être vrai jusqu'à ce que $\Phi2$ le soit.

Chaque opérateur peut être borné dans le temps via l'ajout d'un interval I après l'opérateur



Logique temporelle et calcul de probabilité

Une formule temporelle peut ensuite être soumise à un opérateur de probabilité : P

$P=? [\Phi]$ permet de demander le calcul de la probabilité que Φ soit vraie pour une chaîne donnée

$P(<|<=) c [\Phi]$ permet de déclencher la vérification de la formule déclarant que Φ est vraie avec une probabilité inférieure strictement (ou égale) à c pour une chaîne donnée



Outil Prism

Un outil pour modéliser les chaînes discrètes et continues

Un langage pour décrire des objectifs d'analyse stationnaires / transitoires

Une méthodologie pour analyser les résultats

Passage à la pratique