



**HAL**  
open science

# Utilisation des modèles graphiques probabilistes pour la mise en place d'une politique de maintenance à base de pronostic

Josquin Foulliaron

► **To cite this version:**

Josquin Foulliaron. Utilisation des modèles graphiques probabilistes pour la mise en place d'une politique de maintenance à base de pronostic. Calcul [stat.CO]. Université Paris-Est, 2015. Français. NNT : 2015PESC1205 . tel-01730669

**HAL Id: tel-01730669**

**<https://theses.hal.science/tel-01730669>**

Submitted on 13 Mar 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE DE DOCTORAT

Spécialité

**Signal, Image, Automatique**

École doctorale : Mathématiques et Sciences et Technologies de l'Information et de la  
Communication

Présentée par

**Josquin FOULLIARON**

Pour obtenir le grade de

**DOCTEUR de l'UNIVERSITÉ PARIS-EST**

Sujet de la thèse :

**Utilisation de modèles graphiques probabilistes pour la mise en  
place d'une politique de maintenance à base de pronostic**

soutenue le 13 novembre 2015

devant le jury composé de :

---

Patrice AKNIN	Directeur scientifique - SNCF ( <i>Directeur de thèse</i> )
Anne BARROS	Professeur HDR - NTNU ( <i>Codirectrice de thèse</i> )
Laurent BOUILLAUT	Chargé de recherche HDR - IFSTTAR ( <i>Encadrant de thèse</i> )
Philippe WEBER	Maitre de conférence HDR - Université de Lorraine ( <i>Rapporteur</i> )
Kamal MEDJAHHER	Maitre de conférence HDR - ENSMM, Besançon ( <i>Rapporteur</i> )
Bruno CASTANIER	Professeur des universités - Université d'Angers ( <i>Examineur</i> )

---



# Remerciements

Je tiens d'abord à remercier Philippe Weber et Kamal Medjaher pour avoir accepté de rapporter cette thèse ainsi que Bruno Castanier pour avoir accepté de l'examiner. J'ai vraiment pris plaisir à présenter puis à discuter avec eux durant ma soutenance. Je tiens à remercier également Anne Barros qui a codirigé cette thèse tout au long de ces 3 années. Merci pour les nombreuses discussions que j'ai eu avec elle, et pour le temps qu'elle a accepté de me consacrer malgré son déplacement en Norvège. Son regard extérieur sur mes travaux a toujours été très enrichissant.

Mes remerciements vont ensuite à Patrice Aknin qui a dirigé cette thèse. L'ensemble de ses remarques et idées a particulièrement contribué à la qualité du travail effectué. Merci enfin à mon encadrant, Laurent Bouillaut, qui a suivi cette thèse tout au long de ces 3 années, pour l'ensemble des conseils, idées et corrections qu'il a apporté, mais aussi pour la patience dont il a toujours fait preuve. Je le remercie également de la confiance qu'il a porté à mes travaux, et c'est avec plaisir que je continuerai de collaborer avec lui après cette thèse.

Un grand merci à tous les super collègues qui sont passés par le Bureau C113 tout au long de ces 3 années (Andry, David, Zineb, Nassim, Nadir, Rony, Baptiste, Florence). Ils ont permis d'assurer une atmosphère drôle, conviviale et enrichissante, ce qui a grandement participé au très bon déroulement de cette thèse.

Merci à l'IFSTTAR, l'ASCE et surtout à Manuel d'avoir permis la création d'une salle de musique disposant notamment d'un piano droit. A ce titre, merci aux collègues du groupe de musique Manuel, Cyril, Olivier, Philippe, Pierre, Rémi, Jean-François, Janelle et Noura avec qui j'ai pu organiser plusieurs séances et concerts de musique faisant contraste avec les moments ennuyeux de la thèse.

Merci à Allou pour le temps qu'il m'a consacré aussi bien pour m'aider sur l'apprentissage des modèles de mélange que pour m'expliquer la nature de ses travaux. Merci à tous les doctorants et post-doctorants : Anne-Sarah, Mohammed, Matthieu, Dihya, Nicolas, Hani, Wissam, Johanna, Moncef, et aux collègues chercheurs des bureaux voisins, Céline, Olivier, Jean-Louis, Hugues, Michel, ainsi que tout ceux que je n'ai pas pu citer ici, et que j'ai croisés pendant les pauses ou au déjeuner et avec qui j'ai toujours pris plaisir à discuter.

Enfin pour terminer, merci à ceux de ma famille qui ont soutenu ce projet depuis le début ainsi qu'à celle qui m'a suivi et encouragé tout au long de ces trois années.



# Table des matières

<b>Notations et acronymes</b>	<b>4</b>
<b>Introduction générale</b>	<b>6</b>
<b>1 Vers une politique de maintenance prévisionnelle</b>	<b>10</b>
1.1 Cadre de travail : Contexte et problématique . . . . .	10
1.2 Les différentes politiques de maintenance . . . . .	11
1.3 Problématique de sûreté de fonctionnement . . . . .	15
1.4 Le concept de pronostic et les différentes approches existantes . . . . .	19
1.5 Modèles stochastiques pour les études de fiabilité . . . . .	24
1.5.1 Méthodes statiques . . . . .	24
1.5.2 Modèles dynamiques . . . . .	26
1.5.3 Conclusion . . . . .	28
<b>2 Objets et concepts mathématiques utilisés pour la modélisation de la maintenance prévisionnelle</b>	<b>30</b>
2.1 Les réseaux bayésiens dynamiques . . . . .	30
2.1.1 Présentation des réseaux bayésiens statiques . . . . .	31
2.1.2 Notion de d-séparation . . . . .	32
2.1.3 Apprentissage des paramètres . . . . .	38
2.1.4 Inférence dans les réseaux bayésiens . . . . .	40
2.1.5 Présentation des Réseaux Bayésien Dynamiques . . . . .	44
2.1.6 Inférence dans un RBD . . . . .	46
2.2 Construction du modèle de dégradation puis de maintenance . . . . .	48
2.2.1 Définition et point de vue général . . . . .	48
2.2.2 Chaîne de Markov . . . . .	50
2.2.3 Modèle graphique de durée . . . . .	51
2.2.4 Modèle VirMaLab . . . . .	53
2.2.5 Conclusion . . . . .	55
<b>3 Traduction du concept de pronostic par modèle graphique probabiliste</b>	<b>56</b>
3.1 Problématique . . . . .	56

3.2	Algorithme de pronostic . . . . .	59
3.2.1	Définitions et notations . . . . .	59
3.2.2	Explications et interprétations . . . . .	61
3.2.3	Calculs d'indicateurs . . . . .	66
3.3	Intégration de lois de temps de séjour conditionnelles (LTSC) . . . . .	69
3.3.1	Idée principale de l'approche . . . . .	69
3.3.2	Définition d'une nouvelle approche . . . . .	71
3.3.3	Représentation graphique de l'approche proposée . . . . .	74
3.3.4	Nouvelle version de l'algorithme de pronostic . . . . .	76
3.3.5	Récapitulatif . . . . .	80
3.4	Représentation de l'algorithme de pronostic complet par RBD . . . . .	81
3.4.1	Définition des nœuds . . . . .	83
3.5	Problématique de l'inférence . . . . .	89
3.5.1	Explications et commentaires . . . . .	93
3.5.2	Calculs d'inférence pour les probabilités conditionnelles . . . . .	94
3.5.3	Perspectives . . . . .	95
3.5.4	Conclusions . . . . .	98
<b>4</b>	<b>Application des modèles développés sur des données simulées</b>	<b>100</b>
4.1	Tests sur des données simulées . . . . .	100
4.1.1	Création d'une base de données d'observation . . . . .	100
4.1.2	Test de l'algorithme de pronostic . . . . .	105
4.2	Application à la gestion d'une flotte de véhicules . . . . .	113
4.2.1	Conclusions . . . . .	117
	<b>Conclusion générale</b>	<b>118</b>
	<b>Bibliographie</b>	<b>122</b>

# Notations et acronymes

EM : Algorithme Expectation-Maximization

IDP : Indépendant

LPC : Loi de probabilité conditionnelle

LTS : Loi de temps de séjour

LTSC : Loi de temps de séjour conditionnelle

MGD : Modèle graphique de durée

MGDC : Modèle graphique de durée conditionnelle

MGP : Modèle graphique probabiliste

MV : Méthode du maximum de vraisemblance

RB : Réseau Bayésien

RBD : Réseau Bayésien dynamique

RUL : Remaining useful life (Durée de vie restante avant la panne)

REX : Retour d'expérience

Sdf : Sûreté de fonctionnement

$f * g$  : Produit de convolution de  $f$  et  $g$

$Act^{(t)}$  : Nœud d'activation du diagnostic

$A_i$  : Temps restant avant la prochaine action de maintenance prévisionnelle sur le système  $i$

$\alpha_j^{max}$  : Temps de séjour maximal possible à l'état  $j$

$C^{(t)}$  : Nœud représentant le mode de dégradation réel actif à l'instant  $t$

$\tilde{C}$  : Nœud représentant le mode de dégradation actif estimé par l'algorithme de pronostic

$\delta_x$  : Fonction de Dirac au point  $x$

$\Delta t$  : Durée entre les deux derniers diagnostics disponibles

$\Delta^{(t)}$  : Nœud déterminant si il y a eu transition ou non et si des états ont été manqués

$D^{(t)}$  : Nœud représentant le résultat du diagnostic

$E(\Psi)$  : Moyenne de la distribution  $\Psi$

$F^{(t)}$  : Fonction de répartition de la première défaillance d'un système

$\perp$  : Symbole d'indépendance

$i_i$  : Temps d'indisponibilité lors de la maintenance du système  $i$

$\lambda(t)$  : Taux de défaillance à l'instant  $t$

$m_k$  :  $k$ -ième mode de dégradation

$oc$  : Observation courante  
 $op$  : Observation précédente  
 $N$  : Nombre d'états de dégradation du système considéré  
 $n_M$  : Nombre de modes de dégradation considérés  
 $N_{val}$  : Nombre de valeurs non nulles dans le potentiel  
 $w_k$  : Poids associé au mode  $k$   
 $pa(X_i)$  : Ensemble des nœuds parents du nœud  $X_i$   
 $pot(X)$  : Potentiel représentant la LPC du nœud  $X$   
 $\Psi_x$  : Loi de temps de séjour de l'état  $x$   
 $\Psi_{x/m}$  : Loi de temps de séjour à l'état  $x$  sachant le mode actif  $m$   
 $R(t)$  : Fiabilité du système à l'instant  $t$   
 $R^{(t)}$  : Nœud contenant la somme des temps de séjour dans les états non encore atteints  
 $S_x$  : Valeur du temps de séjour à l'état  $x$   
 $S^{(t)}$  : Variable représentant le temps de séjour restant dans l'état courant  
 $t_d$  : Instant de la première défaillance  
 $t_{oc}$  : Instant de l'observation courante  
 $T_{SEC}^{(t)}$  : Estimation du temps déjà écoulé dans l'état courant jusqu'à l'instant  $t$   
 $T_{SR}^{(t)}$  : Estimation du restant dans l'état courant avant la prochaine transition  
 $\theta_i$  : Paramètres du réseau bayésien  
 $T_{max}$  : Temps de séjour maximal possible dans un état  
 $U[a, b]$  : Distribution uniforme sur l'intervalle  $[a, b]$   
 $|X|$  : Complexité spatiale de  $X$   
 $X^{(t)}$  : Variable représentant l'état d'un système  
 $x_P$  : Etat de panne  
 $Y$  : Base d'observation  
 $y_i^{(l)}$  : Temps de séjour observé à l'état sur la  $l$ -ème séquence d'observation de la base  $Y$ .

# Introduction générale

Depuis les trente dernières années, les systèmes de transports se sont complexifiés et le nombre d'usagers a considérablement augmenté. C'est notamment le cas dans le domaine du ferroviaire où la maîtrise de la disponibilité des infrastructures et des matériels roulants représente un enjeu sans cesse grandissant pour répondre aux besoins en mobilité de nos sociétés modernes. Ainsi, les contraintes de disponibilité des matériels roulants sont toujours plus importantes et leurs sollicitations également croissantes. De plus, un certain nombre d'appareils mis en service il y a une trentaine d'années sont aujourd'hui en fin de vie et nécessitent donc des interventions de plus en plus fréquentes. Pour bien répondre à ces exigences en terme de disponibilité, une orientation possible réside dans l'amélioration des stratégies de maintenance. En partant du constat que toute intervention de maintenance induit nécessairement un temps d'indisponibilité du système plus ou moins long, et donc engendre des pertes économiques potentiellement importantes, il apparaît que les choix stratégiques retenus en terme de maintenance impactent également la sécurité des systèmes et de leur usagers. Le problème de l'optimisation de la maintenance consiste donc à la fois à optimiser la disponibilité du système, à réduire ses coûts de maintenance, tout en garantissant un haut niveau de fiabilité. Cela est donc pour les industriels un enjeu extrêmement important d'un point de vue économique et peut impacter la qualité de leur image sur le marché.

Historiquement, les premières approches de maintenance développées étaient uniquement de nature corrective. Avec l'augmentation de la complexité des systèmes, et des besoins de maximiser la disponibilité (notamment pendant la seconde guerre mondiale), la notion de maintenance préventive a été introduite et appliquée à la majeure partie des secteurs industriels. Pour cela deux approches principales ont longtemps été considérées :

- La maintenance systématique (MPS) qui permet de maîtriser la charge de maintenance mais dont le pas de remplacement est parfois difficile à optimiser en particulier pour des systèmes dont la distribution de durée de vie a une forte variance ou lorsque le temps moyen entre deux défaillances varie dans le temps.
- La maintenance conditionnelle (MPC) qui évite l'écueil de la surmaintenance en intervenant « juste à temps » mais ne permet pas toujours de maîtriser la charge de maintenance. Pour remédier à ces inconvénients, les acteurs industriels se sont récemment tournés vers une politique de maintenance préventive appelée : **maintenance prévisionnelle**.

S'appuyant, comme son nom l'indique, sur une prévision de l'évolution future du système, cette approche possède l'avantage de concilier les avantages des MPC et MPS, en cherchant à anticiper quand est-ce qu'il sera nécessaire d'intervenir « juste à temps ». Le laboratoire GRETTIA de l'IFSTTAR au sein duquel se déroule cette thèse a développé une méthodologie générique nommée VirMaLab (pour Atelier Virtuel de Maintenance) basée sur le formalisme des réseaux bayésiens dynamiques proposant des outils d'aide à la décision pour évaluer, comparer et optimiser des stratégies de maintenance pour des systèmes multi-états (éventuellement multi-composants). Cette approche permettait jusqu'alors de modéliser les stratégies de maintenance mêlant les aspects correctifs, systématiques et conditionnels.

Le besoin croissant des industriels pour de nouvelles approches préventives de la maintenance a conduit au sujet de cette thèse dont l'objet était le développement d'une politique de maintenance prévisionnelle pour des systèmes multi-composants, puis de représenter cette politique sous forme de réseaux bayésiens dynamiques pour pouvoir intégrer le concept de pronostic à la plateforme VirMaLab. Même si de plus en plus de systèmes possèdent de nos jours de nombreux capteurs permettant de disposer d'informations en ligne sur l'état de santé des systèmes, cette thèse traite spécifiquement des systèmes dont les états sont discrets et observables uniquement de façon périodique. De telles contraintes sont caractéristiques de nombreux systèmes du domaine ferroviaire pour lesquels le diagnostic ne peut se faire que par rapport à référentiel relatif à un espace d'états discrétisé, et que celui-ci n'est accessible qu'à certains instants donnés. Ceci est par exemple le cas pour un matériel roulant dont l'état de certains composants n'est accessible que lorsque celui-ci retourne périodiquement en atelier ou encore dans le cas d'une ligne de chemin de fer pour laquelle l'état de chaque coupon de rail n'est disponible que lors du passage périodique, sur l'ensemble de la voie, d'un véhicule de mesure.

Le manuscrit s'organise en 4 chapitres :

Après avoir posé quelques notions générales de fiabilité, le premier chapitre passera en revue les principales politiques de maintenance existantes en insistant sur les principaux avantages relatifs à l'approche de maintenance préventive prévisionnelle. Le concept de pronostic sera ensuite défini, ainsi que les principales approches existantes pour conduire de tels calculs en discutant notamment de leurs avantages et inconvénients. L'ensemble de ces considérations ainsi que les caractéristiques des systèmes étudiés permettront de justifier le choix de l'approche fiabiliste utilisée dans cette thèse.

Le second chapitre s'attache à présenter en détails le cadre de travail de cette thèse. L'ensemble du formalisme mathématique utilisé y est donc défini. Les réseaux bayésiens seront donc dans un premier temps introduits, ainsi que leur extension : les réseaux bayésiens dynamiques.

Le modèle de maintenance qui sert de points de départ à cette thèse est ensuite détaillé. Après avoir rappelé la définition des chaînes de Markov, et les hypothèses implicites à leur utilisation, un modèle semi-Markovien introduit par R. Donat dans [18] est présenté. Ce modèle nommé "Modèle graphique de durée" est un RBD permettant de représenter un processus de dégradation à partir d'un ensemble de lois de temps de séjour dans chaque état. Enfin, l'architecture générale du modèle de maintenance VirMaLab est présentée.

Le troisième chapitre introduit les principales contributions de cette thèse. Un algorithme de pronostic basé sur le modèle graphique de durée est tout d'abord proposé. Son but est de calculer la durée de vie résiduelle du système à partir d'un ensemble de distributions de temps de séjour dans chaque état, puis de mettre à jour cette estimation à chaque nouveau diagnostic disponible. Le besoin de réduire la variance de ces estimations et donc d'améliorer la qualité des calculs de pronostic a conduit à introduire, dans une seconde partie, un nouveau modèle de dégradation basé sur le concept de lois de temps de séjour conditionnelles. L'objectif de cette contribution est de mieux capter le processus de dégradation du système étudié dans le cas où celui-ci est soumis à plusieurs modes de dégradation coexistants ou lorsque le processus de dégradation du système est le produit de la dégradation de plusieurs sous-composants dont l'état n'est pas visible. Ces travaux ont donné lieu à une nouvelle version de l'algorithme de pronostic basé sur ce nouveau modèle de dégradation ainsi qu'à une version enrichie du modèle graphique de durée nommée modèle graphique de durée conditionnel (MGDC). Enfin, la troisième partie propose une structure graphique spécifique de RBD représentant les outils précédemment définis. La conduite de calculs sur ce RBD pose la question de la complexité des calculs d'inférence. Pour répondre à ce problème, un algorithme d'inférence spécifique permettant de réduire la complexité des calculs a été proposé et est détaillé en fin du chapitre 3.

Le quatrième chapitre présente un cas d'application permettant de mettre en évidence les possibilités de la démarche proposée ainsi que le type de résultats que l'on peut espérer obtenir à l'aide de ces travaux. L'algorithme de pronostic est donc testé sur des données simulées à partir de mélanges de lois de Weibull. Les résultats sont comparés en fonction du modèle de dégradation choisi. Une distribution de l'erreur associée aux prédictions est ensuite calculée par inférence. Enfin, un cas d'application de l'algorithme de pronostic à la gestion d'une flotte de matériel roulant est présenté. Pour cela, un premier algorithme de décision est proposé dont l'objectif est de calculer, à partir de la date de fin de vie estimée pour chaque système de la flotte, la date de la prochaine action de maintenance prévisionnelle de chacun d'entre eux.

Pour finir, en guise de conclusion, un récapitulatif de l'ensemble des travaux conduits dans cette thèse sera faite. Les principales difficultés rencontrées seront évoquées ainsi que les différentes perspectives envisagées pour y répondre.



# Chapitre 1

## Vers une politique de maintenance prévisionnelle

L'objectif de ce premier chapitre est de présenter les problématiques liées à la question de la maintenance afin d'expliquer les raisons qui ont poussé à choisir l'approche et les différents outils mathématiques utilisés dans cette thèse. Dans une première partie, les caractéristiques du cadre de recherche seront énoncées. Cela permettra de définir le vocabulaire et les principales notions qui seront utilisées tout au long de ce manuscrit. Puis, les différentes politiques de maintenance existantes seront décrites en discutant de leurs avantages et inconvénients. La deuxième partie s'intéressera précisément à la question du pronostic en faisant un état de l'art des principales méthodologies existantes pour justifier le choix de celles qui ont été utilisées.

### 1.1 Cadre de travail : Contexte et problématique

Comme cela a été dit dans l'introduction, l'objectif de cette thèse est de mettre en place une politique de maintenance optimale de type prévisionnelle pour les systèmes complexes. Les définitions qui vont suivre serviront tout au long du manuscrit.

Un système complexe est un ensemble de composants reliés entre eux par des flux d'informations d'entrée sortie. Chaque composant a pour but d'assurer une ou plusieurs fonctions données. Un système sera considéré en **panne** lorsque celui-ci n'est plus en mesure d'accomplir la fonction qui lui est attribuée. Un composant peut être soit dans un état de bon fonctionnement soit dans un état **défaillant** soit dans un état **de panne**. Dans le cas général, l'état de santé d'un système multi-composants est une fonction de l'état de tous ses composants. En effet, une panne d'un composant n'implique pas forcément une panne du système, cela dépend de la position du composant dans l'architecture du système. Un composant dont la panne implique une panne du système est nommé **composant critique**.

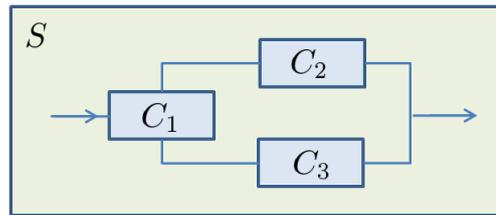


FIGURE 1.1: Système à trois composants montés en série-parallèle

*Exemple : Le système S représenté sur la figure 1.1 a 3 composants  $C_1, C_2, C_3$ . Le composant  $C_1$  est un composant critique. Une panne de ce composant coupera totalement le flux et le système sera alors en panne. En revanche, une panne du composant  $C_2$  ou  $C_3$  n'entraîne pas une panne du système puisque le flux peut toujours circuler. Le système fonctionne alors toujours sera considéré comme étant dans un état défaillant.*

Dans ce qui suit pour simplifier, le terme de système sera utilisé dans son sens global, en l'assimilant à un composant qui se dégrade au cours du temps.

Deux types de systèmes peuvent être distingués :

Ceux qui disposent de capteurs dédiés permettant d'avoir une surveillance en temps réel de leur état de santé à travers plusieurs indicateurs.

*Exemple : Le système locomotive dispose de dispositifs électroniques embarqués qui permettent de fournir différents descripteurs de son état de santé.*

Ceux dont l'état de dégradation n'est observable qu'à des instants discrets par l'intermédiaire d'un outil de diagnostic spécifique déployé ponctuellement ou par une observation humaine.

*Exemple : Un rail est un système purement mécanique, dont l'état de dégradation (la présence de défaut externe) est observable par l'utilisation d'un véhicule de diagnostic spécifique à ultrasons que l'on peut faire passer de façon périodique sur la voie.*

## 1.2 Les différentes politiques de maintenance

Une définition du terme de maintenance est donnée de façon assez précise dans la norme AFNOR X 60 000 [3] :

*”La maintenance est l’ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie du système, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise.”*

La maintenance consiste donc à réparer des équipements qui ne sont plus en mesure de satisfaire leur fonction principale ou bien à agir de façon à les maintenir dans un état fonctionnel. Une action de maintenance peut rendre le système indisponible pendant un certain temps. Cela génère un coût que l’on peut calculer de la façon suivante :

$$\text{Coût maintenance} = \text{Coût de la réparation} + \text{Coût d’indisponibilité}$$

Une bonne politique de maintenance doit donc à la fois garantir un niveau de fiabilité acceptable pour éviter l’apparition de pannes tout en évitant de faire de la sur-maintenance qui pourrait être inutilement coûteuse. Un tel compromis n’est pas facile à déterminer pour un système multi-composants possédant une architecture complexe. C’est pourquoi une organisation particulière de la maintenance a été proposée dans les années 80-90 basée sur la connaissance que l’on peut avoir de la fiabilité d’un système. Cette approche porte le nom de **maintenance basée sur la fiabilité** (MBF). L’objectif étant d’optimiser les paramètres de maintenance en trouvant un compromis entre les coûts d’entretien du système et sa disponibilité. Cela nécessite de définir :

- Un modèle de dégradation du système généralement probabiliste, ajusté à partir des données disponibles.
- Une modélisation des actions de maintenance envisagées et leurs effets sur l’état du système.
- Une fonction de coût permettant d’évaluer une stratégie de maintenance donnée.

Les différents types de maintenance se classent en deux grandes catégories : [3] [63]

- La maintenance corrective
- La maintenance préventive

L’approche **maintenance corrective** consiste à intervenir sur le système seulement lorsque que l’on constate que celui-ci est tombé en panne.

Dans l’approche **maintenance préventive**, on intervient à l’avance sur le système de façon à éviter qu’une panne ne survienne.

Dans chacune des deux approches précédentes, il existe plusieurs politiques de maintenance différentes qui vont être maintenant décrites en détail.

Pour ce qui concerne la maintenance **corrective**, on distingue deux types d'actions :

- La maintenance **curative** : C'est la première politique de maintenance qui est apparue dans l'histoire. Elle consiste, une fois que la panne est détectée, à effectuer une opération de maintenance qui va rétablir complètement le système dans un état fonctionnel. Il s'agit, soit d'une réparation importante, soit d'un remplacement complet du composant défectueux. Dans tous les cas, elle suppose une indisponibilité du système pendant un certain temps.

*Exemple : Soit une porte de quai d'une station de métro. La porte ne s'ouvre plus. Une alerte est envoyée au poste de commande central. Le technicien arrive et constate que la panne est due à une défaillance du moteur. Il remplace donc le moteur.*

- Une action **palliative** consiste à intervenir sur le système lorsqu'il est en panne mais pour effectuer une réparation temporaire minimale, juste suffisante pour redémarrer le système temporairement dans l'attente d'effectuer ultérieurement une réparation plus profonde de type curatif. Ce type d'action peut s'effectuer lorsque les pièces de rechange ne sont pas tout de suite disponibles ou quand une importante disponibilité du système est requise à un instant donné même en version dégradée.

*Exemple : Si un train tombe en panne sur une voie, on ne peut pas se permettre de bloquer la voie trop longtemps pour effectuer une grosse réparation, on effectue donc si possible une réparation minimale qui va permettre de redémarrer le train temporairement le temps de le ramener en atelier avant d'effectuer une grosse réparation nécessaire à une remise en service définitive.*

Ces deux politiques de maintenance de nature corrective, sont valables tant que les conséquences sécuritaires, environnementales et économiques d'une panne sont faibles ou que les coûts de réparation sont négligeables.

Ceci était le cas dans les processus de production traditionnels (au 19<sup>ème</sup> siècle) ou une machine en panne était rapidement remplaçable par une autre le temps d'être réparée. A partir des années 70 dans les secteurs à hauts risques tels que les transports (ferroviaire, aéronautique), la pétrochimie, le nucléaire, un état de panne ou un défaut important pouvait avoir des conséquences catastrophiques sur le plan sécuritaire, environnemental et économique. Ces entreprises sont donc les premières à s'être intéressées à des politiques de maintenance de type préventif.

De plus, le phénomène de mondialisation qui s'est accéléré depuis la fin des années 90 a

provoqué une augmentation des contraintes de production pour répondre à une concurrence toujours plus rude. Les entreprises ont donc cherché à optimiser au maximum leur cadence et leur rentabilité c'est-à-dire augmenter à la fois la quantité, la qualité tout en diminuant les coûts, et les délais de production. Une telle volonté d'optimisation de la productivité des systèmes est une des raisons majeures du développement des politiques de maintenance préventives dans tous les secteurs industriels [41].

Il existe principalement trois types de maintenance préventive.

- La maintenance **systematique** est une approche dans laquelle on se contente de remplacer les composants de façon périodique ou calendaire sans chercher à connaître dans quel état réel le système se trouve. Le choix d'une période optimale de remplacement est le problème fondamental de cette approche. Si la période choisie est trop grande, on prend le risque de subir la panne d'un composant remplacé trop tardivement. Au contraire, si la période est trop petite, on effectuera des remplacements inutiles c'est à dire de la surmaintenance coutant inutilement cher.

*Exemple : Sur le tronçon central du RER A, les rails sont changés de façon systematique tous les 2 à 4 ans.*

- La maintenance **conditionnelle** consiste à intervenir sur le système dès lors que l'état observé du système dépasse un seuil de dégradation considéré comme critique et nécessitant une action de maintenance rapide. Cela nécessite de connaître l'état courant du système. Soit le système est muni d'un dispositif de surveillance pouvant donner plusieurs indicateurs sur l'évolution de son état de santé en continu, soit le système n'est que périodiquement observable par l'intermédiaire d'une procédure de diagnostic. Dans ce dernier cas, le problème consiste à définir la fréquence du diagnostic déployé. La période d'auscultation optimale dépend de la dynamique de dégradation du système. Elle doit être correctement ajustée de façon à capter, sans trop de retard, les éventuels changements d'état. Celle-ci peut éventuellement varier dynamiquement en fonction de l'état observé [6].

En pratique, un grand nombre d'industriels utilise un compromis entre la maintenance systematique et conditionnelle. Les approches conditionnelles si elles permettent d'intervenir juste à temps, ont l'inconvénient majeur de conduire bien souvent à des actions de maintenance non anticipées, donc subies. Or, des actions de maintenance urgentes peuvent être jusqu'à deux ou trois fois plus coûteuses que des actions de maintenance planifiées. C'est pourquoi est apparu un nouveau

type de maintenance appelé **maintenance prévisionnelle**.

- La maintenance **prévisionnelle** (*predictive maintenance* en anglais) consiste à organiser les actions de maintenance en fonction d'une prévision de l'évolution future de l'état de dégradation du système. Ces prévisions pourront être mises à jour au fur et à mesure que de nouvelles observations concernant l'état de santé courant du système seront disponibles.

Cette dernière approche allie à la fois les avantages de la maintenance systématique et de la maintenance conditionnelle. Elle présente l'avantage de pouvoir anticiper les futurs besoins de maintenance des systèmes. Elle permet donc de préparer à l'avance l'organisation de la maintenance, d'anticiper les futurs investissements à venir et par voie de conséquence, d'optimiser la gestion des stocks, du personnel de maintenance, et la planification des tâches à venir, c'est à dire l'organisation de la production. Tous ces points sont essentiels pour un industriel.

Le tableau 1.1 ci-dessous regroupe l'ensemble des politiques de maintenance venant d'être présentées.

Maintenance corrective	Maintenance curative
	Maintenance palliative
Maintenance préventive	Maintenance systématique
	Maintenance conditionnelle
	Maintenance prévisionnelle

**TABLE 1.1:** Récapitulatif des politiques de maintenance

Le réglage d'une politique de maintenance nécessite donc de se baser sur un certain nombre d'indicateurs caractérisant l'état de dégradation du système. Le paragraphe suivant définit ceux qui sont le plus couramment utilisés.

### 1.3 Problématique de sûreté de fonctionnement

Les outils qui seront développés dans cette thèse sont conçus pour répondre aux problématiques du domaine de la sûreté de fonctionnement (SdF). Il s'agit d'un domaine d'étude présent dans tous les secteurs industriels dont l'objet est d'étudier la question de la défaillance

d'un système et de ses conséquences tant sur le plan sécuritaire qu'économique. De nombreux ouvrages abordent cette question [69][54]. Pour pouvoir évaluer le comportement global d'un système, un certain nombre de concepts ont été définis. Les définitions suivantes sont issues de la norme AFNOR X 60-319.

- Fiabilité : Aptitude d'une entité à accomplir une fonction requise, dans des conditions données, pendant un intervalle de temps donné.
- Maintenabilité : Dans des conditions données d'utilisation, aptitude d'une entité à être maintenue ou rétablie, sur un intervalle de temps donné, dans un état dans lequel elle peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et des moyens prescrits.
- Disponibilité : Aptitude d'une entité à être en état d'accomplir une fonction requise dans des conditions données.
- Sécurité : Aptitude d'une entité à éviter de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques.

Etudier l'ensemble de ces concepts a conduit à poser un certain nombre d'indicateurs. Soit un composant mis en service à la date  $t = 0$ . On note  $t_d$  l'instant de la première défaillance.

Le **MTTF** (*Mean Time To Failure*) est une quantité représentant le temps moyen avant l'apparition de la première défaillance. Le **MLT** (*Mean Logistic Time*) désigne quant à elle le temps écoulé entre l'apparition de la panne et celui du début de l'intervention de réparation. Cette réparation a une durée moyenne représentée par la quantité **MTTR** (*Mean Time To Repair*). La défaillance suivante permet de définir le temps moyen entre deux défaillances **MTBF** (*Mean Time Between Failures*). La figure 1.2 extraite de [10] illustre l'ensemble des quantités précédentes.

Les objets mathématiques les plus couramment utilisés en fiabilité sont maintenant rappelés.

La **fonction de répartition** de la première défaillance notée  $F(t)$  est une fonction de probabilité  $F : \mathbb{R}^+ \rightarrow [0, 1]$  utilisée en fiabilité pour représenter la proportion d'éléments déjà tombés en panne avant l'instant  $t$ .

$$F(t) = P(t_d \leq t)$$

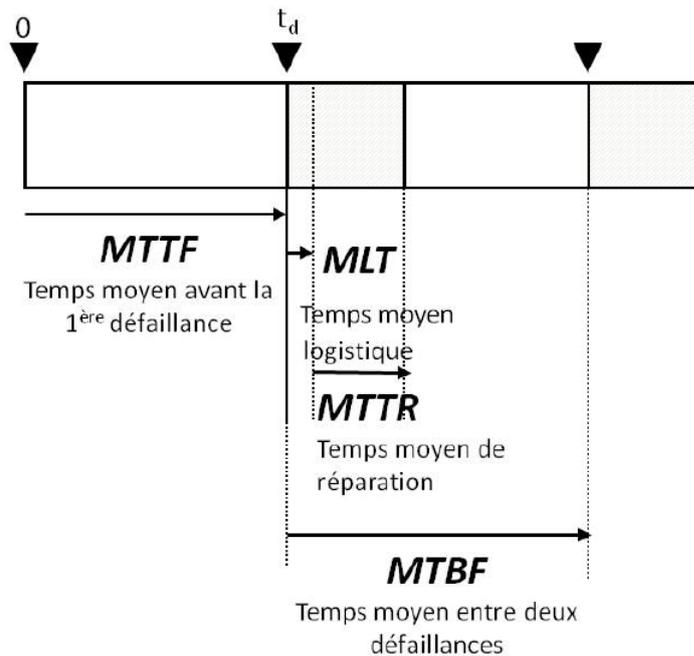


FIGURE 1.2: Illustration des principaux indicateurs utiles en sûreté de fonctionnement (tirée de [10])

On note  $f(t)$  la densité de probabilité associée, c'est à dire :

$$f(t) = F'(t) = \frac{dF(t)}{dt}$$

La **fiabilité** mathématique d'un composant à l'instant  $t$  est une fonction, souvent notée  $R(t) : \mathbb{R}^+ \rightarrow [0, 1]$  ; elle correspond à la probabilité que le système soit en état de bon fonctionnement à l'instant  $t$  ou de façon équivalente que  $t_d$  soit supérieur à  $t$ . De façon formelle :

$$R(t) = 1 - F(t) = P(t_d > t)$$

Le **MTTF** est défini comme l'espérance mathématique de la durée de vie du système.

$$MTTF = E[t_d] = \int_0^{+\infty} t \cdot f(t) dt = \int_0^{+\infty} R(t) dt$$

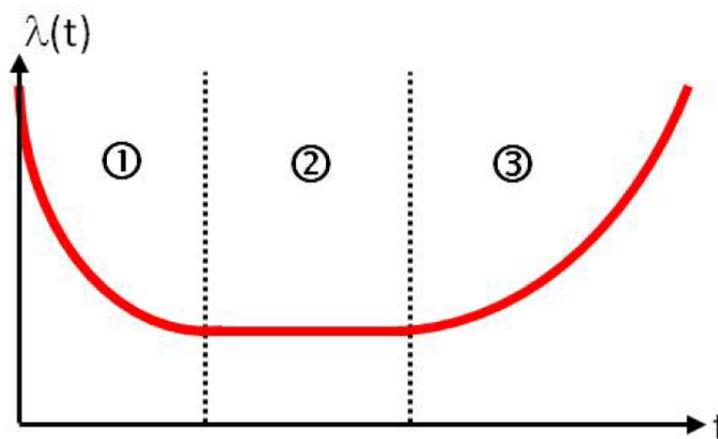
Il est souvent utilisée pour déterminer certains paramètres comme par exemple le pas de renouvellement des composants dans le cas d'une politique de maintenance systématique.

Le **taux de défaillance** instantané est défini par :

$$\lambda(t) = \frac{-R'(t)}{R(t)}$$

Classiquement, l'évolution du taux de défaillance d'un système peut se découper en trois phases représentées par la courbe en baignoire sur la figure 1.3.

La phase 1 est appelée phase de jeunesse, caractérisée par un taux de défaillance élevé lié aux défauts de jeune âge. La phase 2 correspond à la phase dite de maturité ayant un taux de défaillance assez faible et constant. Enfin, la dernière phase est appelée phase de vieillissement, car ayant un taux de défaillance assez élevé dû à l'usure et au vieillissement du système. L'ensemble de ces indicateurs va dépendre étroitement de la



**FIGURE 1.3:** Evolution classique du taux de défaillance, avec ses 3 phases de vie (figure tirée de [10])

vitesse de dégradation du système considéré ainsi que de la politique de maintenance qui a été décidée. Cette thèse s'intéresse à la mise en place de politiques de maintenance prévisionnelle. Comme expliqué précédemment, cette vision repose sur des prévisions concernant l'évolution de l'état de santé du système. Ce type de calculs de prédiction fait appel au concept de pronostic. Cette notion va être présentée dans la partie suivante ainsi que les principales approches existantes pour effectuer un calcul de pronostic.

## 1.4 Le concept de pronostic et les différentes approches existantes

Le pronostic est un concept assez récent qui s'est développé dans l'industrie avec le besoin d'adopter des politiques de maintenance prévisionnelles. C'est devenu un domaine de recherche scientifique de plus en plus important donnant lieu à de nombreux travaux [55] [61] [24][47][42][53] [1]. Il existe plusieurs façons de définir un calcul de pronostic. On peut par exemple le définir comme la probabilité que le système soit dans un certain état à un horizon de temps donné. Dans cette thèse, la définition suivante [49] qui est la plus courante sera considérée :

**Définition** : Un calcul de **pronostic** est le calcul de durée restante en fonctionnement avant défaillance du système (DEFAD, ou RUL en anglais pour *Remaining useful life*).

Il s'agit d'une estimation du temps restant avant que le système atteigne un état considéré comme « critique ». C'est à dire un état proche de la panne nécessitant obligatoirement une action de maintenance. Cette estimation est en général conditionnée à un niveau de dégradation courant.

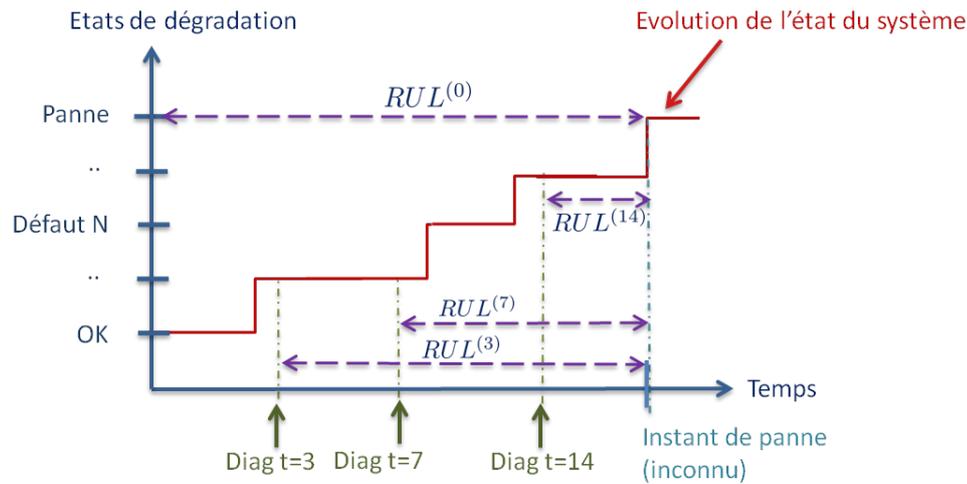
*On représente sur la figure 1.4 un système à états discrets prenant leurs valeurs dans  $\{OK, \text{defaut } 1, \dots, \text{defaut } N, \text{Panne}\}$ . A certains instants discrets ( $t=0,3,7,14$ ), un diagnostic de l'état réel du système est disponible. La RUL calculée à l'instant  $t$  (notée  $RUL^{(t)}$ ) correspond à la durée entre l'instant  $t$  et le futur instant de panne qui est inconnu. Cette estimation est améliorée chaque fois qu'un nouveau diagnostic est disponible.*

En notant  $X^{(t)}$ , l'état du système à l'instant  $t$  et  $x_P$  l'état de panne, la RUL est une quantité aléatoire se définissant formellement par :

$$RUL^{(t)} = \inf_{T, T > t} \{X^{(T)} = x_P\} - t \quad (1.1)$$

Un tel calcul repose de façon générale sur les informations suivantes :

- Une estimation de l'état courant du système à l'aide d'un dispositif de monitoring ou de diagnostic
- Une connaissance éventuelle de l'évolution des conditions de fonctionnement à venir (environnement, contexte d'utilisation, ..)
- Un modèle de dégradation
- La définition d'un seuil de dégradation considéré comme panne ou fin de vie



**FIGURE 1.4:** Illustration de la RUL dans le cas d'un système à états discrets

Il est notamment nécessaire de modéliser correctement la dynamique de dégradation. Il existe un très grand nombre d'approches dans la littérature pour la conduite d'un calcul de pronostic. Cette diversité est liée au fait que le pronostic est présent dans des domaines de recherche différents (mécanique, statistiques, intelligence artificielle..). Pour représenter la dynamique de dégradation, il faut dans un premier temps choisir un modèle mathématique puis apprendre les paramètres de ce modèle. Cet apprentissage de paramètres se fait à partir de l'ensemble des données disponibles concernant le système. Aborder le choix d'une méthode en fonction de la nature des données disponibles nous paraît l'approche la plus pertinente. Pour déterminer laquelle choisir, il convient de se poser des questions concernant les caractéristiques du système que l'on veut étudier. Les points les plus importants sont les suivants :

- Existe-t-il des modèles physiques décrivant la mécanique de dégradation interne du système ?
- Le système est-il muni de capteurs pouvant donner des informations sur son état de dégradation ?
- Si oui ces informations sont-elles disponibles en continu ou accessibles uniquement à certains instants ?
- Existe-t-il des bases d'historiques d'observation concernant le système ?

Les réponses à ces questions permettront de s'orienter vers une approche en particulier, voire d'en combiner plusieurs.

Les diverses méthodes de pronostic seront présentées en se basant sur la classification pyramidale que l'on trouve dans [14] ou [29] fondée justement sur cette vision. Selon cette classification, 3 types de méthodologies pour la conduite d'un calcul de pronostic peuvent être distinguées.

- Les approches basées sur les modèles physiques
- Les approches guidées par les données de surveillance
- Les approches dites fiabilistes et basées sur le retour expérience

Pour chacune d'entre elles, leur cadre d'application, les hypothèses qu'elles supposent ainsi que leurs avantages et inconvénients seront discutés.

1) L'approche modèle est utilisée lorsque l'on dispose d'expressions analytiques décrivant le processus de dégradation du système dont il est question [34] [35] [38]. En général, ces expressions sont des solutions d'équations différentielles ou de systèmes dynamiques issus du domaine de la physique. De par leur caractère déterministe, ces expressions sont extrêmement précises et nécessitent peu de données d'observation. La qualité du pronostic est très bonne et la marge d'erreur sur la prédiction très faible à condition que le modèle soit adapté et surtout son périmètre d'utilisation assez large. En pratique, dans la majorité des cas, de telles expressions sont extrêmement difficiles à obtenir. Elles sont soit très coûteuses car les paramètres des modèles mécaniques qui interviennent sont très difficiles à estimer, soit dans d'autre cas il n'existe tout simplement aucun modèle pour décrire de façon précise le comportement mécanique du système. De plus, ces expressions tiennent difficilement compte de la variabilité de possibles facteurs externes qui peuvent agir sur la dynamique de dégradation (qualité des matériaux, environnement physique, variation de la sollicitation, utilisation, etc. . .).

2) L'approche guidée par les données s'applique lorsque l'on a d'un dispositif de surveillance qui peut fournir des indicateurs en temps réel de certaines variables mesurables du système [59] [14] [39]. A partir de l'état de l'ensemble de ces descripteurs, une estimation de l'état de santé global du système est déduite. Ces données se présentent généralement sous la forme de séries temporelles. Il est alors possible d'effectuer des projections sur un horizon donné à l'aide de méthodes statistiques (régression, modèle ARMA..), et donc de faire des calculs de pronostic. Dans cette approche, il n'y pas de volonté précise de comprendre pourquoi le système se comporte comme observé. L'idée est d'observer le système à plusieurs instants donnés, de rechercher une tendance générale dans l'évolution des données, puis en faisant l'hypothèse qu'il n'y aura pas de discontinuité de comportement dans le futur, de faire une projection de l'évolution de l'état de santé sur la base de la tendance observée. Cette approche ne nécessite pas de connaissance propre du processus de dégradation. En revanche, comme les estimations sont uniquement basées sur les données entrantes, la notion de risque d'erreur n'est pas toujours bien intégrée. Elle ne peut pas prendre en compte dans ses prévisions des

phénomènes perturbateurs tant que ceux-ci n'ont pas encore été observés en ligne. De plus, il n'est pas toujours évident d'associer un niveau de santé général du système à un ensemble de descripteurs physiques (ex : température, pression, niveau de fissure).

3) L'approche basée sur le retour d'expérience est utilisée lorsque les seules données disponibles sur le comportement du système sont : les bases de retour d'expérience (REX) et les avis d'experts [52] [21]. Les bases REX sont des bases de données qui contiennent des historiques d'observations de l'état du système. Dans ce cas, le processus de dégradation du système est modélisé par un **processus stochastique** connu en fiabilité (processus Gamma, processus de Weibull), ou par des modèles graphiques probabilistes (réseaux neuronaux, chaînes de Markov, réseaux de Pétri, **réseaux bayésiens dynamiques...**). Les paramètres de ces modèles sont déterminés à partir d'algorithmes d'apprentissage statistiques appliqués sur les bases de données disponibles. Le principe de cette approche est de capter de manière probabiliste l'ensemble de l'aléa connu sur le comportement du système (c'est à dire l'aléa contenu dans la base d'observation disponible). Cet aléa est caractérisé par des distributions de probabilités. La variance de ces distributions permettent d'intégrer aux estimations de pronostic obtenues la notion de risque d'erreur et donc de leur associer des intervalles de confiance. Mais, pour obtenir une qualité d'apprentissage correcte, c'est-à-dire obtenir de bons paramètres, et espérer avoir un modèle suffisamment précis, il est indispensable d'avoir un volume conséquent de données d'observation du comportement du système. Ensuite, les paramètres appris, quand bien même seraient ils précis, sont des paramètres décrivant le comportement moyen du système durant la période d'observation couverte par la base de données disponible. Il est toutefois possible de procéder à un apprentissage en ligne qui intègre chaque nouvelle observation disponible du système comme variable d'entrée de l'apprentissage des paramètres. Dans ce cas, les paramètres d'apprentissage se mettent à jour de façon dynamique en fonction des nouvelles observations [56]. Cette approche bien que plus précise permet de tenir compte des évolutions observées mais n'est pas adaptée au cas de perturbations rapides. Il faut alors appliquer des algorithmes d'analyse de données, de détections de changement de régime, et éventuellement effectuer un apprentissage complémentaire dans une fenêtre temporelle réduite. Par ailleurs, lors de l'apprentissage calculant le comportement moyen du système, la qualité des paramètres estimés dépend fortement de la nature homogène ou non des données qui seront utilisées. Avec un apprentissage classique (méthode du maximum de vraisemblance), le processus de dégradation estimé intégrera une moyenne de l'ensemble des facteurs perturbateurs cumulés qui agissent sur ce processus et ce uniquement dans la fenêtre temporelle de la base REX. Cela ne permet pas d'identifier individuellement ces facteurs, ni de prendre en compte l'effet de la variation de l'un d'entre eux dans le présent comme dans le futur. Le mélange de l'ensemble de ces facteurs compose la variance des distributions de probabilités apprises. De plus les bases REX disponibles, sont longues à obtenir, et contiennent très souvent des données manquantes. La qualité des résultats obtenus

est donc très dépendante de la façon dont on effectue l'apprentissage, le nombre de variables explicatives que l'on choisit etc. Concernant les avis d'experts, leur pertinence va dépendre aussi du type de question qui vont être posées ; il convient donc d'être très prudent dans la façon dont on va traiter ces informations et leur associer des probabilités. Toutefois, un avantage important de cette approche est qu'elle peut aussi intégrer des données d'observations en ligne. Il est en effet possible de croiser des informations de diagnostic avec les modèles probabilistes appris hors-ligne et donc obtenir par des calculs de probabilités conditionnelles des estimations de pronostic tenant compte aussi de la tendance observée sur les données. Cela ne permet pas de se substituer à l'approche précédente basée sur les données car l'approche fiabiliste ne repose pas sur une modélisation de la dynamique propre des données, mais cela peut pallier aux divers inconvénients mentionnés précédemment et améliorer sensiblement la précision des calculs de pronostic effectués.

Pour résumer, la première approche repose sur la physique interne et théorique du système ; la seconde, traite de l'aspect temps réel et repose sur les observations en ligne du système ; la dernière tente d'apprendre et caractériser stochastiquement un comportement global du système à partir d'une fenêtre temporelle d'observation dans le passé. On remarquera que chacune des trois approches présentées précédemment n'est qu'une façon particulière de modéliser la dynamique de dégradation du système en fonction de la nature des données disponibles.

### Cadre de travail

Dans le cadre de cette thèse, on s'intéresse à des cas où les systèmes considérés sont à **états discrets** et **périodiquement observables**. Le système ne connaît pas d'auto-réparations, le processus de dégradation est donc monotone. On suppose que l'on ne dispose d'aucun modèle physique décrivant la mécanique de dégradation interne du système mais que des bases de données de retour d'expérience et des historiques d'observation sont disponibles. Cette configuration est par exemple fréquente dans le domaine ferroviaire. Il a donc été choisi de considérer une approche **hybride** basée à la fois sur les données de façon à intégrer au mieux les données d'observation disponibles ainsi que sur les modèles fiabilistes pour pouvoir exploiter les bases de données REX disponibles. La dégradation va donc être décrite par un modèle stochastique qui pourrait éventuellement prendre en compte des observations entrantes.

## 1.5 Modèles stochastiques pour les études de fiabilité

On peut diviser l'ensemble des méthodes stochastiques utilisées en fiabilité en 2 groupes : les méthodes statiques et les méthodes à trajectoire d'état temporelles dynamiques.

### 1.5.1 Méthodes statiques

Les méthodes statiques sont principalement de deux types :

- Les modèles de durée de vie qui s'attachent à donner une distribution de la date de fin de vie des systèmes sans considérer l'évolution dynamique de ces systèmes au cours du temps. Par conséquent, seule la date de fin de vie est considérée et celle-ci est caractérisée par une distribution de probabilité.
- Les arbres de défaillance simples dont le but est de représenter l'ensemble des relations cause-effet qui pouvant conduire un système à être en panne.

Dans cette thèse, on s'intéresse à des calculs quantitatifs de durée de vie, seuls les modèles de durées de vie seront abordés. Les distributions les plus couramment utilisées pour les modèles de fin de vie sont maintenant présentées. Certaines seront utilisées dans la partie application de cette thèse pour simuler les bases d'observation.

#### Le modèle exponentiel

Le modèle exponentiel est utilisé en fiabilité pour modéliser la distribution de la durée de vie du système avec la densité de probabilité suivante :

$$f : \mathbb{R}^+ \rightarrow [0, 1], f(t) = \lambda e^{-\lambda t}$$

ce qui conduit à la fonction de répartition suivante

$$F(t) = P(t_d \leq t) = 1 - e^{-\lambda t}$$

et donc  $R(t) = 1 - F(t) = e^{-\lambda t}$

Le taux de défaillance du système est alors constant :

$$\lambda(t) = \frac{-R'(t)}{R(t)} = \lambda$$

Cette distribution a de plus la propriété mathématique suivante :

$$\forall s > 0, P(t > t_d + s | t > s) = \frac{P(t > s + t_d)}{P(t > s)} = \frac{e^{-\lambda(t_d+s)}}{e^{-\lambda s}} = e^{-\lambda t_d} = P(t > t_d)$$

Ce modèle fait donc l'hypothèse selon laquelle la probabilité que le système tombe en panne après une date donnée  $t_d$  ne dépend pas de l'âge du système. En d'autres termes, ce modèle considère que toute observation de l'état du système n'a aucun impact sur la distribution de sa date de fin de vie. Cette hypothèse est difficilement acceptable en pratique. C'est la raison pour laquelle ce modèle est rarement utilisé. Cependant, nous verrons plus loin que certains modèles probabilistes comme les chaînes de Markov supposent implicitement l'utilisation de cette distribution de probabilité. Enfin, rappelons que celle-ci est une extension de la loi géométrique à un domaine continu.

### Le modèle de Weibull

Cette distribution est très souvent utilisée en fiabilité en raison de sa flexibilité pour modéliser la durée de vie d'un système du fait qu'elle dépend de trois paramètres. Sa densité de probabilité est :

$$f : \mathbb{R}^+ \rightarrow [0, 1], f_{\sigma, \beta, \gamma}(t) = \frac{\beta(t - \gamma)^{\beta-1}}{\sigma^\beta} e^{-\left(\frac{t-\gamma}{\sigma}\right)^\beta}$$

où  $\beta$  représente le paramètre de forme,  $\sigma$  le paramètre d'échelle et  $\gamma$  le paramètre de position. La fiabilité s'écrit alors :

$$R(t) = e^{-\left(\frac{t-\gamma}{\sigma}\right)^\beta}$$

et le taux de défaillance

$$\lambda(t) = \frac{\beta(t - \gamma)^{\beta-1}}{\sigma^\beta}$$

Ce modèle permet de modéliser finement les différentes phases de vie des courbes en baignoire comme celle représentée sur la figure 1.3. La phase de jeunesse sera correctement décrite avec un paramètre  $\beta < 1$  c'est à dire avec un taux de défaillance décroissant. Si  $\beta = 1$  on obtient alors  $\lambda(t) = \frac{1}{\sigma} = \frac{1}{MTTF}$  ce qui désigne un taux de défaillance constant correspondant à la période de maturité. On retrouve dans ce cas l'expression de la loi exponentielle présentée précédemment. Enfin un  $\beta > 1$  permettra de décrire la phase de vieillesse avec un taux de défaillance croissant.

## 1.5.2 Modèles dynamiques

Les modèles dynamiques tiennent explicitement compte de l'aspect temporel modélisant l'évolution de la dégradation du système dans le temps par des processus stochastiques. On présente ci-dessous les modèles les plus courants.

### Le processus Gamma

Le processus Gamma est très utilisé en fiabilité pour modéliser l'évolution de la dégradation d'un composant à espace d'états continu muni d'une relation d'ordre [47][48] [23] [46]. C'est une famille de variable aléatoire notée  $(X^{(t)})_{t \geq 0}$  ayant les caractéristiques suivantes :

- $X^{(0)} = 0$
- Pour tout incrément  $0 < s < t$ , l'accroissement  $X^{(t)} - X^{(s)}$  suit une loi gamma dont la densité est la suivante :

$$f : \mathbb{R}^+ \rightarrow [0, 1], f_{\alpha, \beta, t, s}(x) = \frac{1}{\Gamma[\alpha(t-s)]} \beta^{\alpha(t-s)} x^{\alpha(t-s)-1} e^{-\beta x}$$

$$\text{où } \Gamma(s) = \int_0^{+\infty} u^{s-1} e^{-u} du$$

$\alpha, \beta \in \mathbb{R}^+$  et sont nommés respectivement paramètre de forme et paramètre d'intensité. C'est un processus monotone croissant (car les accroissements sont toujours positifs). De plus, ces accroissements sont indépendants puisque leur distribution ne dépend que de la taille de l'accroissement en lui même ( $t - s$ ). Cette propriété est importante et donnera lieu à des discussions dans le chapitre 3. Les indicateurs de fiabilité et de pronostic s'écrivent par des formules analytiques complexes à base d'intégrales qui ne sont bien souvent pas directement calculables et donc approchées soit par des méthodes de Monte-Carlo soit par des schémas numériques discrets.

### Les modèles graphiques probabilistes (MGP)

L'autre grande catégorie de modèles stochastiques utilisés pour les modèles de dégradation et de maintenance est celle des modèles graphiques probabilistes (MGP). Il s'agit, dans ce cas, de représenter des distributions aléatoires à l'aide de graphes. Les modèles les plus couramment utilisés sont : les réseaux neuroflous [50], les chaînes de Markov [7], ainsi que les réseaux bayésiens (RB) et réseaux bayésiens dynamiques (RBD) [27][51]. Les réseaux bayésiens sont un formalisme très puissant pour représenter des systèmes multi-états contenant des relations de dépendance et d'indépendance conditionnelles

complexes. Ils peuvent d'autre part fournir une représentation intuitive de l'architecture de systèmes complexes multi-composants tout en pouvant mélanger dans un même modèle des variables aussi bien quantitatives que qualitatives. Ils peuvent aussi contenir des nœuds d'observations, qui permettent donc de prendre en compte des données de diagnostic. C'est pour cette raison que leur utilisation est de plus en plus fréquente dans le domaine de la sûreté de fonctionnement et de l'analyse des risques. [64][8][36] [30]. Une bonne rétrospective des travaux portant sur leur utilisation dans ce domaine peut être trouvée dans [65]. Leur extension, les RBD permettent d'ajouter l'aspect temporel à ce formalisme pour représenter des systèmes complexes multi-états évoluant au cours du temps ainsi que leurs éventuelles relations d'interdépendance. Ils peuvent notamment représenter des arbres de défaillance [9], modéliser une politique de maintenance [37][66] et commencent à être utilisés pour des calculs d'indicateurs de pronostic [39]. De plus, de nombreux algorithmes d'apprentissage et d'inférence sont disponibles [15][40][2][20] [67].

Ce formalisme a été choisi par l'équipe *Diagnostic et Maintenance* de l'IFSTTAR pour développer leur plateforme VirMalaB qui intègre à la fois un modèle de dégradation ainsi que la modélisation des processus de maintenance et de diagnostic (plus de détails seront donnés dans le chapitre 2). Le but de cette thèse étant d'enrichir cette plateforme de l'ensemble des politiques de maintenance prévisionnelles, le même formalisme a été choisi. Cela permettra facilement d'intégrer les outils de pronostic au modèle VirMaLaB.

De plus en plus de travaux sont conduits pour le développement de modèles de maintenances et notamment à base de calculs de pronostic. Nous en citons quelques uns :

Dans [55], un formalisme est introduit pour décrire l'architecture d'un système complexe en le considérant comme un ensemble de composants reliés entre eux par des flux d'entrée sortie. Chaque composant est caractérisé par un automate de mode. Le pronostic est vu comme, d'un mode de dégradation donnée, le chemin (séquence de modes) le plus probable jusqu'au mode défaillant. Le chemin doit être cohérent avec l'ensemble des événements passés et des observations. Les lois de vieillissement sont contenues implicitement dans les probabilités de transition de l'automate. Celles-ci approximent en général des lois de Weibull.

Dans [61], une approche basée sur les données est utilisée. L'ensemble des profils de dégradations possibles sont appris hors-lignes à partir de données d'observation issues de capteurs à espace d'états continus. Chaque profil est décrit par un modèle de dégradation représenté à l'aide d'une chaîne de Markov cachée à mélange de gaussiennes. Soit un système fournissant de nouvelles données d'observation en ligne, le modèle de dégradation appris le plus vraisemblable parmi ceux déjà appris est sélectionné. La

séquence d'états parcourue est reconstituée par l'algorithme de Viterbi pour déterminer l'état de dégradation courant associé à la dernière observation. L'état courant étant connu, le modèle de dégradation précédemment retenu permet de calculer l'ensemble des séquences d'états possibles conduisant à l'état de panne. A chaque chemin possible une RUL est associée en sommant les temps de séjour dans chaque état traversé composant la séquence. Les temps de séjour sont supposés indépendants et sont distribués selon une loi gaussienne. Il est donc possible de calculer facilement un intervalle de confiance autour des estimations de RUL. Le chemin le plus critique (le plus court) et le plus long, permettent le calcul d'une borne minimale et maximale pour la RUL.

Dans [33], le processus de dégradation est modélisé par un processus de Markov déterministe par morceau. C'est un processus à sauts où les sauts surviennent de façon aléatoire selon un taux constant, ou de façon déterministe. La RUL peut être exprimée par un calcul analytique souvent approché à l'aide d'une méthode de Monte-Carlo.

Dans [37], les auteurs utilisent un réseau bayésien dynamique pour modéliser un système multi-composant à états discrets ainsi que ses différentes actions de maintenance possibles. Ce système peut évoluer dans différentes conditions opérationnelles. Le processus de dégradation est considéré comme markovien. Différentes politiques de maintenance sont évaluées en calculant une distribution du nœud d'état à un horizon donné et en faisant jouer les paramètres de maintenance.

Dans l'ensemble de ces études, la RUL est à chaque fois un indicateur hors-ligne d'information non intégré dynamiquement dans le processus de modélisation. Il n'est donc pas relié directement à la décision de maintenance. De plus, les lois de temps de séjour sont bien souvent contraintes par des hypothèses concernant la forme des distributions. Le but de cette thèse est de considérer la RUL comme un indicateur dynamique ayant vocation à terme à être intégré au modèle de maintenance sans faire d'hypothèse sur la forme des lois de temps de séjour de chaque état de dégradation.

### 1.5.3 Conclusion

Ce chapitre a détaillé les problématiques liées à la sûreté de fonctionnement qui sous-tendent la question de l'optimisation des politiques de maintenance. En détaillant l'ensemble des politiques de maintenance existantes, nous avons montré les raisons qui ont amené à un développement des politiques de maintenance prévisionnelle et présenté les avantages que celles-ci peuvent apporter pour des industriels. Nous avons définis précisément le calcul de pronostic, puis à travers l'état de l'art, avons justifié le choix de l'approche par modèles stochastiques pour modéliser le processus de dégradation de

notre système. Parmi ceux-ci, les réseaux bayésiens dynamiques (RBD) ont été choisis en raison de leur capacité à s'adapter aux spécificités de notre cadre de travail.

Le prochain chapitre sera donc consacré précisément à la présentation du formalisme des RBD, puis à la description du modèle de dégradation qui a servi de point de départ à cette thèse.

## Chapitre 2

# Objets et concepts mathématiques utilisés pour la modélisation de la maintenance prévisionnelle

Ce chapitre présente en détail le formalisme mathématique qui a été utilisé pour construire le modèle de maintenance. La première partie de ce chapitre présentera le concept de réseau bayésien dynamique qui est l'objet mathématique au centre du modèle. Puis nous décrirons un RBD spécifique nommé "modèle graphique de durée" introduit par R. Donat [18] permettant d'améliorer la façon de modéliser la dynamique de dégradation d'un système. Enfin, le modèle de maintenance VirMaLaB basé sur les deux objets précédents sera présenté ; le but de cette thèse étant d'en proposer une extension intégrant le concept de pronostic.

### 2.1 Les réseaux bayésiens dynamiques

Les réseaux bayésiens (RB) ont été défini pour la première fois dans les années 1980 par J.Pearl [51]. Leur intérêt principal est de pouvoir représenter de façon intuitive la connaissance de type probabiliste. Les réseaux bayésiens **dynamiques** (RBD) sont une extension des RB permettant de considérer des situations qui, de plus, évoluent dans le temps. Pour faciliter la compréhension du concept de réseau bayésien dynamique, les réseaux bayésiens statiques seront dans un premier temps présentés. Le passage des RBs aux RBDs est naturel, un RBD pouvant être vu comme un simple RB de grande dimension. Dans cette thèse, ne sont utilisés que des réseaux bayésiens finis à domaine et à temps discret. Les définitions qui seront présentées se placeront donc dans ce cadre. Nous reviendrons par la suite sur les raisons qui ont motivé ces choix.

### 2.1.1 Présentation des réseaux bayésiens statiques

Un réseau bayésien est un modèle graphique représentant la loi jointe d'un ensemble de variables aléatoires. Il suppose un certain nombre de relations d'indépendance entre certaines variables, ce qui permet de représenter l'ensemble de ces variables et leurs relations à l'aide d'un graphe orienté.

Formellement, un réseau bayésien fini discret est défini par un couple  $\mathcal{G} = (G, \theta)$  avec :  $G = (X, A)$  : un graph orienté, dans lequel  $X = (X_1, \dots, X_n)$  désigne l'ensemble des sommets du graphe correspondant chacun à une variable aléatoire et  $A \subset (X \times X)$  l'ensemble des arêtes orientées du graphe.

$\theta = (\theta_1, \dots, \theta_n)$  est un ensemble de lois de probabilité conditionnelles (LPC) associées aux nœuds du graphe avec  $\theta_i = P(X_i | pa(X_i))$  la probabilité conditionnelle de la variable  $X_i$  sachant les variables de l'ensemble des nœuds parents de  $X_i$ .

*Exemple : La figure 2.1 est un RB représentant 4 variables (W,X,Y,Z). La flèche allant de Y vers Z signifie que la distribution de l'état de Z dépend de l'état Y. Ou encore, pour chaque état de Y, il y aura une distribution de l'état de Z différente. Il s'agit donc de la distribution de l'état de Z sachant l'état de Y. Les pointillés rouges désignent les probabilités correspondantes aux paramètres de chaque nœud. Le nœud Y a deux nœuds parents W et X. La loi de probabilité conditionnelle associée à ce nœud sera donc  $P(Y|W, X)$ . L'ensemble des LPC du graphe sont les paramètres du modèle.*

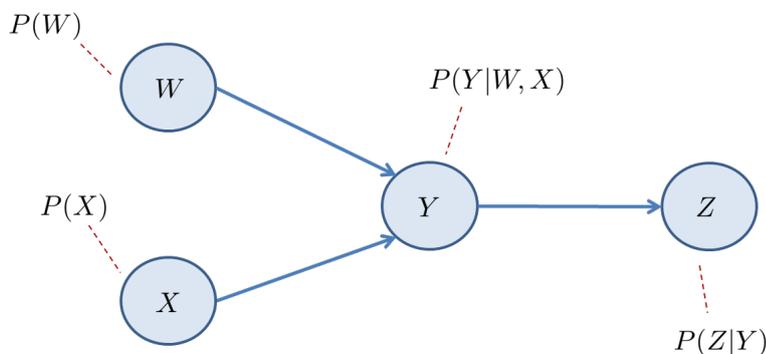


FIGURE 2.1: Exemple de réseau bayésien à 4 nœuds

Ce type de représentation induit des relations d'indépendance conditionnelles entre certaines variables en fonction de la topologie du graphe. La présence de telles relations est liée plus précisément à ce que l'on appelle la propriété de d-séparation. Cette notion va maintenant être détaillée car elle est déterminante pour conduire de façon optimale des calculs d'inférence dans les RB comme dans les RBD. Elle sera donc implicitement

utilisée dans la suite du manuscrit lorsque ces questions seront abordées dans le troisième chapitre.

### 2.1.2 Notion de d-séparation

Dans un graphe orienté, on peut rencontrer trois types de connexions élémentaires différentes entre les nœuds. La connexion en **série**, la connexion **divergente** et la connexion **convergente**.

Une connexion en série est représentée sur la figure 2.2. Pour mieux comprendre,

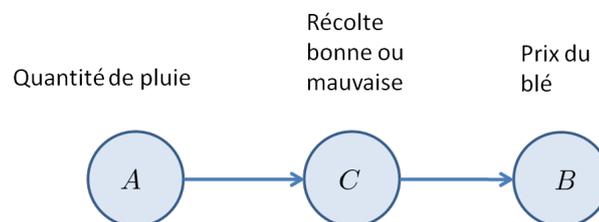


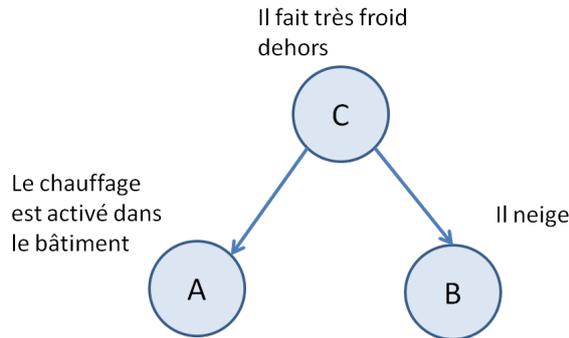
FIGURE 2.2: Connexion en série illustrée par un exemple

illustrons cette situation avec un exemple : Soit  $A$  représentant la quantité de pluie qui tombe au cours d'une année,  $C$  est l'événement à 2 états : la récolte est bonne ou mauvaise ; et  $B$  représente le prix du blé. Si l'on ne sait rien sur la qualité de la récolte, une information sur la quantité de pluie va modifier la distribution concernant le prix du blé. Les deux événements sont donc a priori dépendants. Maintenant si l'on sait que la récolte a été bonne, puisque le prix du blé dépend uniquement de cette donnée, savoir qu'il a plu ou non n'apporte aucune information en plus sur le prix du blé.  $A$  et  $B$  deviennent donc dans ce cas indépendants.

Le graphe impose donc la relation d'indépendance conditionnelle suivante :  $B$  est indépendant de  $A$  sachant  $C$  ce qui se note  $(B \perp\!\!\!\perp A | C)$ . La connaissance de  $C$  bloque tout flux d'information directe de  $A$  vers  $B$ .

Une connexion divergente est représentée sur la figure 2.3 .

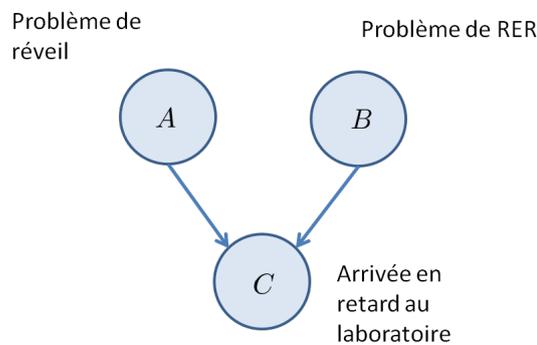
Soit  $A$  l'événement : "le chauffage est activé dans le bâtiment",  $B$  l'événement : "il neige" et  $C$  l'événement : Il fait très froid dehors. A priori, les événements  $A$  et  $B$  sont dépendants. En effet, si il neige dehors alors c'est qu'il fait probablement aussi très froid et que donc le chauffage a été mis dans le bâtiment. Le fait qu'il neige influence donc



**FIGURE 2.3:** Connexion divergente illustrée par un exemple

la probabilité que le chauffage soit mis donc  $p(A|B) \neq p(A)$ . Maintenant si l'on sait qu'il fait froid dehors. Comme le fait qu'il neige ne dépend que de cette condition, une information sur le chauffage dans le bâtiment n'apporte aucune information de plus sur la probabilité qu'il neige. Conditionnellement à  $C$ ,  $A$  et  $B$  sont donc indépendants. ( $A \perp\!\!\!\perp B|C$ ). Ici encore, la connaissance de  $C$  bloque toute circulation d'information entre  $A$  et  $B$ .

Une connexion convergente est représentée sur la figure 2.4.



**FIGURE 2.4:** Connexion convergente illustrée par un exemple

Soit  $A, B, C$  les nœuds représentant respectivement les événements : problème de réveil, problème sur la ligne de RER, arrivée en retard au laboratoire. A priori, la probabilité que le réveil tombe en panne un matin est complètement indépendante de celle que le RER subisse une panne quelconque. Maintenant, si l'on sait que l'arrivée au laboratoire a été tardive, ces deux événements deviennent dépendants. En effet, si le réveil n'est pas en panne, alors il est fortement probable que le retard soit dû au RER et dans ce cas, la pro-

tabilité que il y ait eu une panne de train est plus forte. On a donc  $p(B|C) \neq P(B|A, C)$  donc  $A$  n'est pas indépendant de  $B$  conditionnellement à  $C$ . Cette situation est donc différente des deux précédentes. La connaissance de  $C$  permet cette fois de faire circuler l'information entre  $A$  et  $B$ .

Pour résumer, dans le cas d'une connexion en série ou divergente,  $A$  et  $B$  sont a priori dépendants et une connaissance de  $C$  bloque le flux d'information donnant lieu à l'existence d'une relation d'indépendance conditionnelle alors que dans le cas d'une connexion convergente,  $A$  et  $B$  sont a priori indépendante et la connaissance de  $C$  autorise le passage d'information ce qui rend les événements dépendants.

La propriété de d-séparation n'est que la simple généralisation de ces 3 situations à un graphe quelconque.

**Définition (d-séparation) :**

Deux nœuds  $A$  et  $B$  sont d-séparés, si pour tout chemin non orienté allant de  $A$  à  $B$ , il existe au moins un nœud  $C$  satisfaisant au moins une des deux conditions suivantes :

- La connexion en  $C$  est en série ou divergente et la valeur de  $C$  est connue (donc bloque le chemin)
- La connexion en  $C$  est convergente et aucune information n'est disponible sur  $C$  ou sur ses descendants (ce qui assure l'indépendance de  $A$  et  $B$  dans ce cas là)

En d'autres termes pour que deux nœuds soient d-séparés, il suffit que tous les chemins possibles reliant ces nœuds soient bloqués. Cela conduit à la proposition suivante.

**Proposition :**

**Si deux ensembles de nœuds  $A$  et  $B$  sont d-séparés par un ensemble de nœud  $C$ , alors  $A$  est indépendant de  $B$  conditionnellement à  $C$ .**

$$(A \perp B)|C$$

Comme l'ensemble des parents d'un nœud bloque tout chemin entre ce nœud et ses non-descendants, la d-séparation entraîne donc la propriété suivante :

**Propriété :**

**Tout nœud est indépendant de ses non descendants conditionnellement à ses nœuds parents.**

Le paragraphe suivant montre comment il est possible d'en déduire la célèbre propriété dite de factorisation qui justifie tout l'intérêt des réseaux bayésiens.

Il est toujours possible de numéroter les nœuds d'un graphe dans ce qu'on appelle l'ordre topologique. C'est une numérotation telle que tout nœud du graphe a toujours un numéro plus élevé que ceux de ses parents. Supposons que le RB soit numéroté de cette façon. D'après la formule de Bayes, on a dans le cas général pour un ensemble de  $n$  variables aléatoires quelconques :

$$P(X_1, \dots, X_n) = P(X_1) \prod_{i=2}^n P(X_i | X_1, \dots, X_{i-1})$$

D'après l'ordre topologique, l'ensemble  $\{X_1, \dots, X_{i-1}\}$  ne contient aucun descendant de  $X_i$ . La propriété d'indépendance conditionnelle mentionnée précédemment induit donc :

$$X_i \perp\!\!\!\perp \{X_1, \dots, X_{i-1} / pa(X_i)\} | pa(X_i)$$

où  $\{X_1, \dots, X_{i-1} / pa(X_i)\}$  est l'ensemble des non-descendants de  $X_i$  qui ne sont pas ses parents. On rappelle que si 3 ensembles de variables aléatoires  $A, B, C$  sont tels que  $A \perp\!\!\!\perp B | C$  alors  $P(A|B, C) = P(A|C)$

Cela donne donc :

$$\forall i, P(X_i | X_1, \dots, X_{i-1}) = P(X_i | pa(X_i))$$

où  $pa(X_i)$  désigne l'ensemble des nœuds parents de  $X_i$ .

Cela permet d'en déduire la propriété fondamentale des réseaux bayésiens.

### Propriété de la loi jointe :

**Soit  $X_1, \dots, X_n$ , l'ensemble des nœuds d'un réseau bayésien discret fini. La loi jointe de l'ensemble des variables du réseau peut se factoriser de la façon suivante.**<sup>1</sup>

$$P(\mathbf{X}_1, \dots, \mathbf{X}_n) = \prod_{i=1}^n P(\mathbf{X}_i | pa(\mathbf{X}_i))$$

L'ensemble des LPC de chaque nœud suffit donc à caractériser entièrement la loi jointe de l'ensemble des variables aléatoires du modèle.

---

1. L'ordre topologique implique que  $pa(X_1)$  est vide donc  $P(X_1 | pa(X_1)) = P(X_1)$

Le **domaine** d'un nœud est l'ensemble des valeurs pouvant être prises par la variable aléatoire associée à ce nœud. Concrètement la LPC d'un nœud est une matrice multidimensionnelle aussi appelée "potentiel". Dans la suite, nous noterons  $dom(X)$  le domaine du nœud  $X$  et  $pot(X)$  le potentiel associé représentant la LPC de ce nœud. La LPC d'un nœud  $X$  qui possède  $d$  nœuds parents est un potentiel à  $d + 1$  dimensions. La complexité spatiale d'un RB est donc directement liée à la taille des domaines de ses nœuds ainsi que ceux de leur parents. Soit la notation  $|\cdot|$  pour désigner la taille d'un domaine ou d'un potentiel, la complexité spatiale (CS) d'un nœud  $X$  est égale à :

$$|pot(X)| = |dom(X)| \prod_{n_i \in pa(X)} |dom(n_i)|$$

Ainsi, dans l'exemple sur la figure 2.1 en supposant que les 4 nœuds aient un domaine de même taille égal à  $S$ , la taille du potentiel associé à  $Y$  sera  $S^3$ . La complexité spatiale d'un réseau bayésien à  $N$  nœuds ( $X = (X_1, \dots, X_n)$ ) est la somme de la complexité spatiale des LPC de chaque nœud. Supposons que tous les nœuds aient des domaines de même taille  $M$  et  $N_{par}$  le nombre maximal de parents d'un nœud du graphe, la complexité générale du RB donc le graphe est noté  $\mathcal{G}$  est :

$$|\mathcal{G}| = \sum_{X_i \in X} |pot(X_i)| = nM^{N_{par}+1}$$

Si l'on voulait stocker de façon classique la loi jointe de  $n$  variables aléatoires, il serait alors nécessaire de définir un potentiel à  $n$  dimensions d'une complexité spatiale égale à

$$|(X_1, \dots, X_n)| = M^n$$

Il est donc extrêmement avantageux en terme de complexité spatiale de représenter une loi jointe d'un ensemble de  $n$  variables aléatoires par un réseau bayésien, à condition que  $N_{par}$  soit plus petit que  $n$ . C'est à dire que l'ensemble des nœuds du graphe n'ait pas un nombre de parents trop élevé. C'est donc l'exploitation des relations d'indépendances conditionnelles existantes du fait de l'absence de flèches entre certains couples de nœuds qui permet de réduire considérablement la complexité spatiale d'un tel modèle et qui justifie son intérêt de ce point de vue là.

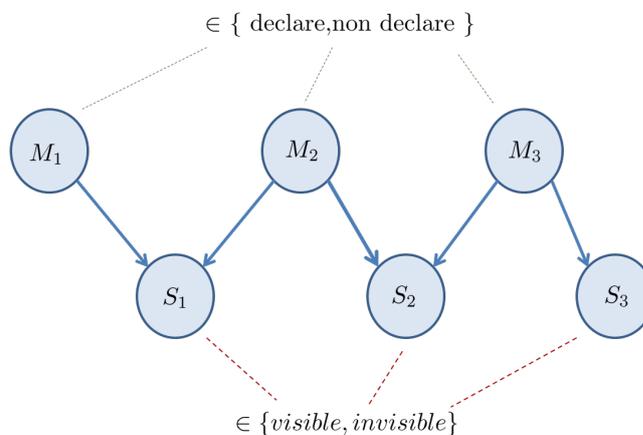
Les RB sont utilisés dans différents domaines pour modéliser diverses situations caractérisées par un ensemble de variables aléatoires dont les états sont interdépendants.

*Exemple 2 : (Problème simple de diagnostic en médecine).*

On considère 3 maladies représentées chacune par une variable aléatoire ( $M_1, M_2, M_3$ ). Chacune de ces variables possède 2 états. Soit la maladie est déclarée, soit elle ne l'est pas. Soit plusieurs symptômes possibles représentés par les nœuds ( $S_1, S_2, S_3$ ). Ces variables possèdent aussi 2 états {symptôme visible, symptôme non visible}. Une maladie peut provoquer plusieurs symptômes et un même symptôme peut être partagé par plusieurs maladies. La figure 2.5 montre un exemple de RB ainsi créé. Si l'on connaît l'ensemble des probabilités conditionnelles nécessaires pour alimenter le modèle c'est à dire les probabilités a priori  $P(M_1), P(M_2), P(M_3)$  ainsi que les probabilités conditionnelles ( $P(S_1|M_1, M_2), P(S_2|M_2, M_3), \dots$ ), alors il est facile de poser un problème de diagnostic qui consiste en un calcul d'une probabilité de type  $P(M|S)$ . C'est à dire, sachant que le patient observé a tel symptôme, quelle est la probabilité qu'il soit atteint de la maladie  $M$ . La formule de Bayes permet dans ce cas simple de calculer ce type de probabilité. Par exemple

$$p(M_1|S_1) = \frac{\sum_{M_2} P(S_1|M_1, M_2)P(M_1, M_2)}{\sum_{M_1, M_2} P(S_1|M_1, M_2)P(M_1, M_2)}$$

Avec  $P(M_1, M_2) = P(M_1)P(M_2)$  puisque qu'un tel modèle suppose que  $M_1$  et  $M_2$  sont a priori indépendants.<sup>2</sup>



**FIGURE 2.5:** Exemple de cas simple en médecine pouvant être modélisé par un réseau bayésien.

2. Car on est dans le cas d'une connexion convergente entre  $M_1, M_2$  et  $S_1$ .

### 2.1.3 Apprentissage des paramètres

Rappelons que les paramètres à apprendre pour un réseau bayésien sont l'ensemble des LPC de chacun de ses nœuds. Pour cela on utilise des avis d'expert qui ont une certaine connaissance du système étudié, des bases de retour d'expérience du système ou encore l'historique des différentes interventions de maintenance qui ont été faites.

Utiliser des avis d'expert consiste à interroger une personne ayant une bonne connaissance du fonctionnement et de la dynamique du système pour lui demander d'évaluer les probabilités conditionnelles que l'on recherche. Cette démarche est possible mais compliquée. Tout d'abord, les valeurs fournies peuvent être influencées par divers facteurs subjectifs non rationnels. De plus, si il est concevable d'évaluer une probabilité simple, il peut être très difficile d'évaluer une probabilité conditionnelle d'autant plus si celle-ci a un grand nombre de parents.

La méthode la plus utilisée en pratique est un apprentissage statistique des paramètres à partir des base de données dont on dispose. Il existe principalement 2 méthodes couramment utilisées et qui le seront dans la partie application de cette thèse (chapitre 4). Dans le cas de bases de données complètes ne contenant aucune données manquantes, la méthode la plus simple est celle **du maximum de vraisemblance** [22]. En présence de données incomplètes, **l'algorithme Expectation Maximization** permet une meilleure estimation des paramètres [17].

#### Méthode du maximum de vraisemblance (MV)

Seul le cas de données discrètes à temps discret est considéré ici.

Soit un réseau bayésien à  $n$  nœuds représentant l'ensemble des variables  $(X_1, \dots, X_n)$ . Et soit une base de  $L$  observations complète  $(y_1^{(l)}, y_2^{(l)}, \dots, y_n^{(l)})_{l=1, \dots, L}$ . Soit  $\theta_{i,j,k}$  le paramètre du RB représentant la probabilité conditionnelle  $P(X_i = j | pa(X_i) = k)$ <sup>3</sup>. En notant  $N_{i,j,k}$  le nombre d'observation dans la base contenant  $(\dots, y_{pa(i)} = k, \dots, y_i = j)$ , on a alors :

$$\theta_{i,j} = \frac{N_{i,j,k}}{\sum_j N_{i,j,k}}$$

Il s'agit simplement de la fréquence empirique d'observations contenant exactement la configuration  $y_i = j$  parmi l'ensemble des observations contenant la configuration des

---

3. Notons que ici  $k$ , est une configuration possible de l'ensemble des variables parentes de  $X_i$ , donc  $k$  est un  $h$ -uplet, ou  $h$  est le nombre de parent de  $X_i$

parents  $pa(X_i) = k$ . La démonstration de ce résultat intuitif se fait en écrivant la log vraisemblance des données.  $\theta_{i,j,k}$  correspond alors au point où celle-ci la vraisemblance atteint son maximum. Le détail de cette démonstration peut être trouvé dans [45] ou dans [19].

*En reprenant l'exemple 2 et en appliquant la méthode (MV), la probabilité  $P(M_1 = \text{declare})$  correspond donc à la proportion des individus d'une population qui ont été atteints de la maladie  $M_1$  et la probabilité conditionnelle  $P(S_1 = \text{visible} | M_1 = \text{declare}, M_2 = \text{non declare})$  correspond à la proportion de patients, parmi ceux qui ont été atteints de la maladie 1 mais pas de la maladie 2 et qui possédaient le symptôme 1.*

### **Algorithme Expectation Maximization (EM)**

L'algorithme EM est utilisé dans deux situations pour lesquelles on dispose de données manquantes, ce qui est souvent le cas dans les bases de données de REX.

- Le cas MAR (*Missing at Random*), c'est à dire les situations pour lesquelles la probabilité qu'une donnée soit manquante dépend des données connues.
- Le cas MCAR (*Missing Completely At Random*) dans lequel la probabilité qu'une donnée soit manquante est complètement indépendante aussi bien des données connues que celles qui sont inconnues.

L'algorithme EM permet alors d'effectuer la procédure d'apprentissage en tenant compte des données manquantes.

Notons que dans le cas MCAR, il est possible de prendre en compte uniquement les observations complètes de la base en considérant que ce sous-ensemble est représentatif. Il est par contre nécessaire d'avoir suffisamment d'entrées complètes disponibles pour pouvoir appliquer la méthode du maximum de vraisemblance.

L'algorithme EM est un algorithme visant à approcher itérativement le paramètre  $\theta$  que l'on cherche à déterminer. Soit  $D$  la base de données d'observation. Soit  $D_o$ , le sous-ensemble des données observées, et  $D_m$ , le sous ensemble des données manquantes. On a donc  $D = (D_o, D_m)$ . Supposons un paramètre  $\theta^n$  estimé à l'étape  $n$ , chaque itération comporte 2 étapes.

1) La première étape (Expectation) consiste à écrire la vraisemblance des données manquantes  $P(D_m | \theta^n)$ . Il est ainsi possible calculer l'espérance de la log vraisemblance d'un jeu de paramètres quelconque calculé en utilisant une distribution de données manquantes selon  $\theta^n$ .

$$Q(\theta|\theta^n) = E_{D_m|D_o,\theta^n} [\text{Log}P(D_o, D_m|\theta)]$$

2) La seconde étape (Maximization) consiste à trouver une valeur  $\theta$  qui maximise cette vraisemblance

$$\theta^{(n+1)} = \underset{\theta}{\text{argmax}} E_{D_m|D_o,\theta^n} [\text{Log}P(D_o, D_m|\theta)]$$

Il a été prouvé dans [17] que cet algorithme converge vers un maximum local de la fonction de vraisemblance ou dans certains cas vers un point selle. Pour espérer atteindre le maximum global, on relance en pratique l'algorithme plusieurs fois de suite, avec une initialisation aléatoire. Le paramètre optimal choisi est celui qui maximise la vraisemblance. Cet algorithme peut aussi être utilisé dans le cas d'un apprentissage de paramètres pour des modèles de mélange. L'étiquette de la classe à laquelle appartient chaque donnée jouant alors le rôle de donnée manquante. Il sera utilisé de cette façon dans la partie application présentée dans le chapitre 4. Notons enfin que les deux méthodes d'apprentissage présentées précédemment s'appliquent de la même façon dans le cas d'un réseau bayésien dynamique.

Un réseau bayésien constitué n'a d'intérêt en pratique que si l'on est capable d'effectuer des calculs d'inférence. La section suivante présente les principales méthodes existantes.

### **2.1.4 Inférence dans les réseaux bayésiens**

Un calcul d'inférence consiste à utiliser la connaissance du modèle (les probabilités a priori et l'ensemble des LPC apprises) pour en déduire n'importe quelle distribution de probabilité conditionnelle sur un ensemble de nœuds du graphe.

Soit  $X = (X_1, \dots, X_n)$ , l'ensemble des nœuds d'un RB. Une requête d'inférence consiste à se donner un ensemble d'observations  $X_o \subset X$  puis sachant ces observations, à chercher la probabilité d'un ensemble de variables  $X_r \subset X$  disjoint de  $X_o$ . Il s'agit donc de calculer la probabilité conditionnelle.

$$P(X_r|X_o)$$

Il est possible de distinguer deux grandes catégories de méthodes ayant pour but d'effectuer des calculs d'inférence.

- Les méthodes d'inférence approchée
- Les méthodes d'inférence exacte

### Méthodes exactes

Les méthodes d'inférence exacte consistent à effectuer des calculs algébriques sur l'ensemble des LPC des nœuds de façon à obtenir la probabilité conditionnelle recherchée. L'algorithme le plus connu est celui de **l'arbre de jonction** développé dans [51][31]. Pour nos travaux, nous nous concentrerons sur la méthode dite d'élimination des variables introduite par [68] qui est de complexité équivalente à l'arbre de jonction.

Rappelons que la loi jointe d'un ensemble de variables aléatoires  $X = (X_1, \dots, X_n)$  se factorise de la façon suivante :

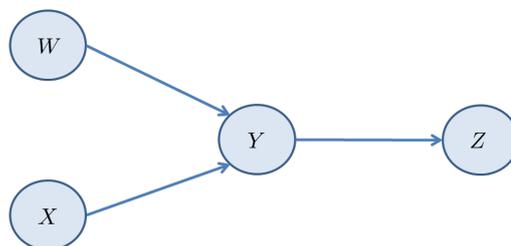
$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | pa(X_i))$$

Dés lors, si l'on souhaite par exemple connaître la loi d'un ensemble  $X_r \subset X$ , il faut sommer la loi jointe de  $X$  sur l'ensemble des variables n'appartenant pas à  $X_r$ , c'est à dire :

$$P(X_r) = \sum_{X_i \in A} P(X) = \sum_{X_i \in A} \prod_{i=1}^n P(X_i | pa(X_i))$$

où  $A = X \setminus X_r$ , l'ensemble complémentaire de  $X_r$  dans  $X$ .

La question est de savoir dans quel ordre il faut effectuer les sommations. Cet ordre d'élimination des variables va déterminer fortement la complexité des calculs qui vont s'effectuer. Ce problème est le point fondamental d'un calcul d'inférence. Pour mieux le comprendre, nous allons l'illustrer en reprenant l'exemple du RB de la figure 2.6 représenté ci-dessous.



**FIGURE 2.6:** Réseau bayésien élémentaire convergent-série de la figure 2.1

D'après la formule de factorisation de la loi jointe d'un RB :

$$P(W, X, Y, Z) = P(W)P(X)P(Y|W, X)P(Z|Y)$$

Supposons que l'on souhaite calculer la probabilité marginale  $P(Z)$ . On a donc

$$P(Z) = \sum_{W, X, Y} P(W)P(X)P(Y|W, X)P(Z|Y)$$

On voit que, en fonction de l'ordre dans lequel on écrit les facteurs du produit, certaines sommations peuvent se déplacer à droite dans l'expression précédente. Il est par exemple possible d'écrire :

$$P(Z) = \sum_X P(X) \sum_W P(W) \sum_Y P(Z|Y)P(Y|W, X) \quad (1)$$

ce qui revient à éliminer les variables dans l'ordre  $(Y, W, X)$  ou bien encore

$$P(Z) = \sum_W P(W) \sum_Y P(Z|Y) \sum_X P(X)P(Y|W, X) \quad (2)$$

ce qui correspond cette fois à l'ordre d'élimination  $(X, Y, W)$ .

Analysons les complexités de ces différents calculs. Reprenons l'expression **(1)**. Supposons pour simplifier que tous les nœuds ont un domaine de taille égale à  $D$ .  $P(Z|Y)$  a donc un potentiel de taille égale à  $D^2$  et  $P(Y|W, X)$  de taille  $D^3$ .

Le produit  $P(Z|Y)P(Y|W, X)$  est donc de taille  $D^4$ . Après la sommation sur  $Y$ , ce produit devient un potentiel à 3 dimensions, donc de taille  $D^3$ . En poursuivant cette analyse sur tout le calcul, en notant  $\sum$  l'opération de sommation et par  $\times$  les opérations de produit entre potentiels, on obtient la chaîne suivante :

$$D^4 \xrightarrow{\sum_Y} D^3 \xrightarrow{\times} D^3 \xrightarrow{\sum_W} D^2 \xrightarrow{\times} D^2 \xrightarrow{\sum_X} D$$

Ce calcul d'inférence conduit donc à manipuler initialement un potentiel de taille  $D^4$ , la taille des objets diminuant au cours du calcul pour arriver au résultat demandé  $P(Z)$  de taille  $D$ . En prenant en compte, l'ordre suivant  $(Y, X, W)$ , on obtient :

$$D^3 \xrightarrow{\sum_X} D^2 \xrightarrow{\times} D^3 \xrightarrow{\sum_Y} D^2 \xrightarrow{\times} D^2 \xrightarrow{\sum_W} D$$

Cette fois, la taille des potentiels est constamment plus petite que celles utilisées dans le calcul précédent. Le calcul s'effectue donc de façon plus rapide et prend moins d'espace

mémoire. Enfin, les ordres  $(W, X, Y)$  et  $(X, W, Y)$  qui sont identiques en terme de complexité (car la structure de  $W$  et  $X$  est symétrique par rapport à  $Y$ ) donnent la chaîne :

$$D^3 \xrightarrow{\Sigma^W} D^2 \xrightarrow{\times} D^2 \xrightarrow{\Sigma^X} D \xrightarrow{\times} D^2 \xrightarrow{\Sigma^Y} D$$

La complexité des calculs est donc encore inférieure en choisissant cet ordre d'élimination. L'ordre d'élimination choisi va donc fortement déterminer la complexité des calculs d'inférence qui vont suivre. La recherche d'un ordre d'élimination optimal dans un graphe quelconque est un problème NP-complet [5]. On essaiera donc dans notre cas, de choisir un ordre d'élimination spécifique au RB que l'on étudiera. L'algorithme d'élimination des variables dans sa version formelle est une généralisation du processus que l'on vient de décrire sur des graphes quelconques. Il nécessite simplement de définir un ordre d'élimination des variables.

Il existe par ailleurs des méthodes d'inférence dites symboliques qui peuvent s'avérer intéressantes dans le cas où l'on doit effectuer plusieurs fois le même type de requête mais en modifiant uniquement certains paramètres. L'idée consiste alors à exprimer l'algorithme d'élimination des variables de façon symbolique en considérant l'ensemble des paramètres pouvant varier comme des paramètres d'une équation symbolique. La requête d'inférence est donc, après une procédure de réduction symbolique, une formule paramétrée. La complexité de cette approche dépend, de façon similaire, de la complexité topologique du réseau bayésien ainsi que de la taille des nœuds. L'efficacité de cette approche a été testée sur un RBD appliqué aux problématiques de maintenance dans [56].

### **Les méthodes d'inférence approchées**

Il y a principalement trois approches :

- Celle qui consiste à effectuer de l'inférence exacte sur un réseau ayant une structure "approchée" de l'original. En pratique, il s'agit de modifier le graphe du RB initial, en supprimant certains arcs dont les dépendances qu'ils représentent sont très faibles. Ces quasi-indépendances amènent à manipuler des LPC de taille réduite ; ce qui permet alors de simplifier considérablement la complexité des calculs d'inférence. La forte réduction de la complexité, par rapport à la faible marge d'erreur produite par ce type de simplification, justifie l'intérêt d'une telle méthode [28].
- Une variante consiste à effectuer des propagations d'informations en utilisant seulement des sous parties des LPC de certains nœuds [4].
- Les méthodes dites stochastiques qui sont basées sur des simulations de Monte-Carlo [25]. C'est à dire que l'on simule un certain nombre de trajectoires sur le réseau. La

fréquence des trajectoires correspondant à l'événement de la requête converge vers la probabilité recherchée. Cette convergence est garantie par la loi des grands nombres. Cependant, il ne s'agit que de probabilités approchées. Il est par ailleurs nécessaire d'effectuer un grand nombre de simulations pour pouvoir prétendre à une bonne précision.

Ayant préalablement présenté le concept de réseau bayésien, les réseaux bayésiens dynamiques sont maintenant abordés.

### 2.1.5 Présentation des Réseaux Bayésien Dynamiques

La thèse de Murphy [44], donne un large éventail des différentes possibilités qu'offrent les RBD pour représenter différents types de problèmes ainsi qu'une très bonne rétrospective sur les algorithmes d'inférence existants. Nous nous contenterons dans cette section d'énoncer les aspects les plus importants et les plus utiles pour la compréhension de la suite de ce manuscrit. Dans la suite de ce manuscrit, le temps est supposé discrétisé,  $t \in \mathbb{N}$ .

Un réseau bayésien dynamique (RBD) est un RB dans lequel l'ensemble des variables considérées peuvent évoluer dans le temps. Il cherche donc à représenter la loi jointe d'un ensemble de processus stochastiques<sup>4</sup> que l'on notera maintenant  $X^{(t)} = (X_1^{(t)}, \dots, X_n^{(t)})$ . En théorie, la distribution de  $X$  à un instant  $t$  notée  $X^{(t)}$  pour  $t > 0$  peut dépendre a priori de l'état de  $X$  à chaque instant entre 0 et  $t$ . En pratique les situations modélisées sont supposées simples et seront représentées par un processus Markovien, c'est à dire que la loi de  $X$  à l'instant  $t$  dépend uniquement des  $r$  tranches de temps précédentes. Cela correspond à l'hypothèse d'indépendance conditionnelle suivante :

$$X^{(t+1)} \perp\!\!\!\perp (X^{(1)}, \dots, X^{(t)}) | (X^{(t+1-r)}, \dots, X^{(t)})$$

qui conduit à :

$$P(X^{(t+1)} | X^{(1)}, \dots, X^{(t-r)}) = P(X^{(t+1)} | X^{(t+1-r)}, \dots, X^{(t)})$$

Comme pour un processus markovien, l'ordre d'un RBD correspond au nombre  $r$  de tranches précédentes qui déterminent l'état de  $X$  à l'instant  $t$ . Si  $r$  est élevé, le réseau contient un grand nombre de dépendances ce qui, comme on l'a vu dans le cas des RB, augmente considérablement la complexité spatiale du modèle, ainsi que la complexité algorithmique des algorithmes d'inférence. Par ailleurs, plus un modèle est de grande

---

4. C'est à dire de variables aléatoires évoluant au cours du temps, donc indexées par  $t$

dimension, plus il est difficile d'apprendre ses paramètres. C'est pourquoi, la majorité des modèles utilisant les RBD, supposent que le réseau est d'ordre 1 c'est à dire :

$$P(X^{(t+1)}|X^{(1)}, \dots, X^{(t)}) = P(X^{(t+1)}|X^{(t)})$$

L'ensemble des variables à l'instant  $t$  dépend donc uniquement de l'état des variables à l'instant  $t - 1$ . Le processus est "sans mémoire".

**Un RBD d'ordre 1 homogène est donc défini par :**

**- Un modèle statique  $BN_0$  donnant la distribution des variables à l'instant initial  $t = 0$  :**

$$P(X^{(0)}) = P(X_1^{(0)}, X_2^{(0)}, \dots, X_n^{(0)}) = \prod_{k=1}^n P(X_k^{(0)}|pa(X_k^{(0)}))$$

**- Un modèle de transition noté  $BN_{\rightarrow}$  décrivant la façon dont la distribution de l'état de  $X$  à l'instant  $t$  est liée à celle de l'instant précédent  $t - 1$ .**

$$P(X^{(t+1)}|X^{(t)}) = P(X_1^{(t+1)}, X_2^{(t+1)}, \dots, X_n^{(t+1)}|X_1^{(t)}, X_2^{(t)}, \dots, X_n^{(t)}) = \prod_{k=1}^n P(X_k^{(t+1)}|pa(X_k^{(t+1)}))$$

Cette formule est de même nature que dans le cas statique. En effet, on peut voir un RBD comme un seul RB statique de taille infinie. Par conséquent, le raisonnement exploitant les relations d'indépendances conditionnelles conduit au même type de formule ; c'est à dire à un produit de lois locales. Enfin, on considère bien souvent en pratique que le processus est homogène c'est à dire que sa loi de transition est invariante. Un RBD d'ordre 1 homogène est donc entièrement caractérisé par la définition précédente.

La figure 2.7 montre un exemple de RBD d'ordre 1. Les flèches en pointillés bleus sont celles qui représentent des dépendances temporelles. Dans cet exemple :

$$X^{(0)} = P(X_1^{(0)})P(X_2^{(0)})P(X_3^{(0)}|X_1^{(0)}, X_2^{(0)})P(X_4^{(0)}|X_3^{(0)})$$

et

$$P(X^{(t+1)}|X^{(t)}) = P(X_1^{(t+1)})P(X_2^{(t+1)}|X_4^{(t)})P(X_3^{(t+1)}|X_3^{(t)}, X_1^{(t+1)}, X_2^{(t+1)})P(X_4^{(t+1)}|X_3^{(t)}, X_3^{(t+1)})$$

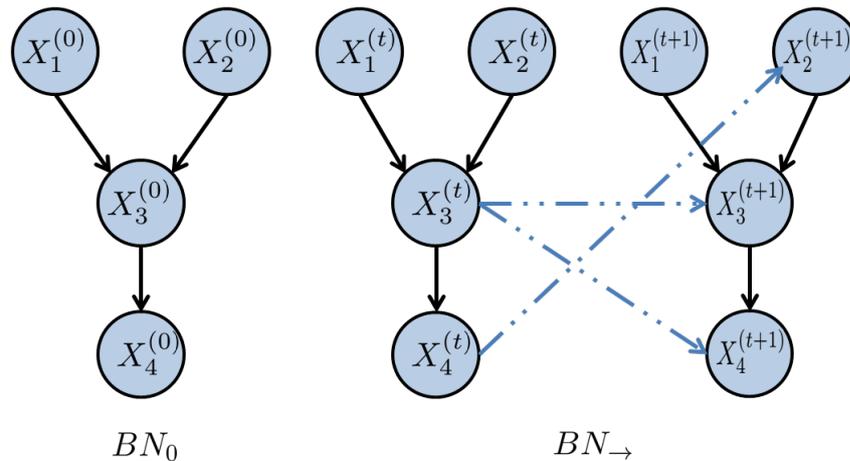


FIGURE 2.7: Exemple de représentation d'un RBD d'ordre 1.

Pour définir le RBD précédent, il est donc nécessaire d'apprendre statistiquement les paramètres de chaque LPC présente dans les formules précédentes. La phase d'apprentissage s'effectue de la même façon que dans le cas statique.

Le problème de l'inférence dans les réseaux bayésiens dynamiques est maintenant abordé. Un algorithme spécifique particulièrement adapté aux RBD appelé **algorithme d'interface** [44] sera présenté. Les méthodes d'inférence spécifiques qui seront développées au cours du chapitre 3 sont fortement basées sur les principes de cet algorithme.

### 2.1.6 Inférence dans un RBD

Deux types de requêtes d'inférence sur les RBD peuvent être distingués :

- Le lissage : consistant à calculer une probabilité conditionnelle entre deux ensembles de la forme  $P(X^{(t-h)}|Y^{(t)})$  dans lequel  $Y$  représente un nœud d'observation.
- Les problèmes de prédiction menant à un calcul de la forme  $P(X^{(t+h)}|Y^{(t)})$  avec  $h > 0$

Pour illustrer les principes sur lesquels reposent l'algorithme d'interface, l'exemple de la figure 2.7 sera repris où  $X = (X_1, X_2, X_3, X_4)$ .

Dans le cas général :

$$P(X^{(t+1)}) = P(X^{(t)})P(X^{(t+1)}|X^{(t)})$$

Pour mettre à jour la loi jointe de  $X^{(t+1)}$ , il est donc nécessaire de connaître la loi jointe de  $X^{(t)}$ . En regardant attentivement le graphe du modèle de transition  $BN_{\rightarrow}$ , il est clair que  $(X_3^{(t)}, X_4^{(t)})$  d-sépare  $(X_1^{(t+1)}, X_2^{(t+1)}, X_3^{(t+1)}, X_4^{(t+1)})$  de  $(X_1^{(t)}, X_2^{(t)})$ . La relation d'indépendance conditionnelle qui s'en suit permet d'écrire que :

$$P(X^{(t+1)}|X^{(t)}) = P(X^{(t+1)}|X_3^{(t)}, X_4^{(t)})$$

Pour mettre à jour la loi jointe de la tranche  $t$  vers la tranche  $t + 1$ , il est donc juste nécessaire de connaître la loi jointe de  $(X_3, X_4)$  car ceux sont les seuls nœuds qui ont des connexions temporelles avec la tranche suivante. Ceci amène à la définition suivante.

**Définition :** Dans un réseau bayésien dynamique d'ordre 1, on définit l'interface gauche de la tranche  $t + 1$ , comme l'ensemble des nœuds de la tranche  $t$  qui ont un lien de dépendance avec au moins un nœud de la tranche  $t+1$ .

Dans un réseau homogène, la loi de transition étant invariante dans le temps, cette interface est la même à chaque instant, on pourra donc la noter sans ambiguïté  $I_g$  tout au long de ce manuscrit. De façon analogue à ce que l'on a vu dans l'exemple précédent, l'interface gauche dans n'importe quel réseau bayésien dynamique d-sépare la tranche  $t + 1$  de la tranche  $t$ . La relation générale suivante s'en suit :

$$P(X^{(t+1)}|X^{(t)}) = P(X^{(t+1)}|I_g^{(t)})$$

Par conséquent, pour mettre à jour la loi jointe de  $X$  de l'instant  $t$  à  $t + 1$ , il suffit de connaître la loi jointe de l'interface gauche. Cette approche prend le nom d'algorithme d'interface :

$$P(X^{(t+1)}) = \sum_{I_g^{(t)}}^{(t)} P(I_g^{(t)})P(X^{(t+1)}|I_g^{(t)})$$

La complexité de l'algorithme d'inférence dépendra donc étroitement de la taille de cette interface.

Les principaux outils mathématiques utilisés étant posés, la construction du modèle de maintenance servant de point de départ à cette thèse est décrite étape par étape.

## 2.2 Construction du modèle de dégradation puis de maintenance

### 2.2.1 Définition et point de vue général

Rappelons que nous considérons ici des systèmes à espace d'états discrets et à temps discrétisé. Un composant à  $N$  états discrets est considéré. Son état est décrit par un processus stochastique noté  $X^{(t)} \in \{1, 2, 3, \dots, x_P\}$  et  $t \in \mathbb{N}$ . 1 sera l'état considéré comme l'état neuf et  $x_P$  désignera un état de panne ou du moins un état nécessitant obligatoirement une action de maintenance. On suppose que la dégradation est un processus monotone c'est à dire qu'aucune auto-réparation n'est possible.

Couramment, les modèles de dégradation modélisent l'évolution de la dégradation par une fonction dépendant directement du temps. Cette fonction est bien souvent définie analytiquement par morceau, en raison du cadre à temps discret. De manière duale, le modèle de dégradation peut chercher à approcher non pas les instants de changement d'état mais les durées de temps de séjour dans chaque état [18]. C'est l'approche qui sera retenue ici. Le modèle de dégradation noté  $\mathcal{M}$  sera donc un ensemble de distribution de temps de séjour dans chaque état.  $\mathcal{M} = \{\Psi_1, \dots, \Psi_P\}$  où  $\Psi_x$  représente la distribution de temps de séjour dans l'état  $x$ .  $\forall x, \Psi_x : \Omega \rightarrow [0, 1]$  est une fonction de probabilité discrète définie sur  $\Omega = \{1, \dots, T_{max}\}$  où  $T_{max}$  est le temps de séjour maximal possible sur l'ensemble de tous les états. Les distributions auront toutes le même univers borné par  $T_{max}$  pour des questions pratiques liées à la représentation de cette approche par RBD (voir dans la suite la section 1.3.3 sur les MGD). Cela n'empêche pas certaines distributions d'avoir un support<sup>5</sup> différent de  $\Omega$ . Le temps de séjour dans l'état  $x$  sera noté  $S_x$  et suit donc la loi  $\Psi_x$ . La figure 2.8 illustre l'ensemble des notations précédentes.

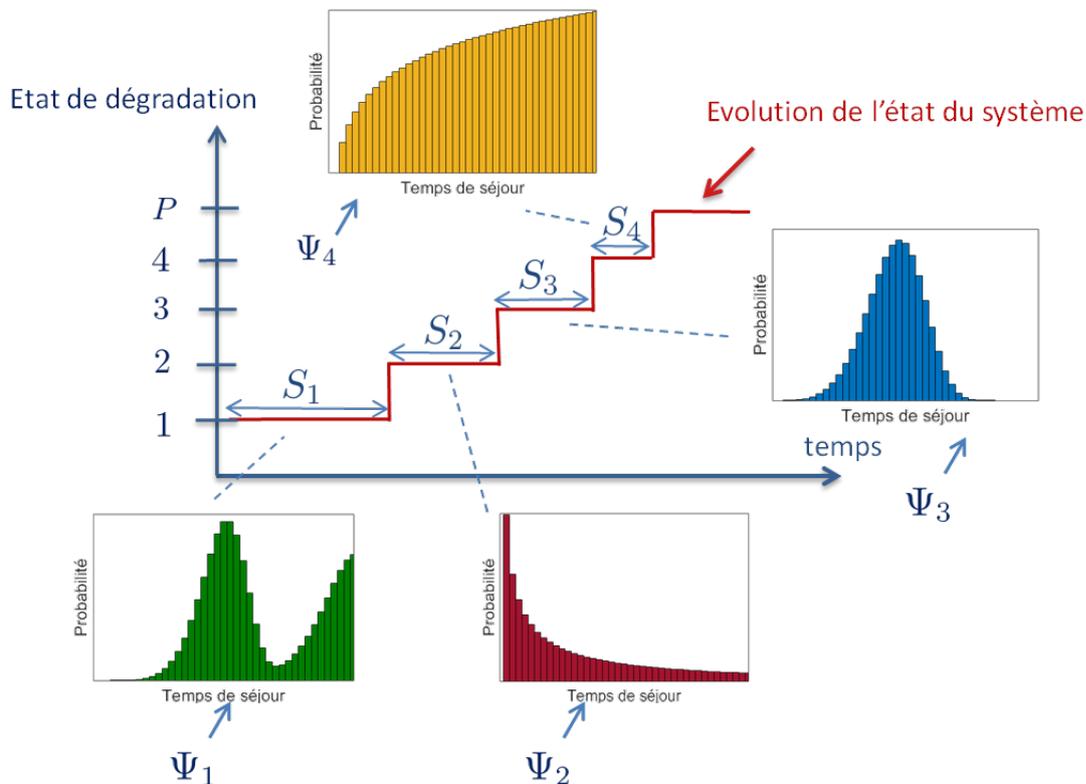
#### Apprentissage des lois de temps de séjour par la méthode MV

Soit un système à  $n$  états et  $Y$  une base de  $L$  séquences d'observations discrètes de l'état du système depuis sa mise en service jusqu'à la panne.  $Y = (y_1^{(l)}, y_2^{(l)}, \dots, y_n^{(l)})_{l=1, \dots, L}$  où  $y_i^{(l)}$  représente le temps de séjour observé à l'état  $i$  sur la  $l$ -ième séquence. En appliquant la méthode MV on obtient :

$$\Psi_i(s) = \frac{\sum_l \mathbb{1}\{y_i^{(l)} = s\}}{L}$$

---

5. Rappelons que le support désigne l'ensemble des valeurs pour lesquelles la fonction n'est pas nulle



**FIGURE 2.8:** Modèle de dégradation vu comme un ensemble de lois de temps de séjour

soit la fréquence empirique du temps de séjour  $s$  à l'état  $i$  dans la base d'observation. Cela nécessite une base dont la taille  $L$  soit assez grande. Cette formule sera fréquemment utilisée lors de l'apprentissage pour les applications tests qui seront décrites dans le chapitre 4. Dans le cas de bases de données contenant des trajectoires incomplètes, l'algorithme EM sera utilisé.

La question principale est de trouver un formalisme mathématique permettant de représenter cette approche basée sur les temps de séjour. Nous présentons maintenant quelques approches pour modéliser les distributions de temps de séjour et donc la dynamique de la dégradation, puis au final le modèle de maintenance.

### 2.2.2 Chaîne de Markov

La façon la plus simple de construire un modèle de dégradation à partir d'un modèle graphique est de représenter la dégradation d'un composant par une simple chaîne de Markov. C'est à dire un RBD d'ordre 1 à un seul nœud. (voir figure 2.9).

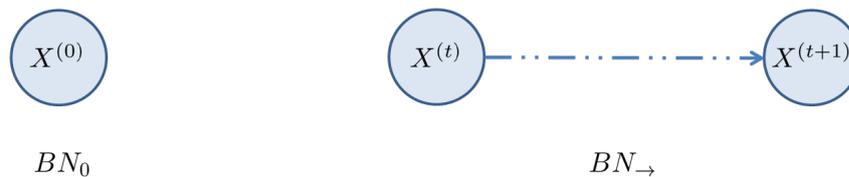


FIGURE 2.9: Une chaîne de Markov élémentaire.

Tout comme un RBD, une chaîne de Markov est définie par :

- Une distribution initiale donnant la loi de  $X^{(0)}$
- Un modèle de transition

Comme il n'y a qu'une seule variable, on utilise souvent une autre représentation plus parlante qui consiste à représenter par un nœud chaque état de la variable. Les flèches quantifient alors les probabilités de transition d'un état à l'autre. Une absence de flèche correspondant à une probabilité de transition nulle.

*Exemple :* La chaîne de Markov illustrée sur la figure 2.10 représente le modèle de transition d'une variable  $X^{(t)}$  qui peut prendre 4 états  $X \in \{1, 2, 3, 4\}$ . Si par exemple,  $X^{(t)}$  est à l'état 3 à un instant donné, elle ne peut transiter que vers l'état 2 avec une probabilité de 0.7 ou rester sur l'état 3 avec une probabilité de 0.3.

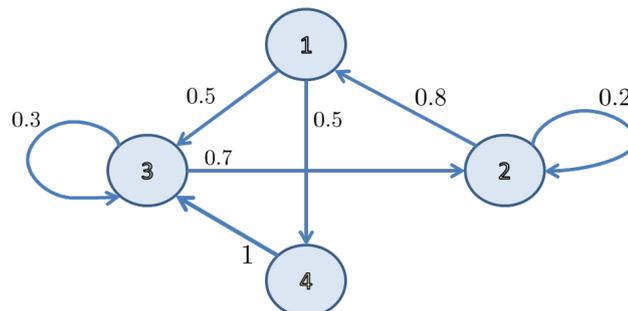


FIGURE 2.10: Exemple de représentation plus intuitive d'une chaîne de Markov

Ce type de modèle induit une hypothèse sur la distribution des temps de séjour. En effet, supposons que  $X^{(t)}$  soit dans un état  $a$ . A cause de la propriété de Markov, la probabilité de transition d'un état vers un autre état à l'instant suivant est toujours la même. La probabilité de boucler  $k$  fois sur le même état  $a$  est donc égal à  $[P(X^{(t+1)} = a | X^{(t)} = a)]^k$ .

*Dans l'exemple précédent, la probabilité de rester  $k$  instants sur l'état 3 est donc  $(0.3)^k$*

La probabilité de séjourner dans un état donné pendant  $k$  unités de temps est donc de la forme  $p^k$ . Un temps de séjour est donc obligatoirement distribué suivant une loi géométrique dans le cas discret (ou exponentielle dans le cas de variables continues). Cette hypothèse n'est en général pas valide pour modéliser en pratique la dégradation d'un composant. En effet, le taux de dégradation n'est que rarement constant. Il est donc nécessaire d'avoir un modèle de dégradation permettant de rejeter le caractère markovien. Le modèle graphique de durée qui va maintenant être présenté permet de dépasser le cas purement Markovien.

### 2.2.3 Modèle graphique de durée

Le modèle graphique de durée est une extension du modèle de dégradation markovien précédent qui a été développé par [19] en partant des modèles de durée évoqués dans [44]. L'idée de base est simple : puisque le modèle markovien ne permet pas de contrôler la distribution de temps de séjour dans chaque état, ce temps de séjour est explicitement représenté dans le modèle à l'aide d'un nœud de durée. Le modèle graphique de durée (MGD) est donc une chaîne de Markov à laquelle un second nœud est ajouté noté  $S^{(t)}$  associé à  $X^{(t)}$  et décrivant explicitement la durée de séjour restante dans l'état courant  $X$  avant sa prochaine transition vers un autre état. Le nœud  $S^{(t)}$  est à valeur entière.  $S^{(t)} \in \{1, \dots, T_{max}\}$ . Comme la variable temps, le temps de séjour est donc discrétisé. De plus, le temps de séjour minimal dans un état est supposé être supérieur ou égal à 1 unité de temps.

La figure 2.11 donne une représentation graphique de ce modèle.

#### *Explications et interprétation*

Lorsque  $X^{(t)}$  est dans un état donné,  $S^{(t)}$  contient la valeur du temps de séjour durant lequel  $X$  va rester sur cet état. Cette durée dépend de l'état courant de  $X$ . La loi de  $S^{(t)}$  va donc dépendre de celle de  $X^{(t)}$  (flèche 1). Ensuite, à chaque instant,  $S^{(t)}$  décroît de 1 unité et donc  $S^{(t+1)} = S^{(t)} - 1$  (codé de façon déterministe par la flèche 2).  $X^{(t)}$  est bloqué sur son état tant que  $S^{(t)}$  est différent de 1. Quand  $S^{(t)} = 1$ ,  $X$  est libre de

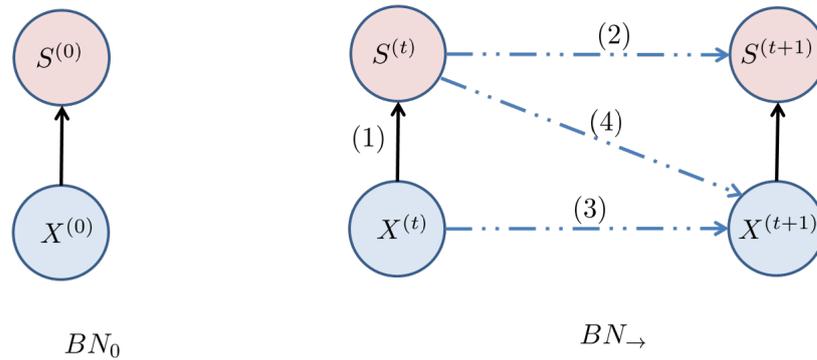


FIGURE 2.11: Modèle graphique de durée

transiter vers un autre état à l'instant suivant selon une certaine matrice de transition (flèches 3). Ce nouvel état est choisi en fonction d'une matrice de transition de  $X$ . Le nouvel état de  $X$  va alors déterminer le choix d'une distribution de temps de séjour pour  $S^{(t)}$  (flèche 1).

### Formalisation

Les lois initiales de  $X^{(0)}$  et  $S^{(0)}$  sont définies comme suit :

$$P(X^{(0)} = x) = V(x)$$

$$P(S^{(0)} = s | X^{(0)} = x) = F_0(x, s)$$

- $V(x)$  est un vecteur de taille  $N$  représentant la distribution initiale de  $X$  (On rappelle que  $N$  désigne le nombre d'état du système modélisé).
- $F_0(x, s)$  est un potentiel à deux dimensions  $N * T_{max}$ . La valeur  $F_0(x, s)$  représente la probabilité que le temps de séjour dans l'état initial soit égal à  $s$  sachant que l'état initial est  $x$ . Ce potentiel contient donc l'ensemble des distributions de temps de séjour dans chaque état. C'est à dire que pour la  $x$ -ème ligne, on a :  $F_0(x, \cdot) = \Psi_x$ . Des domaines de même taille égale à  $T_{max}$  pour chaque  $\Psi_x$  ont donc été choisis de façon à pouvoir renseigner dans un même nœud l'ensemble de ces lois.
- Les lois de transition sont définies par :

$$P(X^{(t+1)} = x | X^{(t)} = x', S^{(t)} = s') = Q^{\rightarrow}(x, s', x') = \begin{cases} Q_{sys}(x', x) & \text{si } s' = 1 \\ I(x', x) & \text{si } s' \geq 2 \end{cases}$$

$$P(S^{(t+1)} = s | X^{(t+1)} = x, S^{(t)} = s') = F^{\rightarrow}(x, s', x') = \begin{cases} F_{sys}(x, s) & \text{si } s' = 1 \\ C(s', s) & \text{si } s' \geq 2 \end{cases}$$

$Q^{\rightarrow}$  est un potentiel à 3 dimensions de taille  $N * T_{max} * N$  défini par morceaux décrivant la loi de transition de  $X$  entre les instants  $t$  et  $t+1$ . Cette loi dépend du temps de séjour restant à l'instant précédent donc de  $s'$ .

Le cas  $s' = 1$ , décrit le cas d'une transition naturelle représentée par  $Q_{sys}$ , qui est un potentiel de taille  $N * N$  donnant la loi de transition de  $X$  vers un autre état. Si  $s' \neq 1$ , l'état de  $X$  est bloqué, la loi de transition est donc la matrice identité de taille  $N * N$ .

$F^{\rightarrow}$  représente la loi de transition de  $S$ . C'est un potentiel de taille  $N * T_{max} * N$  défini par morceaux. Si  $s' = 1$  alors cela signifie qu'une transition imminente est vers un nouvel état à l'instant suivant. Le nouveau temps de séjour dans le nouvel état doit alors être choisi. Dans ce cas la transition est réglée par  $F_{sys}$  qui est un potentiel  $N * T_{max}$  contenant l'ensemble des lois de temps de séjour. En général, ces lois sont identiques à celles intervenant à  $t = 0$  et donc  $F_{sys} = F_0$ . Si  $S \geq 2$ , l'ensemble du temps de séjour dans l'état courant n'est pas encore écoulé. Celui-ci décroît alors de 1 unité entre  $t$  et  $t + 1$ . Cette décrémentation est définie par le potentiel de taille  $T_{max} * T_{max}$  qui est une loi déterministe définie par  $C(s, s') = 1$  si  $s = s' - 1$  et 0 sinon.

Le modèle de dégradation présenté fonctionne dans un cadre discret et permet d'utiliser n'importe quelle distribution de probabilité discrète comme loi de temps de séjour. Le modèle de maintenance VirMaLaB basé sur ce modèle de dégradation, est présenté dans le paragraphe suivant.

## 2.2.4 Modèle VirMaLab

La plateforme VirMaLab (*Virtual Maintenance Laboratory*) a pour objectif de représenter l'ensemble d'une politique de maintenance, de l'évaluer et d'en optimiser les paramètres de maintenance. Elle est basée sur le formalisme des RBD et son architecture générale est représentée sur la figure 2.12.

Un modèle générique se décompose en 4 modules distincts :

1) Le module de dégradation a pour but de modéliser la dynamique de dégradation cachée du système modélisé selon l'approche **modèle graphique de durée**. Cette dégradation peut être observable ou non. La valeur du nœud représentant l'état du système peut

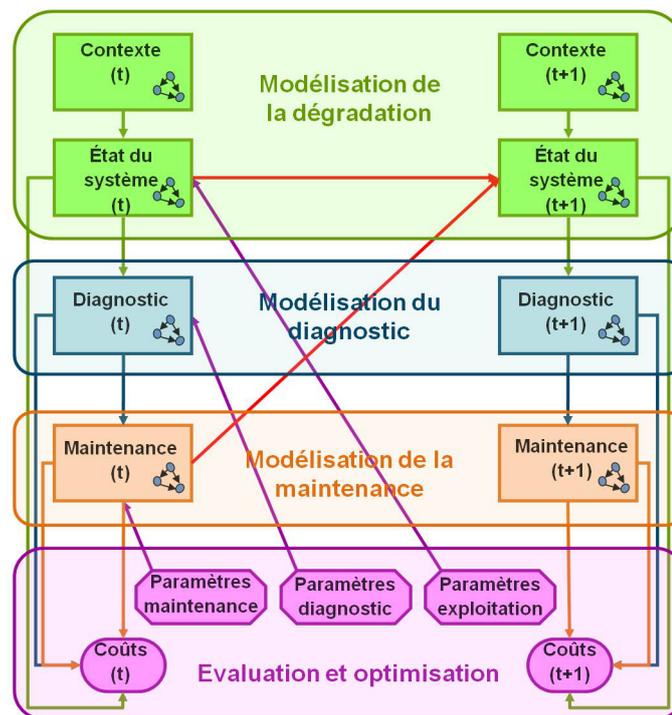


FIGURE 2.12: Modèle générique VirMaLaB [13]

donc ne pas être connue du mainteneur. Le modèle graphique de durée peut prendre en compte une variable de contexte dont la valeur peut modifier les lois de temps de séjour considérées dans chaque état.

2) Le second module décrit la politique de diagnostic qui est appliquée au système. Le nœud de diagnostic représente l'information dont dispose le mainteneur à l'issue d'un processus de surveillance. La loi de transition reliant l'état du système au résultat du diagnostic permet de modéliser les fausses alarmes ainsi que les taux de non détection à l'aide d'une matrice de confusion différente de la matrice identité. Le diagnostic peut être conditionné par une variable d'activation ce qui permettra de modéliser les politiques de maintenance où le diagnostic n'est pas présent en continu mais de façon calendaire.

3) Le troisième module représente les actions de maintenance appliquées au système. Ce module se compose d'un nœud de maintenance qui dépend du diagnostic, dans le cas où l'on souhaite modéliser une politique de maintenance **conditionnelle**, ou d'une variable d'activation si l'on souhaite modéliser une politique de maintenance **systématique**. Une action de maintenance influe sur l'état réel du système à l'instant suivant. Cela est

représenté par la flèche rouge reliant la maintenance à l'état du système à l'instant suivant  $t + 1$ .

4) Le module d'optimisation contient plusieurs outils permettant d'affecter des indicateurs de coût ou de performance aux nœuds contenus dans les modules précédents. Il agit sur l'ensemble des paramètres de maintenance (pas de remplacement, pas de diagnostic,..) et contient diverses fonctions d'utilité associées aux nœuds de maintenance et de diagnostics qui permettent d'associer un coût à chacun de ces processus. Pour plus de détails sur le formalisme décrivant les lois de chacun des nœuds de ce modèle voir [19].

Le choix du formalisme des réseaux bayésiens permet de représenter des variables de natures très différentes et rend possible la représentation dans un seul et même modèle de quantités aussi diverses que l'état d'un système évoluant au cours du temps, d'actions de maintenance, de procédures d'inspection, de fonctions de coûts relatives à toutes ces actions et de pouvoir quantifier l'ensemble des variables manipulées en terme probabiliste.

## **2.2.5 Conclusion**

Après avoir introduit les principaux concepts et définitions liés au formalisme des réseaux bayésiens dynamiques, nous avons présenté le point de vue adopté pour représenter la dégradation d'un système à états discrets à savoir la modélisation des distributions de temps de séjour. En utilisant le formalisme des RBD, un modèle de dégradation a été présenté : le modèle graphique de durée. Enfin, le modèle complet VirMaLaB est un RBD général qui encapsule le processus de dégradation du système, les processus de diagnostics et les actions de maintenance. Il est ainsi possible de modéliser l'intégralité du processus de décision suivi par un mainteneur de façon à pouvoir notamment effectuer des études de fiabilité ou encore d'optimiser les paramètres des stratégies de maintenance. Une fois ce modèle complet posé, les contributions de cette thèse vont pouvoir être présentées. Le chapitre 1 a montré l'intérêt de développer des politiques de maintenance prévisionnelles. Or le concept de pronostic n'apparaît pas dans le modèle VirMaLaB. Il s'agira donc de présenter dans le prochain chapitre l'ensemble des développements qui sont proposés dans le but d'ajouter cette possibilité au modèle.

# Chapitre 3

## Traduction du concept de pronostic par modèle graphique probabiliste

Ce chapitre présente les contributions majeures de cette thèse. Rappelons que l'objectif principal de cette thèse est d'introduire la notion de maintenance basée sur le pronostic dans le modèle VirMaLab. Pour cela, un algorithme ayant pour but de calculer la RUL pour des systèmes à états discrets et périodiquement observables va être défini dans une première section. Puis en cherchant à optimiser la précision des calculs de pronostics effectués, un nouveau modèle de dégradation nommé Modèle Graphique de Durée Conditionnelle (MGDC) sera introduit dans une deuxième section. Enfin pour pouvoir intégrer ces calculs dans le modèle VirMaLab, une représentation graphique de ce processus de calcul sera proposée dans une troisième section. Cela conduira notamment à traiter la question de la complexité des calculs d'inférence en proposant notamment des procédures spécifiques exploitant certaines propriétés du graphe.

### 3.1 Problématique

L'objectif est d'introduire le concept de pronostic dans le modèle VirMaLab, c'est à dire calculer une RUL sur un horizon donné. Il est possible d'effectuer ce type de calcul avec un simple modèle graphique de durée. En effet, soit un instant  $t$  donné, la probabilité pour que la RUL à l'instant  $t$  soit égale à  $h$  est celle de l'événement : "le système tombe en panne à l'instant  $t+h$ ", soit :

$$P(X^{(t+h)} = x_P, X^{(t+h-1)} \neq x_P)$$

Une telle probabilité peut être facilement calculée avec un algorithme d'inférence sur le MGD. De plus, si l'on veut tenir compte d'observations disponibles, il faut ajouter au modèle graphique de durée le module de diagnostic du modèle VirMalab. Concrètement

il s'agit d'ajouter 2 nœuds supplémentaires au MGD :

- Un nœud  $D^{(t)} \in \{1, \dots, x_P\}$  représentant le résultat de diagnostic.

Dans ce cas, la matrice de transition entre  $X^{(t)}$  et  $D^{(t)}$  décrivant la relation existante entre l'état du système et le résultat donné par le processus de diagnostic peut être, soit la matrice identité si l'on considère que le diagnostic est parfait et rend compte toujours de l'état réel du système, soit une matrice non identité permettant de modéliser les cas de non détection et de fausses alarmes.

- Un nœud  $Act^{(t)} \in \{Active, Desactive\}$  qui permet d'activer ou non le diagnostic pour modéliser les situations où le diagnostic n'est disponible qu'à certains instants.

Le RBD résultant est équivalent à une chaîne de Markov cachée qui est représentée sur la figure 3.1. Dès lors, supposons qu'aux instants  $(t_1, \dots, t_n)$  les résultats de diagnostic  $(d_1, \dots, d_n)$

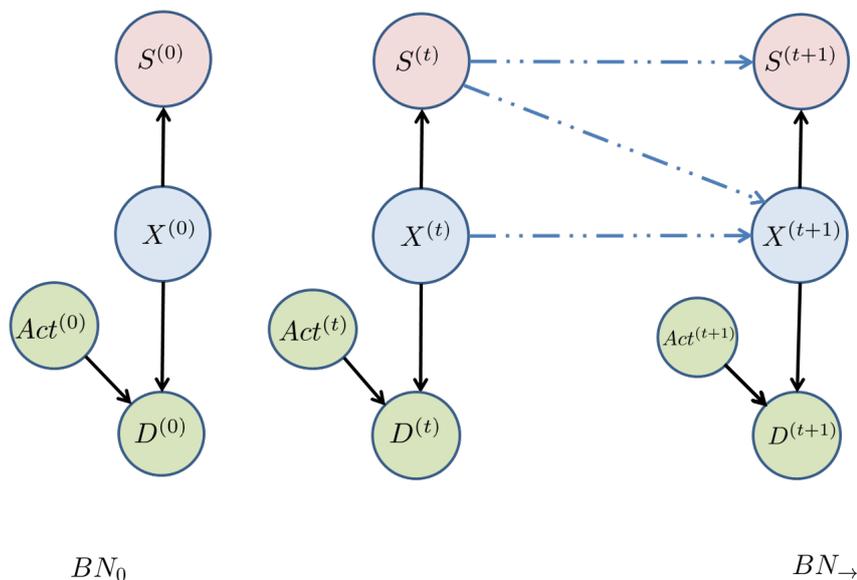


FIGURE 3.1: Modèle graphique de durée incluant le diagnostic

soient disponibles. L'algorithme d'élimination de variables permet facilement d'intégrer les observations disponibles en calculant la probabilité conditionnelle suivante :

$$P(X^{(t+h)} = x_P, X^{(t+h-1)} \neq x_P | d^{(t_1)} = d_1, \dots, d^{(t_n)} = d_n) \quad (3.1)$$

En faisant varier  $h$  dans le calcul précédent, il est possible d'obtenir une distribution de la RUL autour d'un horizon donné.

Ce résultat n'est pas directement utilisable pour représenter la RUL dans un RBD pour deux raisons qui vont être maintenant détaillées.

- Utiliser le résultat de l'expression 3.1 pour définir la loi d'un noeud  $RUL^{(t)}$ , représentant la RUL à l'instant  $t$ , nécessiterait de coder l'ensemble du calcul d'inférence (basé sur l'algorithme d'interface) permettant de parvenir à l'expression 3.1 dans la loi de probabilité conditionnelle (LPC) du noeud  $RUL^{(t)}$ . Or, d'après cette expression, la quantité  $RUL^{(t)}$  dépend de l'historique de tous les diagnostics disponibles jusqu'à l'instant  $(D^{(t_1)}, \dots, D^{(t_n)})$ . Cela supposerait donc de considérer que le noeud  $RUL^{(t)}$  possède autant de noeuds parents qu'il y a de diagnostics déjà effectués sur le système. Ce nombre grandissant à chaque instant  $t$ , cela impliquerait un graphe changeant de façon dynamique. Enfin cela obligerait à d'écrire l'ensemble de la procédure d'inférence de manière symbolique puis après simplification d'obtenir une fonction à  $t$  variables (les  $t$  diagnostics disponibles, jusqu'à l'instant  $t$ ) donnant en sortie une fonction correspondant à la distribution de la RUL à l'instant  $t$ . Tout cela ne semble pas faisable avec le formalisme des RBD.
- Rappelons que l'objectif principal est de pouvoir représenter le processus de décision d'un mainteneur qui adopterait une politique de maintenance de nature prévisionnelle. Représenter une politique de maintenance par un modèle mathématique suppose que la politique de maintenance suive un référentiel bien précis. Le modèle VirMaLab suppose que le noeud d'actions de maintenance dépend d'un noeud de pronostic à valeurs discrètes. Coder une politique de maintenance dans un RBD revient à définir une LPC décrivant la relation de dépendance conditionnelle entre le noeud de RUL et le noeud d'action de maintenance. Or, une LPC est une fonction paramétrique dont les paramètres sont des valeurs numériques discrètes et non des fonctions. Il semble donc difficile de coder une règle de décision qui serait par exemple basée sur un test considérant la distribution de la RUL (par exemple qui prendrait en compte la variance de la RUL). Il est donc nécessaire de travailler avec une instanciation de cette distribution pour qu'elle soit utilisable et codable dans une LPC.

Le calcul de cette instanciation est l'objet de l'algorithme de pronostic qui va être présenté dans la section suivante. Cette valeur sera ensuite utilisée pour conduire une action de maintenance éventuelle.

## 3.2 Algorithme de pronostic

Dans le paragraphe suivant, est présenté ce qui sera appelé la **version algébrique** de l'algorithme de pronostic. C'est à dire une suite d'opérations ayant pour but de fournir une valeur numérique de la RUL. Il s'agit d'une procédure de calcul semi-déterministe pouvant être vue comme une instantiation de la procédure d'inférence conduisant au calcul de l'expression 3.1 de la section précédente. Dans un second temps, la **version graphique** par RBD de l'algorithme de pronostic sera présentée.

### 3.2.1 Définitions et notations

Rappelons qu'à l'issue du processus d'apprentissage, on dispose d'un ensemble de lois de probabilité discrètes notées  $\Psi_x, x \in \{1, \dots, x_P\}$  correspondant aux lois de temps de séjour dans les différents états de la variable  $X$ . Ces lois ont été apprises à partir des bases de données d'observation disponibles par la méthode EMV ou l'algorithme EM si les données étaient incomplètes. Le but de l'algorithme est de calculer une estimation initiale de la RUL en se servant de ces lois de temps de séjour apprises, puis de mettre à jour cette RUL chaque fois qu'une nouvelle estimation de l'état du système est disponible par l'intermédiaire d'un des systèmes de diagnostic. Le processus se décompose donc en deux étapes :

- Fournir une estimation initiale de la RUL (phase d'initialisation).
- Définir une procédure pour mettre à jour cette RUL en tenant compte d'une nouvelle observation issue du diagnostic (phase de mise à jour).

Les principales notations qui seront utilisées dans la formalisation de l'algorithme sont les suivantes :

$oc$  : Observation courante

$op$  : Observation précédente

$t_{oc}$  : Instant de l'observation courante

$x_{oc} = x^{(t_{oc})}$  : Etat observé à l'instant courant

$S_x^{(t)}$  : Estimation du temps de séjour dans l'état  $x$  à l'instant  $t$

Pour construire l'algorithme, il faut de plus introduire les quantités suivantes :

$T_{SEC}^{(t)}$  : Estimation du temps écoulé dans l'état courant jusqu'à l'instant  $t$

$T_{SR}^{(t)}$  : Estimation du temps restant dans l'état courant depuis l'instant  $t$  jusqu'à la pro-

chaîne transition.

$R^{(t)}$  : Estimation de la somme des temps de séjour dans les états futurs (qui suivent  $x_{oc}$ )

$\Delta t$  : Intervalle de temps écoulé<sup>1</sup> entre les deux dernières observations :  $\Delta t = t_{oc} - t_{op}$

La figure 3.2 ci-dessous illustre certaines des variables introduites précédemment.

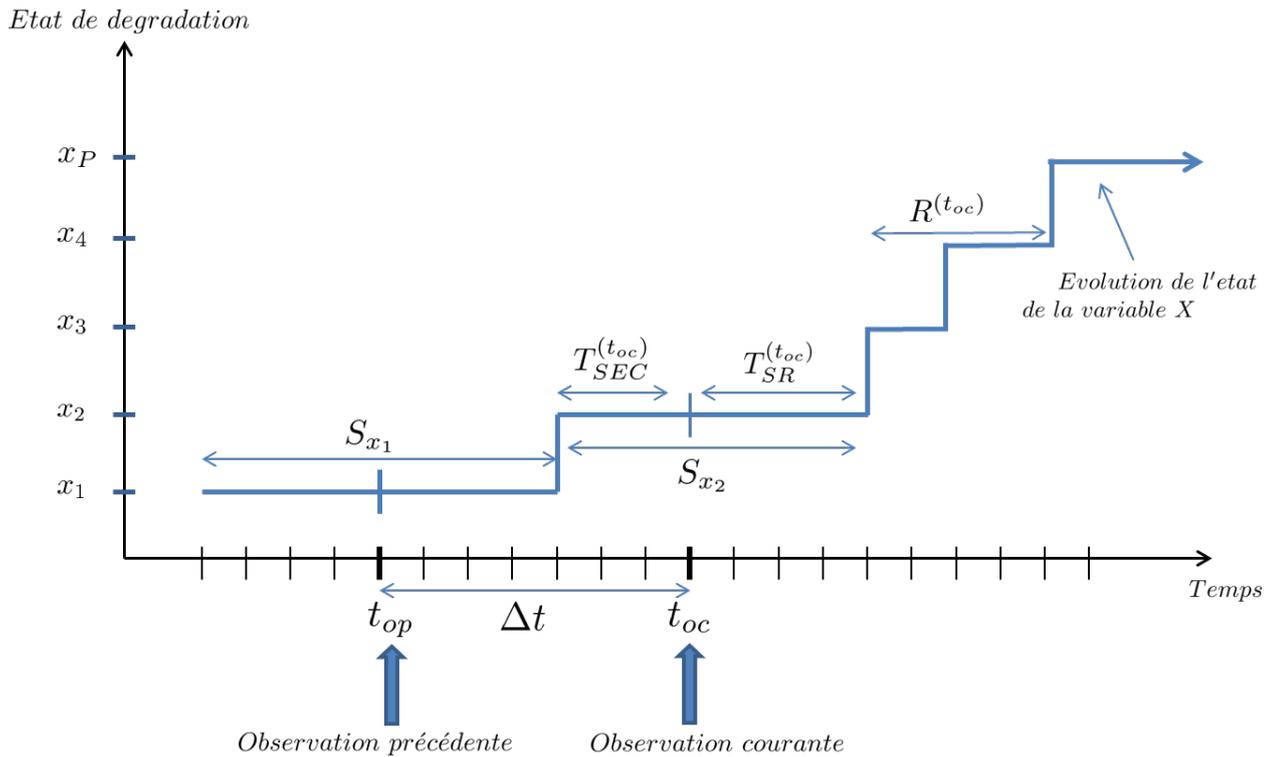


FIGURE 3.2: Illustration des variables précédentes

Précisons les notations suivantes :

- $\Psi_x$  est la distribution de temps de séjour dans l'état  $x$  correspondant à  $P(S^{(t)}|X^{(t)} = x)$
- $A \sim \Psi_x([a, b])$  signifie que la valeur pour  $A$  est tirée aléatoirement selon une fonction  $\Psi'$  qui est une normalisation de  $\Psi$  sur l'intervalle  $[a, b]$ . Elle est donc définie par :

1. Précisons que cette quantité étant relative aux instants courants  $t_{oc}$  et  $t_{op}$ , elle n'est donc pas supposée constante.

$$\forall k_i \in [a, b], \Psi'(k_i) = \frac{\Psi(k_i)}{\sum_{k_i \in [a, b]} \Psi(k_i)} \quad (3.2)$$

- $\alpha_j^{max}$  est le temps de séjour maximal possible dans l'état  $j$ . En d'autres termes :  $\alpha_j^{max} = \sup \{k, P(S^{(t)} = k | X^{(t)} = j) > 0\}$
- $U[a, b]$  est la distribution uniforme sur l'intervalle  $[a, b]$
- $t^{(0)}$  est l'instant d'initialisation de l'algorithme
- Dire que  $S_j$  est compatible, sous entend compatible avec la dernière observation disponible

Quelques hypothèses sont implicites :

- Le système reste au minimum 1 unité de temps dans chaque état.
- L'état du système ne peut évoluer que dans le sens croissant des états ( pas d'auto-réparation).
- L'état  $x_P$  est un état absorbant.
- Si le système transite précisément à l'instant d'un diagnostic, l'état visible à cet instant est celui d'après la transition.

La façon de construire la version algébrique de l'algorithme de pronostic a été en partie guidée par les contraintes liées à la future représentation graphique de cet algorithme par RBD. Il fallait, par exemple, que la taille du réseau ne dépende pas du nombre d'états considérés et qu'il y ait le moins de dépendances possibles entre les variables. Le choix des variables intermédiaires de travail dans la version algébrique a donc été fait dans le but de limiter au maximum la complexité de la représentation graphique ultérieure. C'est la raison pour laquelle, par exemple, le temps de séjour dans l'état courant a été coupé en deux variables séparées  $T_{SEC}$  et  $T_{SR}$ .

La RUL est globalement la somme des estimations des temps de séjour dans chaque état, moins le temps déjà écoulé depuis le lancement de l'algorithme. L'algorithme 1 présente l'ensemble des étapes pour son calcul. Les données d'entrée notées  $x^{(t_{oc})}$  sont des estimations discrètes de l'état courant du système.

### 3.2.2 Explications et interprétations

L'ensemble des étapes de la phase d'initialisation et de mise à jour est décrit.

**Algorithme 1** Algorithme de pronostic**Initialisation :** (t=0)

- 1:  $T_{SEC}^{(0)}$  est l'âge initial du système réglé par l'utilisateur
- 2:  $S_{x^{(0)}}^{(0)} \sim \Psi_{x^{(0)}}([T_{SEC}^{(0)} + 1, \alpha_{x^{(0)}}^{MAX}]$
- 3:  $T_{SR}^{(0)} = S_j^{(0)} - T_{SEC}^{(0)}$
- 4:  $\forall i = x^{(0)} + 1, \dots, x_P - 1, S_i^{(0)} \sim \Psi_i$  (tirage aléatoire initial des temps de séjour)
- 5:  $R^{(0)} = \sum_{j=x^{(0)}+1}^{x_P-1} S_j^{(0)}$
- 6:  $RUL^{(0)} = T_{SR}^{(0)} + R^{(0)}$

**Mise à jour :** Soit  $x^{(t_{oc})}$  un nouveau résultat de diagnostic disponible :

- 7: **if**  $\mathbf{x}^{(t_{oc})} = \mathbf{x}^{(t_{op})}$  (cas où aucun changement d'état n'est observé) **then**
- 8:  $T_{SEC}^{(t_{oc})} = T_{SEC}^{(t_{op})} + \Delta t$
- 9: **if** ( $T_{SR}^{(t_{op})} > \Delta t$ ) (cas où l'estimation précédente est compatible avec la nouvelle observation) **then**
- 10:  $T_{SR}^{(t_{oc})} = T_{SR}^{(t_{op})} - \Delta t$
- 11: **else**
- 12:  $T_{SR}^{(t_{oc})} = \Psi_{x^{(t_{oc})}}[T_{SEC}^{(t_{oc})} + 1, \alpha_{x_{oc}}^{MAX}] - T_{SEC}^{(t_{oc})}$
- 13: **end if**
- 14:  $R^{(t_{oc})} = R^{(t_{op})}$
- 15: **else if** ( $\mathbf{x}^{(t_{oc})} - \mathbf{x}^{(t_{op})} = 1$ ) (si un changement d'état est observé) **then**
- 16:  $R^{(t_{oc})} = R^{(t_{op})} - S_{x^{(t_{oc})}}^{(0)}$
- 17: **if** ( $T_{SR}^{(t_{op})} \leq \Delta t$ ) (si la précédente estimation de  $T_{SR}$  est compatible) **then**
- 18:  $T_{SEC}^{(t_{oc})} = \Delta t - T_{SR}^{(t_{op})}$
- 19: **else**
- 20:  $T_{SEC}^{(t_{oc})} = \Delta t - \Psi_{x^{(t_{op})}}[T_{SEC}^{(t_{op})} + 1, T_{SEC}^{(t_{op})} + \Delta t] + T_{SEC}^{(t_{op})}$
- 21: **end if**
- 22: **if**  $S_{x^{(t_{oc})}}^{(0)} > T_{SEC}^{(t_{oc})}$  (si le tirage initial du temps de séjour est compatible) **then**
- 23:  $T_{SR}^{(t_{oc})} = R^{(t_{oc})} - R^{(t_{op})} - T_{SEC}^{(t_{oc})}$
- 24: **else**
- 25:  $T_{SR}^{(t_{oc})} = \Psi_{x^{(t_{oc})}}[T_{SEC}^{(t_{oc})} + 1, \alpha_{x_{oc}}^{MAX}] - T_{SEC}^{(t_{oc})}$
- 26: **end if**
- 27: **else if**  $k = \mathbf{x}^{(t_{oc})} - \mathbf{x}^{(t_{op})} > 1$  (si un saut de plusieurs états est observé) **then**
- 28:  $T_{SEC}^{(t_{oc})} \sim U[0, \Delta t - k]$
- 29:  $T_{SR}^{(t_{oc})} = \Psi_{x^{(t_{oc})}}[T_{SEC}^{(t_{oc})} + 1, \alpha_{x_{oc}}^{MAX}] - T_{SEC}^{(t_{oc})}$
- 30:  $R^{(t_{oc})} = R^{(t_{op})} - \sum_{j=x^{(t_{op})}+1}^{x^{(t_{oc})}} S_j^{(0)}$
- 31: **end if**
- 32:  $RUL^{(t_{oc})} = T_{SR}^{(t_{oc})} + R^{(t_{oc})}$
- 33:  $\forall t \in ]t_{oc}, t_{fo}[ \quad RUL^{(t)} = RUL^{(t_{oc})} - (t - t_{oc})$  où  $fo$  désigne la future observation

**Etape d'initialisation** (lignes 1 à 6)

2 : Le temps de séjour estimé dans l'état initial doit être supérieur au temps déjà écoulé dans cet état. Le tirage aléatoire est effectué avec une restriction de la loi de temps de séjour pour forcer cette contrainte.

3 : Le temps restant dans l'état initial est égal au temps de séjour total estimé dans l'état initial moins le temps déjà écoulé  $T_{SEC}$ .

4 : L'ensemble des temps de séjour dans les états suivants est choisi par tirages aléatoires.

5 : Le système est dans l'état initial donc  $R^{(t)}$  est égal à la somme des temps de séjour dans les états suivants.

6 : La RUL est bien égale au temps restant dans l'état initial plus la somme des temps de séjour dans les états futurs.

**Étape de mise à jour** (lignes 7 à 33)

Trois situations différentes sont considérées pour la mise à jour de la RUL : le cas où il n'y a pas eu de changement d'état depuis la dernière observation, celui où il y a eu un changement d'état vers l'état suivant, ainsi que le cas extrême où il y aurait un saut de plusieurs états entre les instants des deux dernières observations.

**1<sup>er</sup> cas** : aucun changement d'état (lignes 7 à 14)

8 : Dans ce cas, le temps écoulé dans l'état (c'est à dire la quantité  $T_{SEC}$ ) est mis à jour de façon déterministe par extension de  $\Delta t$ . Ceci est illustré sur la figure 3.3.

12 : Concernant la quantité  $T_{SR}$ , il faut que sa valeur, calculée lors de la dernière observation, soit compatible avec la nouvelle observation.<sup>2</sup> Si tel est le cas, il est alors juste nécessaire de soustraire  $\Delta t$  de la quantité précédemment estimée. Ce cas est illustré en figure 3.4. Dans le cas contraire, il faut recalculer une nouvelle estimation du temps de séjour dans l'état courant. Le tirage selon  $\Psi$  est effectué sachant que ce temps de séjour doit être obligatoirement plus grand que le temps écoulé dans l'état jusqu'à l'instant courant. Il suffit alors de retirer cette valeur du temps déjà passé dans l'état à la nouvelle estimation du temps de séjour total qui vient d'être calculée pour obtenir la nouvelle valeur de  $T_{SR}$ . Ceci est illustré sur la figure 3.5.

14 : Dans le cas où il n'y a pas de changement d'état,  $R^{(t)}$  est évidemment invariant.

**2<sup>ème</sup> cas** : Changement d'état simple (lignes 15 à 26).

Comme le système a avancé d'un état, la nouvelle estimation de  $R^{(t)}$  est égale à l'an-

---

2. Le terme compatible signifie qu'il n'y a pas de contradiction entre la nouvelle observation et les précédentes estimations qui ont été faites.

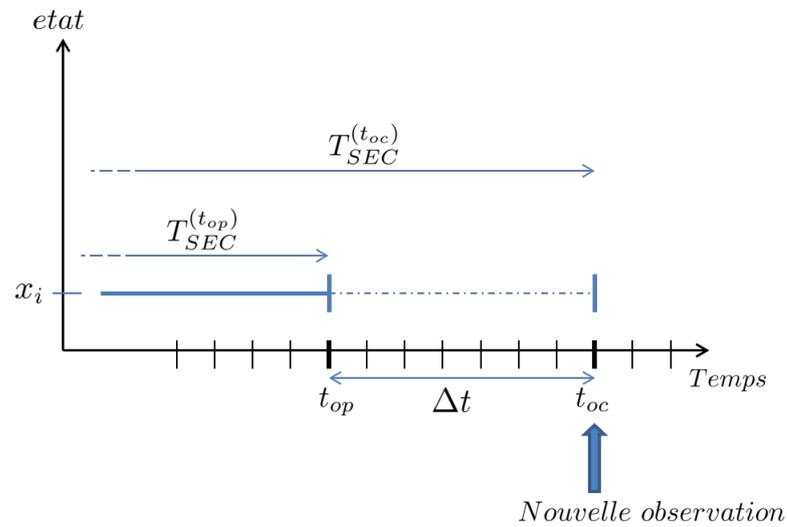


FIGURE 3.3: Illustration de l'extension de  $T_{SEC}$

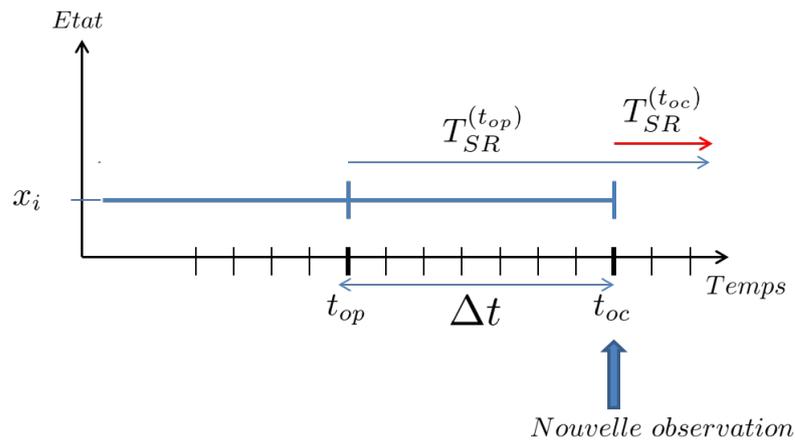


FIGURE 3.4: Réestimation de  $T_{SR}$  dans le cas compatible

cienne moins le temps de séjour initial estimé dans le nouvel état (L16).

Dans ce cas, la question est de parvenir à estimer à quel moment a eu lieu la transition d'état entre les deux derniers instants d'observation. Pour cela, il faut vérifier si l'estimation du temps restant dans l'état précédent ( $T_{SR}^{(t_{op})}$ ) est compatible avec la nouvelle observation ; c'est à dire est inférieure à  $\Delta t$ . Si c'est le cas, la nouvelle estimation de  $T_{SEC}^{(t_{oc})}$  est égale à  $\Delta t - T_{SR}^{(t_{op})}$  comme cela est illustré sur la figure 3.6.

Dans le cas d'une incompatibilité illustrée sur la figure 3.7, il faut recalculer le temps de séjour réel passé dans l'état précédent. On sait que l'instant de transition a eu lieu

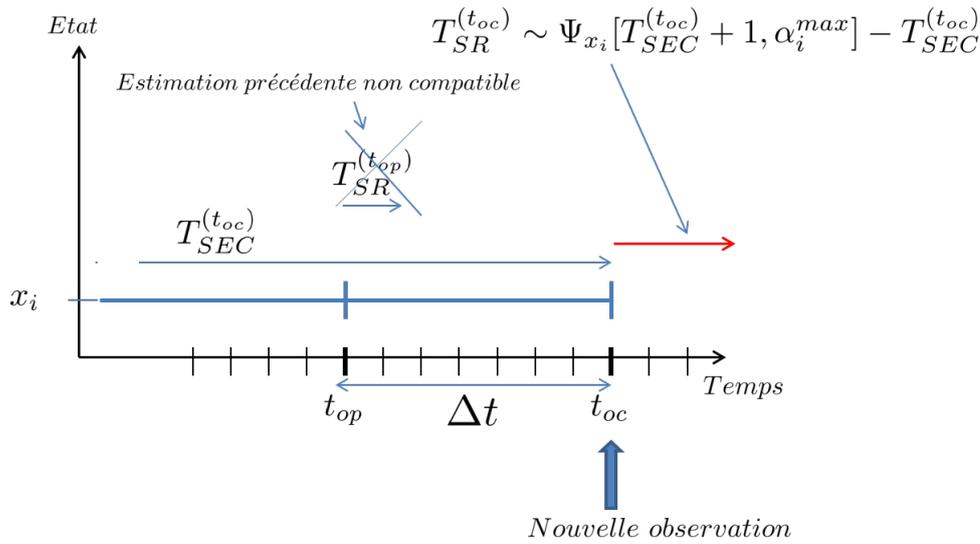


FIGURE 3.5: Réestimation de  $T_{SR}$  dans le cas incompatible

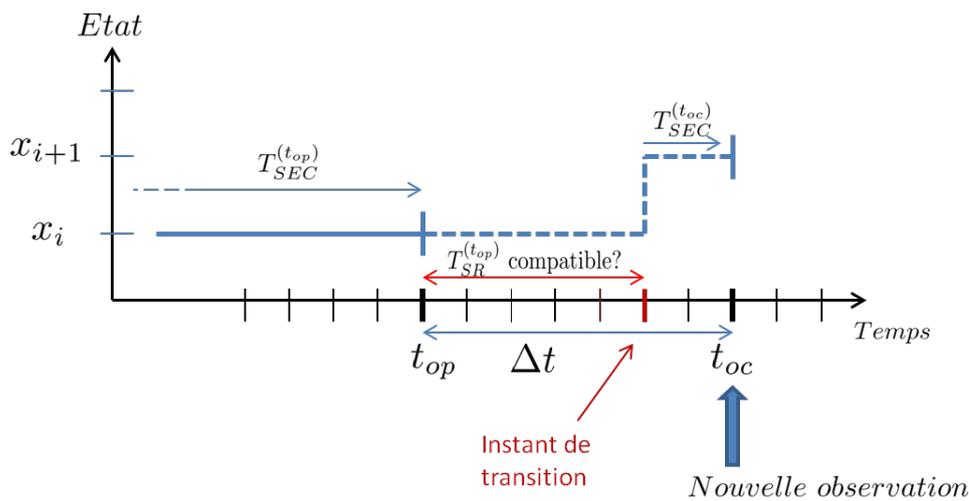


FIGURE 3.6: Cas de changement d'état avec compatibilité

entre  $t_{op}$  et  $t_{oc}$  ; donc, en faisant un tirage dans l'intervalle  $[T_{SEC}^{(t_{op})}, T_{SEC}^{(t_{op})} + \Delta t]$ ,  $T_{SR}^{(t_{op})}$  est recalculé. Puis, en soustrayant cette valeur de  $\Delta t$ , la valeur recherchée  $T_{SEC}^{(t_{oc})}$  est obtenue (ligne 20). Concernant la nouvelle valeur de  $T_{SR}$  dans le nouvel état, il faut vérifier que le temps de séjour initialement estimé dans ce nouvel état soit compatible, c'est à dire qu'il soit supérieur au temps déjà écoulé (condition de la ligne 22). En fonction de la réponse, celui-ci est recalculé de façon identique au cas où il n'y pas de changement d'état.

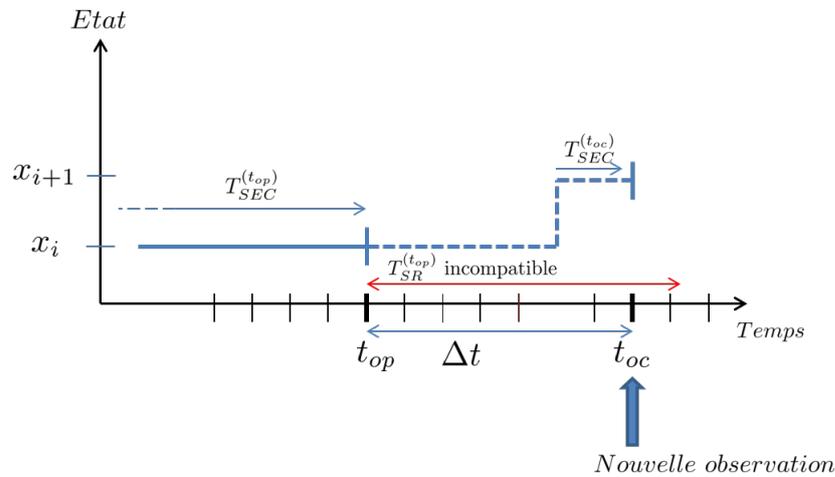


FIGURE 3.7: Cas de changement d'état avec incompatibilité

### 3<sup>ème</sup> cas : Changement d'états multiple (ligne 27 à 31)

La dernière situation concerne le cas où un saut de plusieurs états a été observé depuis la dernière observation. Cette situation est un cas extrême et suppose que soit le processus de dégradation se soit accéléré de façon brutale, soit que le pas de diagnostic ait été choisi trop grand. Dans cette situation, utiliser des lois de temps de séjour pour reconstituer le temps de séjour réel passé n'a pas beaucoup de sens dans la mesure où cette situation est déjà très en dehors du comportement normal ou du moins de celui qui a été capté lors du processus d'apprentissage. Par conséquent, l'instant réel de la dernière transition manquée entre  $t_{op}$  et  $t_{oc}$  est choisi par un tirage uniforme. La nouvelle durée  $T_{SEC}$ , c'est à dire le temps écoulé entre la dernière transition ratée et l'instant de la nouvelle observation, se fait selon cette distribution. L'intervalle de tirage de la loi uniforme tient compte du nombre d'états manqués puisque l'on suppose que l'on reste au moins une unité de temps dans chacun d'eux. Les quantités suivantes  $T_{SR}^{(t_{oc})}$  et  $R^{(t)}$  sont calculées de façon similaire aux cas précédents. Précisons enfin que l'hypothèse selon laquelle que l'état observé à l'instant d'une transition soit celui après la transition est la raison pour laquelle, à certaines lignes, les tirages aléatoires considérés doivent être supérieurs à  $T_{SEC} + 1$  et non pas simplement à  $T_{SEC}$ .

### 3.2.3 Calculs d'indicateurs

L'algorithme précédemment défini calcule donc une RUL notée ( $RUL^{(t)}$ ). Nous allons donc maintenant nous intéresser à la nature de l'erreur commise sur cette prédiction ou en

d'autres termes, à la variance des résultats. Notons  $O_n$  la n-ième observation disponible et  $\Delta_n t = t_{O_n} - t_{O_{n-1}}$ .

L'algorithme de pronostic, après quelques simplifications algébriques permet d'obtenir la formule suivante.

$$RUL^{(t_{O_n})} = \underbrace{S_j^{(t_{O_n})} + \sum_{j=x^{(t_{O_n})+1}^{x_F-1} S_j^{(0)}}_{\text{Somme des temps de séjour estimés dans chaque état depuis l'état courant}} - \underbrace{\sum_{k=1}^n \Delta_k t}_{\text{Temps écoulé jusqu'à la dernière observation}} \quad (3.3)$$

$$\text{avec } S_{x_{oc}}^{(t_{O_n})} = T_{SEC}^{(t_{O_n})} + T_{SR}^{(t_{O_n})}.$$

La RUL calculée est la somme des dernières estimations des temps de séjour dans chaque état moins le temps écoulé jusqu'à la dernière observation. Les distributions de temps de séjour sont bien supposées indépendantes les unes des autres. De plus, le dernier terme de l'équation 3.3 étant déterministe, sa variance est nulle. On a donc :

$$Var(RUL^{(t_{O_n})}) = Var(S_{x_{O_n}}^{(t_{O_n})}) + \sum_{j=x^{(t_{O_n})+1}^{x_F-1} Var(S_j^{(0)}) \quad (3.4)$$

La variance de la RUL calculée dépend uniquement des variances sur l'ensemble des estimations de temps de séjour sur chaque état. Améliorer la qualité du calcul de pronostic nécessitera donc de réduire la variance des lois de temps de séjour considérées. Ceci sera un des objectif du nouveau modèle de dégradation présenté dans la section suivante.

Soit  $X$  une quantité réelle donnée et  $\bar{X}$  l'estimation de cette quantité. La notation  $\epsilon(X) = (\bar{X} - X)$  désignera la distance entre la quantité estimée et la quantité réelle inconnue. Précisons bien qu'il ne s'agit pas d'une valeur absolue. Cette quantité peut être positive ou négative.

$$\epsilon(RUL)^{(t_{O_n})} = \epsilon(S_{x_{O_n}}^{(t_{O_n})}) + \sum_{j=x^{(t_{O_n})+1}^{x_F-1} \epsilon(S_j^{(0)}) \quad (3.5)$$

Les termes de la somme peuvent être aussi bien positifs (surestimation) que négatifs (sous estimation). La formule précédente a simplement pour but d'attirer l'attention sur le fait que l'on peut être en présence de cas pour lesquels, si les erreurs (positives et négatives) commises sur les estimations initiales des temps de séjour dans chaque état se compensent, alors l'erreur totale alors commise sur la prédiction initiale de la RUL

(noté  $\epsilon(RUL)$ ) peut être très faible. Une bonne prédiction initiale peut donc être liée à de mauvaises raisons. Dans ce type de situation, il est probable qu'au premier changement d'état, la RUL réajustée soit moins bonne que la RUL initiale. La convergence de l'algorithme vers la RUL réelle n'est donc en théorie pas monotone.

Il est possible dans l'algorithme de remplacer l'ensemble des tirages aléatoires effectués selon les lois de temps de séjour par un choix déterministe égal à la moyenne de ces mêmes distributions. Pour chaque temps de séjour, la moyenne de la distribution étant la valeur qui minimise le risque d'erreur. De cette façon, l'estimation de la RUL obtenue minimise le risque d'erreur de prédiction d'un point de vue probabiliste. C'est cette méthode qui sera utilisée pour toutes les utilisations de l'algorithme de pronostic qui vont suivre.

### **Observations générales sur le fonctionnement de l'algorithme.**

Analysons d'un point de vue global le comportement de l'algorithme de pronostic. Les tirages initiaux des temps de séjours ont un impact fort sur l'évolution de l'erreur de prédiction et sur la convergence des calculs vers la RUL réelle. En effet, il ne peut y avoir de réduction de l'erreur commise entre la RUL calculée initialement et la RUL réelle inconnue que lors d'un réajustement des temps de séjour. Or, ceux-ci ne sont réajustés que dans 2 situations :

- Un changement d'état
- Une extension d'un temps de séjour

Le premier cas implique que l'algorithme ne peut être efficace que si le système dispose d'un minimum d'état intermédiaires observables avant la panne. Un système contenant uniquement 2 états (*OK* et *Panne*) ne permet aucun réajustement possible avant la panne, et l'erreur commise tout au long du processus est entièrement liée à la somme de la variance de la loi de temps de séjour de l'état *OK*.

Le second cas signifie, qu'il y a un meilleur potentiel de réajustement si un temps de séjour est sous-estimé que dans le cas où il est surestimé. En effet, un temps de séjour surestimé ne pourra être corrigé qu'à la prochaine transition, tandis qu'un temps de séjour sous-estimé se réajustera par extension de façon progressive avec le temps qui s'écoule. Les réestimations se font alors selon les fonctions  $\Psi_x$  mais dans des intervalles de taille plus petites (dont la borne min est  $T_{SEC}$ ).

Dans les deux cas, les réestimations de temps de séjour ont pour effet de réduire la variance des distributions considérées pour les tirages aléatoires et donc d'améliorer la

précision de la RUL. Comme nous l'avons dit précédemment, la variance de la RUL est égale à la somme des variances de chaque temps de séjour dans chaque état. Plus cette variance est élevée plus le risque d'erreur de prédiction est fort. La variance des distributions apprises dépend des données utilisées pour l'apprentissage ainsi que de la façon dont le processus de dégradation est modélisé. Pour manipuler des données à faible variance, il faut s'assurer que celles-ci soient le plus homogènes possible ; c'est à dire correspondent à des observations sur des composants dont les comportements (les dynamiques de dégradation) sont suffisamment proches. Une base d'observations sur des systèmes de comportements variables va conduire très probablement à une grande variance des distributions obtenues lors de l'apprentissage et donc à une mauvaise qualité du pronostic effectué. Nous avons donc voulu développer un modèle de dégradation plus précis qui puisse, dans certaines situations, mieux capter le processus de dégradation notamment dans le cas où les bases d'observations ne sont pas totalement homogènes. L'objectif principal est principalement d'obtenir des distributions de temps de séjour avec une plus faible variance. La description de ce nouveau modèle est l'objet de la section suivante.

Enfin, cet algorithme fournit une valeur numérique. Or une prédiction n'a d'intérêt en pratique que si celle-ci est associée à un intervalle de confiance (c'est à dire une mesure de la variance de la quantité aléatoire). Ces quantités peuvent être fournies par un calcul de la distribution de la RUL pouvant être obtenu par l'équation 3.1 en faisant varier l'horizon. Ce calcul s'effectue par inférence sur le RBD de la figure 3.1. Le pic de la distribution obtenue doit alors coïncider avec la valeur calculée par l'algorithme de pronostic lorsque celui-ci est utilisé en remplaçant les tirages aléatoires par des tirages de moyenne.

### **3.3 Intégration de lois de temps de séjour conditionnelles (LTSC)**

#### **3.3.1 Idée principale de l'approche**

Le modèle graphique de durée (MGD) présenté dans les sections précédentes repose sur l'hypothèse selon laquelle les lois de temps de séjour (LTS), dans chaque état, sont indépendantes (IDP) les unes des autres. Cette hypothèse est aussi présente dans d'autres modèles de dégradation. Les processus stochastiques couramment utilisés en fiabilité (processus gamma, processus de Wiener) sont des processus à accroissements

stationnaires indépendants. Soient deux instants  $t_1$  et  $t_2$ . l'accroissement du processus entre  $t_1$  et  $t_1 + \Delta t$  ne dépend que de  $\Delta t$ . La loi de l'accroissement entre  $t_2$  et  $t_2 + \Delta t$  est indépendante de l'accroissement précédent. Cette hypothèse est donc équivalente à celle du modèle discret dans le cas d'un espace d'états continu. La figure 3.8 illustre les similitudes de ces hypothèses entre les cas discrets et continus.

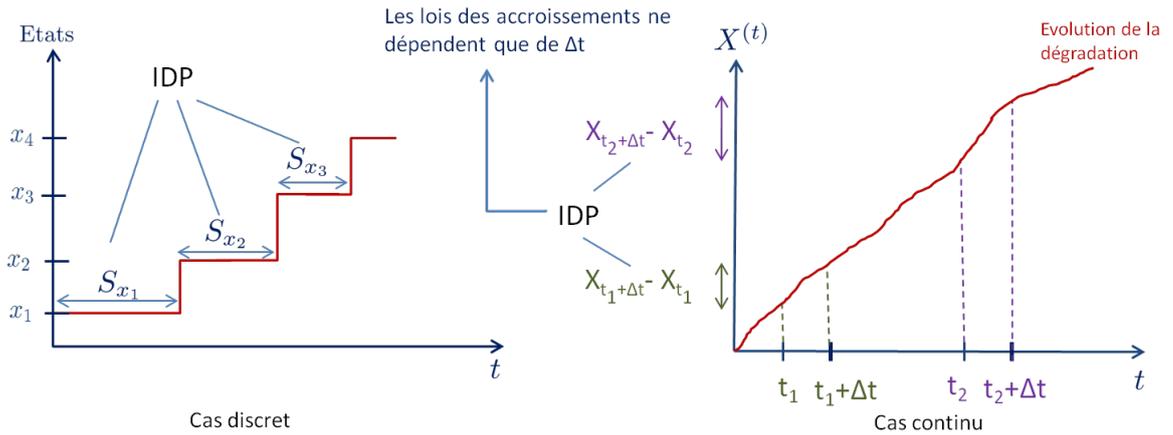


FIGURE 3.8: Illustration de l'hypothèse d'IDP dans les cas discrets et continus

Cette hypothèse d'indépendance, que l'on retrouve aussi bien dans le modèle graphique de durée que dans certains modèles continus, est assez forte et peut conduire dans certains cas à des imprécisions lors de la modélisation du processus de dégradation. Considérons, par exemple, un simple système à états discrets comme décrit précédemment possédant quatre états et donc trois LTS représentées sur la figure 3.9. Imaginons disposer d'une

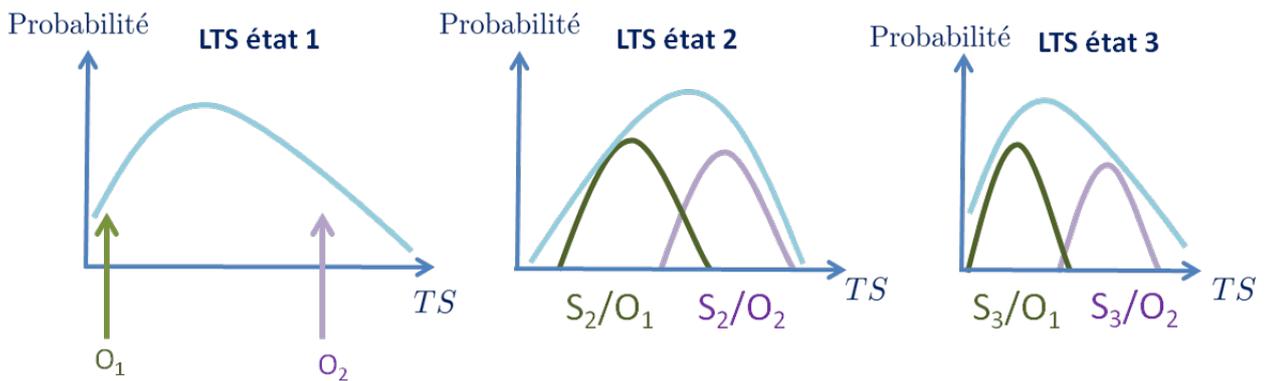


FIGURE 3.9: Exemple illustratif

suite de diagnostics de l'état du système. Si le temps de séjour passé dans l'état 1 est très inférieur à la moyenne de la distribution qui a été apprise (cas  $O_1$  en beige). Alors le processus de dégradation qui s'exerce sur le système est plus rapide que la moyenne. Un tel processus n'a aucune raison d'être ponctuel et cette dynamique rapide pourrait probablement continuer à s'exercer sur le système. Par conséquent, le temps de séjour passé dans l'état suivant (état 2) a de fortes chances d'être aussi plus faible que la moyenne et ainsi de suite dans les états suivants. A l'inverse, si le temps de séjour dans l'état 1 est très grand, on peut s'attendre pour les mêmes raisons à observer des temps de séjour longs dans les autres états puisque le système a de fortes chances d'être dans un processus de dégradation lent (cas de l'observation  $O_2$ ). Autrement dit, la connaissance de  $O_1$  est une information additionnelle qui est susceptible de modifier la distribution des temps de séjour dans les états suivants en réduisant leur variance.

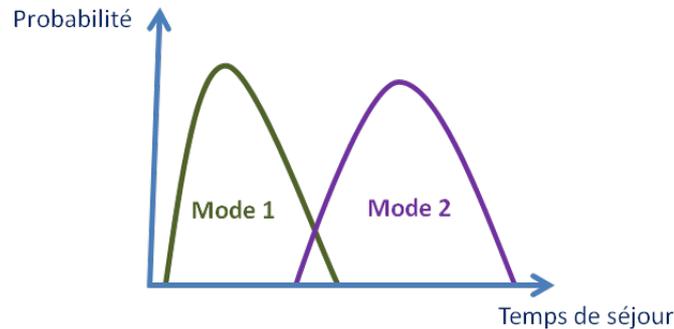
Ceci amène donc à considérer l'hypothèse que les temps de séjour ne sont pas indépendants les uns des autres. Les considérer comme indépendants amènerait alors à perdre de l'information et par conséquent à réduire la qualité de convergence de l'algorithme de pronostic.

Nous voulons donc trouver un moyen de modéliser ce type de situation faisant intervenir plusieurs **modes de dégradation** (ou profils de dégradation) se mélangeant, en rejetant l'hypothèse d'IDP entre les LTS des différents états intermédiaires.

### 3.3.2 Définition d'une nouvelle approche

Dans ce paragraphe, les mêmes notations que celles introduites dans le chapitre 2 sont utilisées pour détailler la phase d'apprentissage des paramètres du modèle MGD. Soit un système à  $n$  états et  $Y$  une base de  $L$  séquences d'observations disponibles de ce système. On reprend les notations de la section 1.2.1 du chapitre 2 :  $Y = (y_1^{(l)}, y_2^{(l)}, \dots, y_n^{(l)})_{l=1, \dots, L}$  où  $y_i^{(l)}$  représente le temps de séjour à l'état  $i$  observé sur la  $l$ -ème séquence.

Dans le but de manipuler des LTS qui ne soient plus indépendantes, la notion de lois de temps de séjour conditionnelles (LTSC) est introduite. La loi de temps de séjour dans l'état initial (état 1) est désormais un modèle de mélange. Un exemple d'un tel modèle est représenté sur la figure 3.10. L'apprentissage du mélange se fait par un algorithme de type EM [17]. Le choix du nombre de composantes peut être déterminé soit par l'utilisateur à partir de la forme de la LTS de l'état initial apprise par un simple EMV ou bien par un critère de sélection de type BIC ou AIC [60].  $n_M$  désignera le nombre de composantes du mélange appris et  $m_j$  la  $j$  ème composante du mélange correspondant



**FIGURE 3.10:** Exemple de modèle de mélange pour l'état initial possédant deux composantes

au  $j$ -ème mode de dégradation considéré. Par conséquent :

$$\Psi_1 = \sum_{i=1}^{n_M} \omega_i m_i$$

où les  $\omega_i$  désignent les poids du mélange. Cet apprentissage du modèle de mélange initial a pour but d'identifier les différentes dynamiques de dégradation à l'œuvre sur le système étudié, pour ensuite parvenir à les suivre dans le temps. Chaque mode est identifié par une composante du mélange initial. Les LTS dans les états suivants seront des lois conditionnées par ces modes. La problématique suivante est celle de l'apprentissage de ces lois conditionnelles à partir de la base de données. Une première possibilité consiste à apprendre un modèle de mélange pour chaque état. Le problème que cela pose est que cela induit ensuite la nécessité de relier chaque composante du mélange obtenu à un mode, ce qui peut s'avérer très délicat (nombre de composantes différentes, ordre des composants inversés, . . .). Pour faire cela avec une procédure automatique, il faudrait être certain que les mélanges de chaque mode sont rangés dans le même ordre pour chaque état. De plus, si jamais les lois des états suivants ne contiennent pas de mélanges ou bien des mélanges avec un nombre de composants différents, il semble difficile de suivre les modes détectés initialement dans chaque état. Comment serait-il possible d'affecter ces derniers au nouveau modèle de mélange obtenu ? De plus, une telle procédure d'apprentissage supposerait des apprentissages de modèles de mélanges indépendants les uns des autres que l'on essaierait ensuite de reconnecter entre eux. Ceci est assez éloigné de l'idée principale de la méthodologie choisie qui repose sur une volonté de suivre un même processus de dégradation identifié à travers l'ensemble des états parcourus.

Une autre approche a donc été utilisée. Celle-ci fait l'hypothèse selon laquelle les

modes de dégradation sont des processus continus qui s'exercent sur le système du début jusqu'à la fin mais que les diagnostics (observations) ne sont sensibles qu'au mode le plus pénalisant, les autres modes étant de fait masqués. **Cela implique donc que chaque observation de la base de données est caractérisé par un seul mode, le mode dominant.**<sup>3</sup> De ce fait le modèle de mélange obtenu pour l'état initial nous permet d'associer à chaque observation de la base un mode actif le plus probable. En notant  $m_{y^{(l)}}$  le mode associé à la l-ième observation  $y^{(l)}$ , cela peut s'écrire :

$$m_{y^{(l)}} = \arg \max_k (P(\text{mode} = k | S_1 = y_1^{(l)})) \quad (3.6)$$

Les probabilités a posteriori apparaissant dans l'expression précédente sont obtenues grâce à l'algorithme EM. Plutôt que de choisir le mode le plus probable, il est possible de choisir le mode par un tirage aléatoire pondéré par les probabilités a posteriori du mélange. Il est donc possible, à partir de là, de partitionner la base de données d'observation en  $n_M$  groupes, chaque groupe étant associé à un mode. La notation  $Y_k$  désignera le sous groupe associé au k-ème mode.

$$Y_k = \{y^{(l)}, m_{y^{(l)}} = k\} \quad (3.7)$$

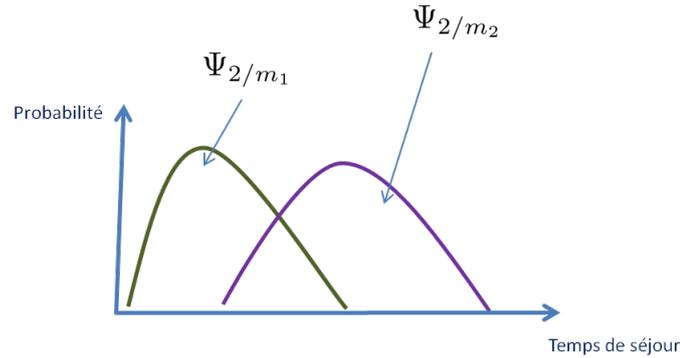
Les lois dans les états suivants, conditionnellement à chaque mode, sont apprises par la méthode EMV, sur le sous ensemble de la base qui lui est associé. Soit  $\Psi_{i/m_k}$  la LTS de l'état i sachant le mode k.  $\forall i > 1$ , la LTS  $\Psi_{i/m_k}$  est donc apprise par EMV sur l'ensemble  $Y_k$ . A l'issue de ce nouveau processus d'apprentissage, on dispose alors d'un modèle de mélange pour l'état initial ainsi qu'un ensemble de lois conditionnelles pour les états suivants. Un exemple est représenté sur la figure 3.11. Pour un état donné, le mélange formé par l'ensemble des LTSC de cet état sera aussi considéré. Dans ce cas, les poids du mélange initial sont implicitement transférés sur l'ensemble des mélanges des états suivants. La loi de temps de séjour global de l'état x s'écrira alors.

$$\Psi_x = \sum_{k=1}^{n_M} w_k \Psi_{x/m_k} \quad (3.8)$$

Pour résumer, les mélanges des état non initiaux sont reconstitués à partir des LTSC apprises par EMV.

---

3. Cette hypothèse est spécifique au processus d'apprentissage. Nous verrons plus loin qu'elle n'est pas nécessaire aux calculs de pronostic.



**FIGURE 3.11:** Type de distributions obtenues dans l'état 2 considérant l'existence de 2 modes de dégradation .

### 3.3.3 Représentation graphique de l'approche proposée

Une représentation de ce nouveau modèle à base de LTSC est maintenant proposée. Celui-ci est une modification du MGD et sera donc nommé modèle graphique de durée conditionnelle (MGDC) car incluant l'utilisation des lois de temps de séjour conditionnelles (LTSC). Pour tenir compte de la notion de "mode de dégradation actif (ou prépondérant)" une variable notée  $C$  a été intégrée à la structure standard des MGD.  $C^{(t)} \in \{1, \dots, n_M\}$  représente le mode de dégradation actif à l'instant  $t$ . Le nouveau modèle de dégradation ainsi construit est représenté sur la figure 3.12.<sup>4</sup>

Comme dans le chapitre précédent, le rôle de chaque nouvelle flèche présente dans le modèle va être détaillé.

A l'instant initial, en fonction du temps de séjour choisi  $S_0^{(0)}$  à l'instant initial, le mode prépondérant est choisi d'après les probabilités a posteriori du modèle de mélange de l'état initial appris  $\Psi_1$  (flèche 0). Tant qu'il n'y a pas de changement d'état, le mode actif reste inchangé (flèche 1). Dans le cas d'un changement d'état (flèches 2 et 5) le nouveau temps de séjour  $S^{(t+1)}$  est tiré en fonction du mode actif à l'instant précédent  $C^{(t)}$  (flèche 3). Le choix de ce nouveau temps de séjour va déterminer le nouveau mode actif  $C^{(t+1)}$  (flèche 4). Les lois initiales et de transition des nœuds qui sont modifiées par rapport à l'approche standard MGD ( $C$  et  $S$ ) sont maintenant définies.

4. Il ne faudra pas confondre le nœud  $C$  et le nœud  $\tilde{C}^{(t)}$  qui sera introduit plus loin et qui représentera le mode calculé par l'algorithme de pronostic

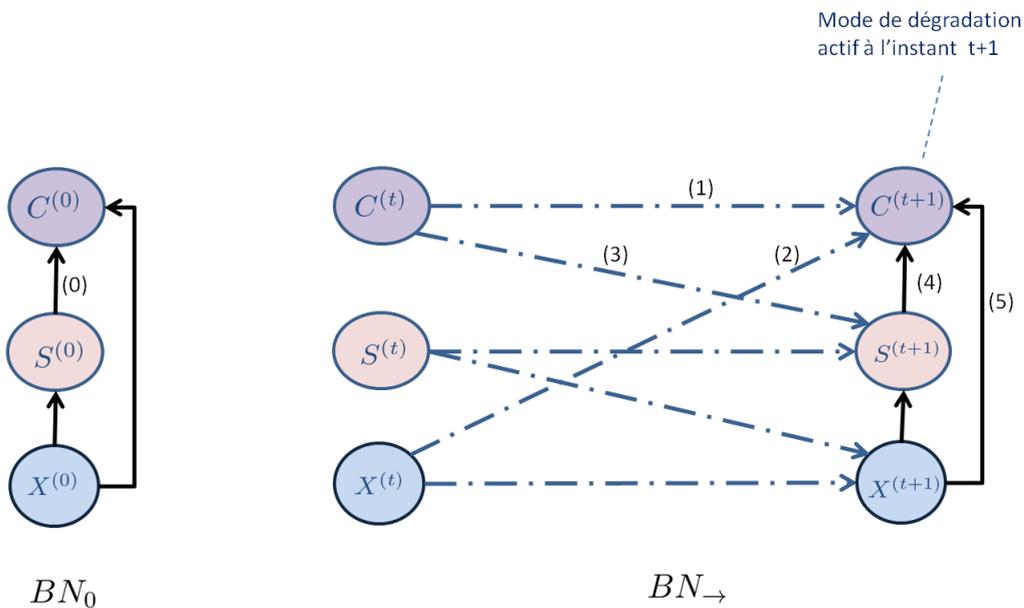


FIGURE 3.12: Modèle graphique de durée conditionnel (MGDC)

Lois initiales :

$$p(C^{(0)=c} | X^{(1)} = x, S^{(1)} = s) = C_{init}(x, s, c) = \begin{cases} 1 & \text{si } c = \arg \max_k (P(\text{mode} = k | s, x)) \\ 0 & \text{sinon} \end{cases}$$

-  $C_{init}$  est une matrice de taille  $n * T_{max} * n_M$  construite à partir des probabilités a posteriori du mélange de l'état initial<sup>5</sup>.

Loi de transition :

$$p(C^{(t+1)} = c | C^{(t)} = c', X^{(t)} = x', S^{(t+1)} = s, X^{(t+1)} = x) = M^{\rightarrow}(c', x, s, c) = \begin{cases} C_{mode}(x, s, c) & \text{si } x \neq x' \\ I(c', c) & \text{si } s' \geq 2 \end{cases}$$

$$\text{avec } C_{mode}(x, s, c) = C_{init}(x, s, c)$$

L'ajout du nœud  $C^{(t)}$  conduit à une modification de la loi de transition de  $S^{(t)}$  comme

5. Rappelons que  $n_M$  est le nombre de composante du mélange, soit encore le nombre de modes détectés à l'apprentissage

indiqué ci-dessous.

$$p(S^{(t+1)} = s | X^{(t+1)} = x, S^{(t)} = s', C^{(t)} = c') = F^{\rightarrow}(s', c', x, s) = \begin{cases} F_{cond}(c', x, s) & \text{if } s' = 1 \\ C(s', s) & \text{if } s' \geq 2 \end{cases}$$

-  $F_{cond}$  est une matrice de taille  $n_M * n * T_{max}$  où la ligne  $F_{cond}(c', x, :)$  est la distribution conditionnelle dans l'état  $x$  sachant le mode  $c'$  notée  $\Psi_{x/c'}$ .

A chaque transition, le temps de séjour écoulé dans l'état précédent ( $S^{(t)}$ ) est utilisé pour en déduire le nouveau mode actif dans le nouvel état ( $C^{(t+1)}$ ). Puis, le nouveau temps de séjour dans le nouvel état est choisi.  $S^{(t+1)}$  est donc tiré en fonction de la loi de temps séjour conditionnelle dans le nouvel état ( $X^{(t+1)}$ ) sachant le dernier mode actif détecté ( $\Psi_{X^{(t+1)}/C^{(t)}}$ ).

Le modèle de dégradation proposé peut donc tenir compte de l'existence éventuelle de plusieurs modes de dégradation. Le modèle de mélange de l'état initial a pour fonction d'identifier ces modes puis réduire la variance des distributions dans les états suivants par l'utilisation de lois conditionnelles. une réduction de la variance conduit a obtenir un pronostic de meilleure précision. Ce nouveau modèle doit être vu comme un outil exploratoire permettant de rechercher l'existence de dépendances entre les lois de temps de séjour et de les exploiter. Il n'a d'intérêt que si les distributions conditionnelles obtenues ont des formes intéressantes. C'est à dire avec une variance plus faible que le modèle classique, et que l'ensemble des mélanges obtenus aient des supports dont les intersections avec les supports voisins soient petits. Dans le contraire, il est préférable de rester sur un modèle classique (MGD) qui donne alors de meilleurs résultats.

### 3.3.4 Nouvelle version de l'algorithme de pronostic

Il est maintenant nécessaire d'adapter l'algorithme de pronostic à l'utilisation du nouveau modèle intégrant les LTSC et la notion de mode de dégradation. Les notations précédentes sont conservées. La notation  $E(\Psi)$  désigne l'espérance de la distribution  $\Psi$ . L'algorithme 2 décrit cette nouvelle version. Le paragraphe suivant décrit les principales modifications apportées par rapport à l'ancienne version.

*Explications du fonctionnement*

**Algorithme 2** Version étendue de l'algorithme de pronostic intégrant les LTSC**Initialisation :**

- 1:  $T_{SEC}^{(0)}$  (initialisé par l'utilisateur, dépend du passé du composant dans l'état estimé)
- 2:  $S_{x^{(0)}}^{(0)} \sim \Psi_{x^{(0)}}([T_{SEC}^{(0)} + 1, \alpha_{x^{(0)}}^{MAX}]$
- 3:  $m_{x^{(0)}} = \arg \max_k (\Psi_{x^{(0)}}(S_{x^{(0)}}^{(0)}))$
- 4:  $T_{SR}^{(0)} = S_{x^{(0)}}^{(0)} - T_{SEC}^{(0)}$
- 5:  $m^{(op)} = m_{x^{(0)}}$
- 6: **for**  $i=x^{(0)} + 1 \dots x_P-1$  **do**
- 7:      $S_i^{(0)} \sim \Psi_{i/m_{i-1}}$
- 8:      $m_i = \arg \max_k (\Psi_{i/k}(S_i^{(0)}))$
- 9: **end for**
- 10:  $R^{(0)} = \sum_{j=x^{(0)}+1}^{x_P-1} S_j^{(0)}$
- 11:  $RUL^{(0)} = T_{SR}^{(0)} + R^{(0)}$

**Mise à jour :** Soit  $x^{(t_{oc})}$  un nouveau résultat de diagnostic disponible :

- 12: **if**  $x^{(t_{oc})} = x^{(t_{op})}$  (cas où aucune transition n'est observée) **then**
- 13:      $T_{SEC}^{(t_{oc})} = T_{SEC}^{(t_{op})} + \Delta t$
- 14:      $m_C = \arg \max_k \Psi_{x_{oc}/k}(T_{SEC}^{(t_{oc})})$
- 15:     **if**  $E(\Psi_{x_{oc}/m_C}) > E(\Psi_{x_{oc}/m^{(op)}})$  et  $\Psi_{x_{oc}/m_C}(T_{SEC}^{(t_{oc})}) \neq 0$  **then**
- 16:          $m^{(oc)} = m_C$
- 17:     **end if**
- 18:     **if**  $(T_{SR}^{(t_{op})} > \Delta t)$  (cas où la dernière observation est compatible) **then**
- 19:          $T_{SR}^{(t_{oc})} = T_{SR}^{(t_{op})} - \Delta t$
- 20:     **else**
- 21:          $T_{SR}^{(t_{oc})} = \Psi_{x^{(t_{oc})}/m^{(oc)}}[T_{SEC}^{(t_{oc})} + 1, \alpha_{x_{oc}}^{MAX}] - T_{SEC}^{(t_{oc})}$
- 22:     **end if**
- 23:      $R^{(t_{oc})} = R^{(t_{op})}$
- 24: **else if**  $(x^{(t_{oc})} - x^{(t_{op})} = 1)$  (si une transition est observée) **then**
- 25:      $R^{(t_{oc})} = R^{(t_{op})} - S_{x^{(t_{oc})}}^{(0)}$
- 26:     **if**  $(T_{SR}^{(t_{op})} \leq \Delta t)$  **then**
- 27:          $T_{SEC}^{(t_{oc})} = \Delta t - T_{SR}^{(t_{op})}$
- 28:     **else**
- 29:          $T_{SEC}^{(t_{oc})} = \Delta t - \Psi_{x^{(t_{op})}}[T_{SEC}^{(t_{op})} + 1, T_{SEC}^{(t_{op})} + \Delta t] + T_{SEC}^{(t_{op})}$
- 30:     **end if**
- 31:      $m^{(oc)} = \arg \max_k (\Psi_{x^{(op)}/k}(T_{SEC}^{(op)} + \Delta t - T_{SEC}^{(oc)}))$
- 32:     **if**  $R^{(t_{oc})} - R^{(t_{op})} > T_{SEC}^{(t_{oc})}$  **then**
- 33:          $T_{SR}^{(t_{oc})} = R^{(t_{oc})} - R^{(t_{op})} - T_{SEC}^{(t_{oc})}$
- 34:     **else**
- 35:          $T_{SR}^{(t_{oc})} = \Psi_{x^{(t_{oc})}/m^{(oc)}}[T_{SEC}^{(t_{oc})} + 1, \alpha_{x_{oc}}^{MAX}] - T_{SEC}^{(t_{oc})}$
- 36:     **end if**
- 37:     **end if**
- 38: **end if**
- 39: **if**  $m^{(oc)} = m^{(op)}$  **then**
- 40:      $R^{(t_{oc})} = R^{(t_{op})} - \sum_{j=x^{(t_{op})}+1}^{x^{(t_{oc})}} S_j$
- 41: **else**
- 42:      $\forall j \in [x^{(oc)} + 1, x_P - 1], S_j \sim \Psi_{j/m^{(oc)}}$
- 43:      $R^{(t_{oc})} = \sum_{j=x^{(oc)}+1}^{x_P-1} S_j$
- 44: **end if**
- 45:  $m^{(op)} = m^{(oc)}$
- 46: Enfin  $RUL^{(t_{oc})} = T_{SR}^{(t_{oc})} + R^{(t_{oc})}$
- 47:  $\forall t \in ]t_{oc}, t_{fo}[$   $RUL^{(t_{op})} = RUL^{(t_{oc})} - (t - t_{oc})$  où  $fo$  est la future observation.

Lors de la phase l'initialisation, le temps de séjour dans l'état initial est tiré selon la loi  $\Psi_1$  qui est le modèle de mélange appris par l'algorithme EM. Ce tirage tient compte de la connaissance de  $T_{SEC}^{(0)}$  fournie par l'utilisateur (L2) puis, en fonction de ce temps de séjour, un mode actif initial le plus probable (L3) est choisi.

Le temps de séjour de l'état suivant est ensuite tiré conditionnellement à ce nouveau mode. A chaque nouveau tirage, le mode le plus probable est réévalué, puis la loi conditionnelle adaptée est utilisée. Observons qu'il ne peut y avoir de changement de mode dans cette phase de réestimation des modes uniquement si le temps de séjour tiré se trouve sur le domaine d'intersection de deux composantes du mélange d'une distribution de temps de séjour. (L6 à L8).

Le tirage initiale de  $T_{SR}$  et de  $R^{(t)}$  se fait de façon identique à la version initiale de l'algorithme (L10-L11)

Lors de la phase de mise a jour de la RUL, les principales modifications concernent l'ensemble des tirages de  $T_{SR}$ . Les lois générales de type  $\Psi_x$  sont remplacées par les lois conditionnelles selon le dernier mode détecté comme actif ( $\Psi_x/m_{oc}$ ) (L21 et L36).

Il y a deux situations de détection de changement de mode :

- Le cas d'un changement d'état vers l'état suivant : dans ce cas, le temps de séjour passé dans l'état précédent est utilisé pour réestimer un nouveau mode le plus probable (L32)
- Le cas sans changement d'état : dans ce cas la quantité  $T_{SEC}$  est prolongée de  $\Delta t$  (tout comme dans la version initiale de l'algorithme). Le mode le plus probable est alors calculé en fonction de cette nouvelle valeur. Ce mode sera considéré uniquement si c'est un mode plus lent que le mode d'origine. Cette condition est indispensable sinon il y aurait toujours un recalage automatique vers le mode le plus rapide à la première extension de  $T_{SEC}$  dans chaque nouvel état. Le mode le plus lent est celui dont la distribution conditionnelle a l'espérance la plus grande. Il faut de plus s'assurer que la détection du mode le plus probable ne se fait pas sur des probabilité nulles, car dans ce cas le recalage est de toute façon faux (L15-L16).

L'estimation de la quantité  $T_{SEC}^{(op)}$  réelle (L29) se fait en fonction de la distribution globale du mélange à l'état  $x^{(op)}$  et non pas selon une loi conditionnelle. En effet, cette estimation essaye de calculer le temps de séjour réel passé dans l'état précédent. Par conséquent, aucune hypothèse ne doit donc être faite concernant le mode de dégradation réel actif. Ce mode sera justement estimé à l'étape suivante en fonction de ce temps de séjour.

La dernière modification se trouve dans la phase finale concernant le calcul de  $R^{(t)}$ .

Une vérification est effectuée pour savoir si le mode actif n'a pas changé par rapport à l'itération précédente. Si c'est le cas, alors il faut recalculer tous les temps de séjour dans les états suivants, conditionnellement à ce nouveau mode actif (L42)

Remarquons enfin que le cas relatif à un saut de plusieurs états n'apparaît pas dans cet algorithme. En effet, ce type de situation est extrême et fortement improbable en pratique. Pour des questions de simplicité, il n'a donc pas été considéré dans cette nouvelle version de l'algorithme.

### *Remarques et observations*

L'algorithme est capable de détecter un changement de mode, en fonction des diagnostics disponibles et de se recalculer en conséquence même si cette hypothèse n'est pas faite lors de l'apprentissage. Il est rappelé qu'un mode de dégradation tel que nous le considérons est constitué par une relation de dépendance conditionnelle entre les temps de séjour du système dans ses différents états.

Le recalage de mode ne se fait que lorsqu'il y a un changement d'état en fonction du temps passé dans l'état précédent ou lorsque qu'il y a une extension progressive du temps de séjour. Ce recalage se fait en prenant le mode le plus probable compte tenu des temps de séjour observés. Il est possible de remplacer ce choix par un tirage aléatoire pondéré par les probabilités a posteriori du mélange de l'état considéré. C'est ce second procédé qui sera utilisé dans la représentation graphique de l'algorithme de pronostic.

La méthodologie qui vient d'être présentée repose sur l'existence d'un modèle de mélange visible pour identifier l'existence de modes de dégradation concurrents. Précisons tout de même que le concept de mode de dégradation n'est qu'une **interprétation** possible de l'existence de relations de dépendance conditionnelles entre les LTS des différents états. D'un point de vue pratique, il est possible de construire des bases d'observations contenant de fortes dépendances entre les états sans que pour autant il n'y ait de modèle de mélanges ( et donc de modes de dégradation ) visibles. Un exemple simple de construction d'une telle base est donné ci-dessous. Considérons un système à deux états pour lequel une base de 60 observations doit être créée. Le problème est donc de construire, en reprenant les notations déjà utilisées, l'ensemble  $(y_i^{(l)})_{i=1,2}^{l=1,\dots,60}$ . C'est à dire, l'ensemble des temps de séjour observés. Cette base est générée à l'aide de distributions uniformes de la façon suivante :

$$\forall l, y_1^{(l)} \sim U([1, 50]) \quad (3.9)$$

$$\forall l, y_2^{(l)} \sim \begin{cases} U([1, 10]) & \text{si } y_1^{(l)} \leq 25 \\ U([10, 20]) & \text{si } y_1^{(l)} > 25 \end{cases} \quad (3.10)$$

Un apprentissage classique par la méthode MV conduirait à obtenir une distribution uniforme dans chaque état et donc aucun mélange visible. Pourtant la base contient une forte dépendance, et chercher l'apprentissage de lois conditionnelles permettrait de réduire considérablement la variance des lois considérées à l'état 2. Pourtant, la méthodologie que nous avons développée ne serait pas capable de le faire dans ce cas là. Avec cet exemple théorique simple, on veut donc insister sur le fait que l'existence de relations de dépendance conditionnelle entre les temps de séjour est donc a priori complètement indépendante de l'existence d'un quelconque mélange visible qui est lui fonction de la répartition de la densité des temps de séjour à l'intérieur de chaque état. Si nous avons postulé l'existence de mélanges, c'est simplement pour tenter de trouver une interprétation pratique à l'existence de ce type de phénomènes de dépendance s'exerçant à travers les états. Pour le dire d'une autre façon, l'existence de tels processus de dépendance en pratique serait liée à l'existence de dynamiques précises et déterminées qui entraîneraient de fortes concentrations visibles de temps de séjour à l'état initial autour de certaines valeurs caractéristiques de ces dynamiques particulières. Pour des situations similaires à celles que nous venons de construire, même si il est peut être improbable en pratique d'en observer, une procédure d'analyse plus générale, de découpage de l'espace d'état initial, qui ne soit pas reliée à l'existence d'un mélange pourrait être imaginée.

### 3.3.5 Récapitulatif

Le MGDC qui vient d'être présenté est un outil exploratoire permettant de rechercher l'existence éventuelle de modes de dégradation en compétition. Il s'agit donc d'un processus de recherche d'une information latente via un MGD spécifique destiné à l'apprentissage de lois conditionnelles présentant in fine une plus faible variance que les LTS obtenues par simple MGD. Si de telles lois existent, alors le modèle est très utile pour accélérer la convergence de la RUL estimée vers la RUL réelle. Si les lois obtenues ont une variance identique à celles apprises par le MGD, alors cela signifie que la connaissance des modes n'apporte aucune information additionnelle sur la vitesse de dégradation et que par conséquent, il est plus efficace de garder le simple MGD.

L'utilisation d'un MGDC peut être intéressant lorsque :

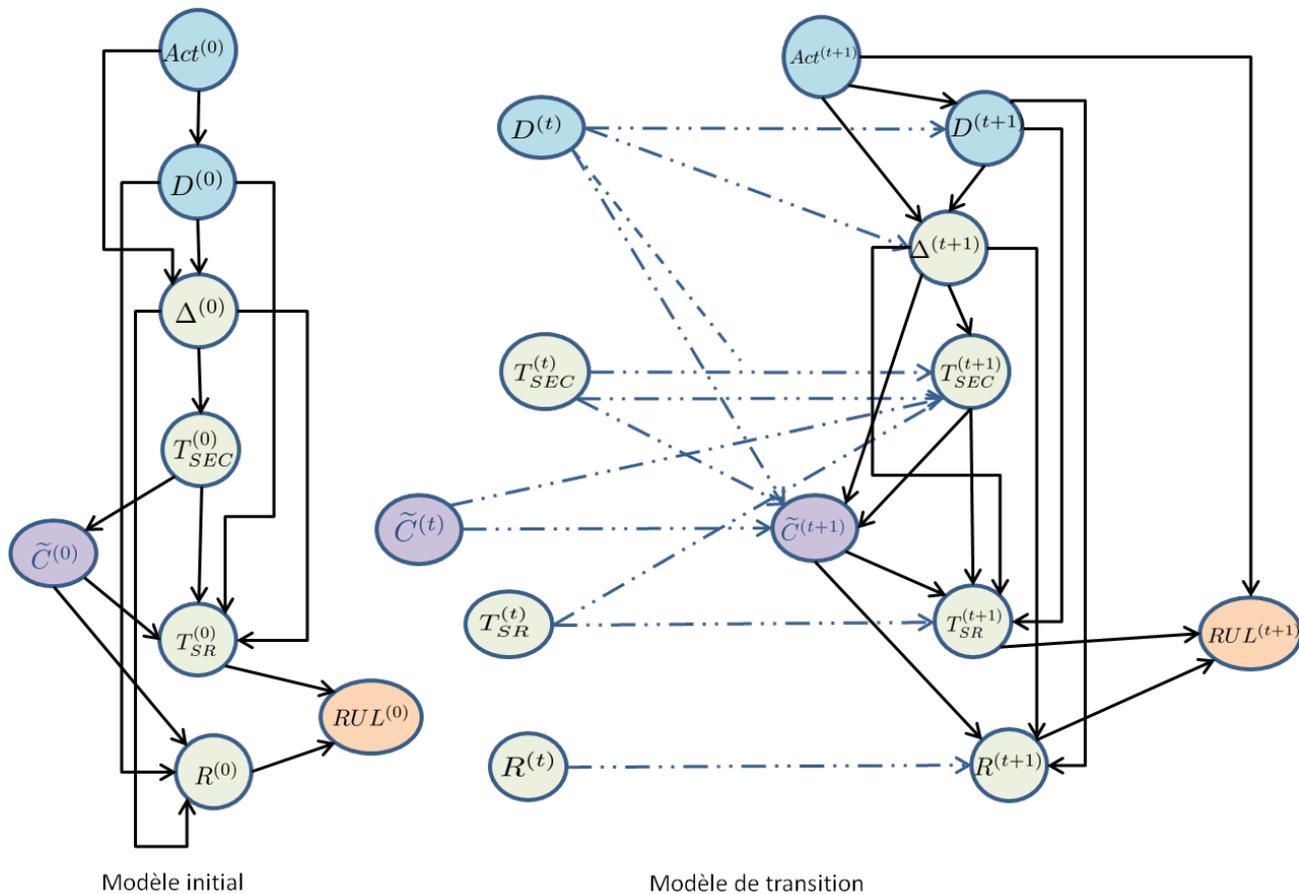
- On dispose d'une base d'observations hétérogène rassemblant des informations de dynamiques différentes, soit par ce qu'elles sont relatives à des composants de fabrication différente, soit parce que les contextes d'utilisation sont différents.
- On effectue une analyse de la dégradation d'un système ayant plusieurs composants mais dont le diagnostic d'état ne peut se faire qu'à l'échelle du système. Autrement dit, les états de dégradation définis sont ceux à l'échelle du système global. La dégradation visible est alors conduite par la dégradation d'un ou de plusieurs sous-composants du système dont l'évolution de leur état propre est parfois difficilement accessible. Les temps de séjour dans chaque état à l'échelle du système sont donc un produit de plusieurs modes de dégradation correspondant chacun à ceux de l'un des sous-composants du système.
- Les états de dégradation d'un composant sont le résultat de plusieurs modes de dégradation concurrents ayant leur dynamique propres et dont les effets peuvent se masquer mutuellement ( par exemple la dégradation d'un axe de transmission mécanique en fatigue, en usure, et en corrosion).

### 3.4 Représentation de l'algorithme de pronostic complet par RBD

Cette section a pour but de donner une représentation de l'algorithme de pronostic en tant que réseau bayésien dynamique (RBD). Celle-ci est nécessaire pour permettre l'introduction du concept de pronostic dans le modèle VirMaLab avec un formalisme homogène. Le processus de création de ce modèle est expliqué en détail. Pour construire un tel réseau, un nœud est associé à chaque variable active de l'algorithme. Par conséquent les variables  $T_{SEC}^{(t)}$ ,  $T_{SR}^{(t)}$  et  $R^{(t)}$  sont représentées explicitement. Pour distinguer les cas où il y a ou non changement d'état, il est nécessaire d'introduire une variable qui contrôle ces différents cas de figure. Celle-ci est notée  $\Delta^{(t)}$ . Enfin pour tenir compte du cas du MGDC intégrant les lois conditionnelles, il faut introduire une variable représentant le mode actif considéré à l'instant  $t$ . Il s'agit du nœud  $\tilde{C}$ . Ce nœud ne doit pas être confondu avec le nœud  $C^{(t)}$  défini dans la partie précédente. En effet, même si ces deux nœuds représentent des quantités de nature identiques, ils font référence à des calculs différents.  $C^{(t)}$  est le mode actif du modèle de dégradation tandis que  $\tilde{C}^{(t)}$  est le mode actif détecté à partir des diagnostics disponibles et utilisé pour faire du pronostic. La version graphique modélise le choix des temps de séjour, tout comme les mises à jour des modes par tirages aléatoires. Il est possible de représenter la variante par tirage de moyenne en modifiant légèrement les lois de transition de certains nœuds.

Enfin, la RUL est représentée par un nœud de sortie du modèle qui a vocation, à terme, à être connecté à un nœud de décision de maintenance.

Le modèle ainsi créé est représenté sur la figure 3.13.



**FIGURE 3.13:** RBD représentant l'algorithme de pronostic

L'ensemble des nœuds liés à l'algorithme de pronostic sont représentés en vert. Les nœuds liés à l'utilisation du MGDC sont représentées en violet. Les données d'entrée  $D^{(t)}$  et  $Act^{(t)}$  représentés en bleu décrivent respectivement le résultat de diagnostic et le nœud d'activation déterminant si le diagnostic est ou pas activé à l'instant  $t$ .

Les lois de probabilités conditionnelles (LPC) de chacun de ces nœuds codent les relations qui existent entre ces différentes variables. L'ensemble des opérations de l'algorithme de

pronostic (AP) sont des opérations algébriques déterministes ou semi déterministes. Les LPC des nœuds en verts sont par conséquent représentées par des potentiels creux <sup>6</sup>.

### 3.4.1 Définition des nœuds

Cette section définit formellement de façon précise les domaines et les lois conditionnelles de chacun des nœuds énoncés précédemment. Les lois qui seront décrites ci-dessous ne sont qu'une réécriture des différentes lignes de l'algorithme de pronostic en terme de lois probabilistes. La différence tient aux contraintes liées à l'utilisation d'un réseau bayésien. La notation  $D_X$  désignera le domaine du nœud X. On note par ailleurs  $M_X$  le potentiel associé au nœud X. Les ' dans les expressions  $M(x, y', z')$ , désignerons des variables de la tranches de temps précédente.  $M^{init}$  désigne le potentiel de la loi initiale et  $M^{\rightarrow}$  celui de la loi de transition. La notation  $\delta_x$  représente la distribution de Dirac en x, c'est à dire  $\forall k, \delta_x(k) = 1, si k = x$  et 0 sinon. Cette notation est prolongée au cas multidimensionnel en supposant que la masse de Dirac s'applique en fonction de la première dimension notée en gras. Donc,  $M(\mathbf{a}, b, c, d) = \delta_x$  signifie  $\forall b, c, d, M(\mathbf{a}, b, c, d) = 1$  si  $a = x$  et 0 sinon.

Il ne faudra pas confondre les notations  $\Delta^{(t)}$  et  $\Delta t$ . La première correspond à la valeur d'un des nœuds du RBD pronostic, tandis que la seconde désigne le pas de temps qu'il y a entre deux diagnostics. Le fait que la valeur  $\Delta t$  apparaisse dans les lois qui vont suivre suppose implicitement que sa valeur est fixe. La représentation suppose donc que le pas de diagnostic est constant. Changer le pas de diagnostic suppose de changer les lois des nœuds. Il est possible de changer la loi des nœuds dans le codage au cours de l'inférence mais cela suppose de toute façon que l'évolution des pas de diagnostics soit définie à l'avance. Un pas de diagnostic dynamique en fonction de l'état observé n'est donc pas modélisable avec ce RBD et supposerait probablement l'ajout de nouvelles variables.

#### Noeud Act

Le nœud *Act* contrôle l'activation du diagnostic. Il peut prendre deux valeurs : 1 si le diagnostic est désactivé, 2 si celui-ci est activé. Il ne possède aucun parent. Sa loi est donc un potentiel qui est en fait un vecteur à 1 dimension. Les instants d'activation sont déterminés à l'avance. Sa distribution est donc déterministe et  $M_{Act} = [0, 1]$  ou  $[1, 0]$ . Cette loi est modifiée directement lors de la procédure d'inférence aux instants où son

---

6. C'est à dire des matrices multidimensionnelles contenant une proportion importante de valeurs nulles

activation est prévue. Soit  $T_{diag}$  : l'ensemble des instants où il est prévu que le diagnostic soit activé.

$$M_{Act}^{(t)} = \begin{cases} [0, 1] & \text{si } t \in T_{diag} \\ (1, 0] & \text{sinon} \end{cases} \quad (3.11)$$

### Noeud D

Le noeud  $D$  représente le résultat du processus de diagnostic. C'est une donnée d'entrée de l'algorithme de pronostic. Celle-ci dépend de l'état du système à l'instant  $t$  c'est à dire du noeud  $X^{(t)}$  du modèle de dégradation. Son domaine <sup>7</sup> est de taille  $N+1$ . Il contient l'ensemble des états de  $X^{(t)}$ , plus un état  $\emptyset$  correspondant au cas où le diagnostic est désactivé.  $D_D = \{1 (\emptyset), 2 (x_1), 3 (x_2), \dots, n+1 (x_P)\}$ . La LPC de ce noeud ne dépend que de la valeur de  $X$  et de  $Act$  au même instant. Par conséquent, la loi initiale est la même que la loi de transition et l'on a donc  $M_D^{\rightarrow} = M_D^{init}$  qui est un potentiel de dimension  $(N+1) * 2 * N$ . Si le diagnostic est désactivé ( $Act=1$ ) alors  $D$  est désactivé et vaut 1 ( ce qui correspond à  $\emptyset$ ). Dans le cas contraire, la valeur de  $D$  est liée à celle de  $X$  par une matrice de confusion notée  $D_\lambda$ .

$$M_D^{\rightarrow} = P(D^{(t)}|X^{(t)}, Act^{(t)}) = M_D(\mathbf{d}, x, a) \begin{cases} \delta_1 & \text{si } a = 1 \\ D_\lambda(d, x) & \text{sinon} \end{cases} \quad (3.12)$$

Le choix de la matrice  $D_\lambda(d, x)$  dépend de la façon dont on modélise le processus de diagnostic. Une matrice identité décrit un système de diagnostic parfait, c'est à dire  $D^{(t)}$  toujours égal à  $X^{(t)}$ . Une matrice non identité permettra de modéliser le risque de fausse alarme et d'erreurs possible de diagnostic.

### Noeud $\Delta$

Le noeud  $\Delta^{(t)}$  a pour but de contrôler les différentes situations possibles de changements d'état. Il possède  $n+1$  états et est à valeur dans  $D_\Delta = \{1 (\emptyset), 2 (\text{pas de transition}), 3 (1 \text{ saut d'état}), \dots, n (n-1 \text{ sauts d'états})\}$ .  $\emptyset$  désigne le cas où le diagnostic est désactivé. Cet état  $\emptyset$  existe pour répercuter l'effet de la désactivation sur les autres noeuds du réseau pronostic.  $\Delta^{(0)}$  est toujours égal à 2. A la première itération, on considère qu'il n'y a pas

7. On rappelle que  $N$  est le nombre d'états que peut prendre le composant étudié

eu de transition soit  $M_{\Delta}^{init} = \delta_2$ . La loi de transition est définie comme suit :

$$P(\Delta^{(t+1)} | D^{(t+1)}, Act^{(t+1)}, D^{(t)}) = M_{\Delta}^{\rightarrow}(\mathbf{D}, d, a, d') = \begin{cases} \delta_1 & \text{si } a = 1 \\ \delta_{d-d'+2} & \text{si } a = 2 \end{cases} \quad (3.13)$$

### Nœud $T_{SEC}$

Le nœud  $T_{SEC}$  représente le temps déjà écoulé dans l'état courant. Celui-ci a pour domaine l'ensemble  $\{0, \dots, T_{max}\}$  de taille  $T_{max} + 1$ . La loi initiale du nœud  $T_{SEC}$  est une distribution déterministe dont la masse est réglée par l'utilisateur et égale à 1 sur la valeur initiale choisie. Ceci s'écrit donc formellement

$$P(T_{SEC}^{(0)} | \Delta^{(0)}) = M_{SEC}^{init}(\mathbf{sc}, D) = \delta_{vu} \quad (3.14)$$

où  $vu$  est la valeur initiale réglée par l'utilisateur. Même si la valeur de  $\Delta^{(0)}$  n'intervient pas dans la loi conditionnelle initiale du nœud, le modèle graphique représente une dépendance entre  $\Delta^{(0)}$  et  $T_{SEC}^{(0)}$ . Cela est lié à une volonté de garder le même graphe de relations entre les variables d'une même tranche de temps quelque soit l'instant considéré.

Cette dépendance entre  $\Delta^{(t)}$  et  $T_{SEC}^{(t)}$  sera effectivement présente à partir de  $t = 1$ .

La loi de transition est semi-déterministe et définie par morceau en fonction de la valeur du parent  $\Delta^{(t)}$ . Comme le domaine prend des valeurs commençant à 0, les index de position sont décalés de 1 unité. La notation  $\Psi_{d,c}^{[sc'+1, sc'+\Delta t]}$  désignera la restriction normalisée de la loi conditionnelle sur l'état  $d$  sachant le mode  $c$  sur l'intervalle  $[sc' + 1, +sc' + \Delta t]$ . La loi de transition est définie de la façon suivante.

$$P(T_{SEC}^{(t+1)} | \Delta^{(t+1)}, D^{(t+1)}, D^{(t)}, \tilde{C}^{(t)}, T_{SEC}^{(t)}, T_{SR}^{(t)}) = M_{SEC}^{\rightarrow}(\mathbf{sc}, D, d, d', c', sc', sr') \\ = \begin{cases} \delta_{sc'} & \text{si } D = 1 \\ \delta_{sc'+\Delta t} & \text{si } D = 2 \\ \delta_{\Delta t - sr'} & \text{si } D = 3 \text{ et } sr' \leq \Delta t \\ \Psi_{d'}^{[sc'+1, sc'+\Delta t]}(\delta t - sc + sc') & \text{si } D = 3 \text{ et } sr' > \Delta t \end{cases} \quad (3.15)$$

Si l'on avait voulu représenter la méthode de tirage de  $T_{SEC}$  par moyenne, il aurait fallu remplacer la ligne  $\Psi_{d'}^{[sc'+1, sc'+\Delta t]}(x)$  par une fonction de Dirac au point  $E(\Psi_{d'}^{[sc'+1, sc'+\Delta t]})$  où  $E$  désigne l'espérance de la distribution. Pour rappel, les cas où  $D > 3$  (plusieurs états manqués) ne sont pas pris en compte dans la version LTSC de l'algorithme de pronostic.

### Nœud $\tilde{C}$

Le nœud  $\tilde{C}$  représente le mode actif à l'instant courant. A l'instant initial cette valeur est calculée à partir du premier temps de séjour tiré dans l'état initial. Celui-ci est impacté par le choix de  $T_{SEC}^{(0)}$  de l'utilisateur. Le domaine de  $\tilde{C}$  est  $D_{\tilde{C}} = \{1, \dots, n_M\}$  de taille  $n_M$  où  $n_M$  est le nombre de modes détectés à l'apprentissage. Formellement :

$$P(\tilde{C}^{(0)} | T_{SEC}^{(0)}, \Delta^{(0)}, D^{(0)}) = M_{\tilde{C}}^{init}(\mathbf{c}, sc, D, d) = \sum_{s=sc}^{T_{max}} \Psi_{d/c}(s) \quad (3.16)$$

La notation précédente fait intervenir les probabilités a posteriori de la restriction de la loi du mélange de l'état d sur l'intervalle  $[T_{SEC}^{(0)}, T_{max}]$ . Tout comme dans la version algébrique, la loi de transition dépend de la situation de changement d'état actif, c'est à dire de la valeur de  $\Delta^{(t)}$ . Elle est définie de la façon suivante :

$$P(\tilde{C}^{(t+1)} | T_{SEC}^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t)}, T_{SEC}^{(t)}, D^{(t)}) = M_{\tilde{C}}^{\rightarrow}(\mathbf{c}, sc, D, c', sc', d') = \begin{cases} \Psi_{d',c}(sc' + \Delta t - sc) & \text{si } D = 3 \\ \delta_{c'} & \text{si } D \neq 3 \end{cases} \quad (3.17)$$

Les formules précédentes 3.16 et 3.17 modélisent le cas où le mode actif est choisi en fonction de la probabilité par un tirage aléatoire pondéré par les probabilités a posteriori et non en prenant le mode le plus probable.

### Nœud $T_{SR}$

La quantité  $T_{SR}$  correspond au temps restant estimé dans l'état courant. Celui-ci a un domaine  $D_{SR} = \{0, \dots, T_{max}\}$  de taille  $T_{max} + 1$  pour les mêmes raisons que pour le nœud  $T_{SEC}$ . Concernant la loi initiale, comme  $\Delta^{(0)}$  a une valeur déterministe réglée sur 2, il n'est pas nécessaire de définir la loi de  $T_{SR}$  pour d'autres valeurs puisque ces expressions ne seront pas prises en compte dans les calculs.

$$P(T_{SR}^{(0)} | \tilde{C}^{(0)}, T_{SEC}^{(0)}, \Delta^{(0)}, D^{(0)}) = M_{SR}^{init}(\mathbf{sr}, c, sc, D, d) = \Psi_{d'/c'}^{[sc+1, T_{max}]}(sr + sc)$$

La loi de transition de  $T_{SR}$ , tout comme dans la version algébrique est définie uniquement dans le cas d'un non changement d'état, ou d'un changement vers l'état suivant.

$$P(T_{SR}^{(t+1)} | \tilde{C}^{(t+1)}, T_{SEC}^{(t+1)}, \Delta^{(t+1)}, D^{(t+1)}, T_{SR}^{(t)}) = M_{SR}^{\rightarrow}(\mathbf{sr}, c, sc, D, d, sr') =$$

$$= \begin{cases} \delta_{sr'} & \text{si } D = 1 \\ \delta_{sr' - \Delta t} & \text{si } D = 2 \text{ et } sr' > \Delta t \\ \Psi_{d/c}^{[sc+1, T_{max}]}(sr + sc) & \text{si } D = 2 \text{ et } sr' \leq \Delta t \\ \delta_{\Delta t - sr'} & \text{si } D = 3 \text{ et } sr' \leq \Delta t \\ \Psi_{d', c'}^{[sc'+1, sc'+\Delta t]}(\Delta t - sc + sc') & \text{si } D = 3 \text{ et } sr' > \Delta t \end{cases} \quad (3.18)$$

### Nœud R

Le nœud R représente la somme de l'ensemble des temps de séjour élémentaires dans les états suivants l'état courant jusqu'à la panne. Les temps de séjour ne sont pas explicitement représentés par un nœud. Cela amène à manipuler constamment des sommes de lois temps de séjours discrète. Le calcul de la loi de la somme de plusieurs variables aléatoires s'effectue grâce au **produit de convolution**.

Soit 2 variables aléatoires discrètes bornées  $X \in D_X = \{0, \dots, T_X\}$  et  $Y \in D_Y = \{0, \dots, T_Y\}$  de distribution  $f_X$  et  $f_Y$ . Soit  $Z = X + Y \in D_Z = \{0, \dots, T_X + T_Y\}$  suivant la distribution  $f_Z$ . Alors :

$$\forall z \in D_Z, \quad P(Z = z) = \sum_{\substack{(x,y) \in D_X \times D_Y \\ x+y=z}} P(X = x)P(Y = y) \quad (3.19)$$

ce qui, de façon équivalente, peut s'écrire sous la forme d'un produit de convolution :

$$f_Z(z) = \sum_{s=0}^{T_Z} f_X(z - s)f_Y(s) = f_X * f_Y \quad (3.20)$$

D'après les hypothèses,  $\Psi_X$  et  $\Psi_Y$  ne sont pas définies jusqu'à  $T_Z$ . On supposera dans la formule précédente que la fonction vaut 0 en dehors de son domaine initial.

De façon similaire, pour n variables  $X_1, \dots, X_n$  et  $Z_n = \sum_{i=1}^n X_i$  alors  $f_{Z_n} = f_{X_1} * \dots * f_{X_n}$

Soit  $\alpha_i$ , la borne supérieure du support<sup>8</sup> de  $\Psi_i$ . Le domaine du nœud  $R$  est  $D_R \in \{0, \dots, \sum \alpha_i\}$ . La valeur initiale  $R^{(0)}$  représente la somme des estimations de temps de séjour de l'état initial jusqu'à la panne. Sa distribution est donc d'après le paragraphe précédent une fonction notée  $\Psi_R^{init}$  et l'on a :

$$\Psi_{R,d}^{init} = \Psi_{d*} * \dots * \Psi_{X_{P-1}} \quad (3.21)$$

avec  $d = D^{(0)}$ , et donc :

$$P(R^{(0)} | D^{(0)}) = M_R^{init}(r, d) = \Psi_{R,d}^{init}(r) \quad (3.22)$$

On notera désormais  $\Psi_{j:k} = \Psi_j * \dots * \Psi_k$  la loi de la somme des temps de séjour de l'état  $j$  à l'état  $k$ .

La loi de transition de  $R^{(t)}$  dépend de la variable de changement d'état  $\Delta^{(t)}$ . Si le diagnostic est désactivé ou qu'il n'y a pas de changement d'état,  $R^{(t)}$  est stationnaire. Lorsqu'il y a un changement d'état vers l'état suivant, il faut calculer  $P(R^{(t)} | R^{(t-1)}, D^{(t)}, \Delta^{(t)})$ . Si un changement d'état a lieu de l'état  $j$  à l'état  $j+1$ , cela équivaut à calculer  $\forall k, n$  :

$$\begin{aligned} P(S_{j+2} + \dots + S_{x_{P-1}} = k | S_{j+1} + \dots + S_{x_{P-1}} = n) &= \frac{P(S_{j+1} = n - k) P(S_{j+2} + \dots + S_{x_{P-1}} = k)}{P(S_{j+1} + \dots + S_{x_{P-1}} = n)} \\ &= \frac{\Psi_{j+1}(n - k) \Psi_{j+2:x_{P-1}}(k)}{\Psi_{j+1:x_{P-1}}(n)} \end{aligned} \quad (3.23)$$

Les fonctions  $\Psi$  sont connues dès le départ puisqu'elles ne dépendent que des modèles de mélange qui ont été appris. La formule générale est donc :

$$M_R^{\rightarrow}(r, D, d, r') = \begin{cases} \delta_{r'} & \text{si } D \leq 2 \\ \frac{\Psi_{d+2}(r'-r) \Psi_{d+3:x_{P-1}}(r)}{\Psi_{d+2:x_{P-1}}(r')} & \text{si } D = 3 \end{cases} \quad (3.24)$$

Dans la formule précédente, le  $d+2$  est lié au fait que l'on considère la valeur numérique exacte de l'état. Or, les indices sont décalés de 1 dans  $D^{(t)}$  à cause de l'existence d'un état de désactivation.<sup>9</sup>

8. Le support d'une fonction est l'ensemble des valeurs pour lesquelles la fonction est non nulle

9.  $D^{(t)} = 1$  ne désigne pas l'état 1 mais l'état de désactivation.

### Nœud $RUL$

a loi de la  $RUL$  ne dépend que de la somme algébrique des valeurs de  $T_{SR}$  et  $R$  au même instant. Par conséquent, il n'y a pas de différence entre la loi initiale et la loi de transition. Celle-ci est de fait déterministe.

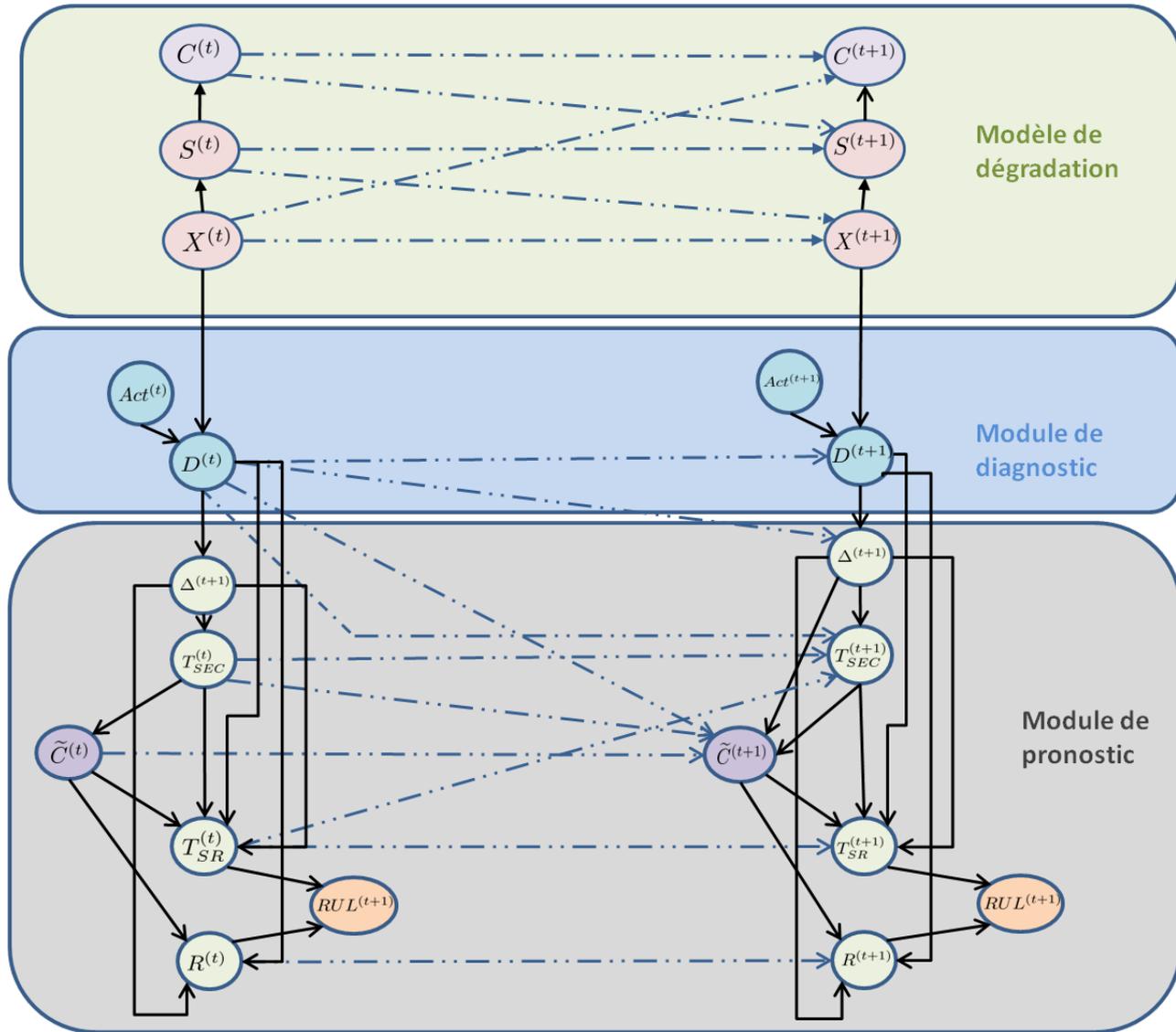
$$P(RUL^{(t)} | T_{SR}^{(t)}, R^{(t)}) = M_{RUL}(\mathbf{ru}, sr, r) = \delta_{sr+r} \quad (3.25)$$

En conclusion, la représentation graphique de l'algorithme de pronostic permet de fournir les mêmes résultats que la version algébrique de l'algorithme de pronostic tout en gardant une vision purement probabiliste en terme de calcul ce qui permet de fournir les distributions associées aux diverses quantités calculées et non pas de simples valeurs numériques. Pour le dire autrement, **la version algébrique n'est qu'un processus d'échantillonnage de la version graphique**. Elle est très simple à programmer et permet de fournir en pratique des estimations de la  $RUL$  rapidement. La version graphique, quant à elle, permet de donner des intervalles de confiance à ces estimations. Mais elle a surtout vocation à représenter le processus de décision basé sur cet algorithme de pronostic de façon à pouvoir évaluer et optimiser une politique de maintenance complète de nature prévisionnelle. Enfin, précisons que la représentation donnée est celle de la version de l'algorithme par LTSC. La version initiale basée sur un simple GDM s'en déduit facilement en supprimant les deux nœuds  $C$  et  $\tilde{C}$  du graphe et en modifiant logiquement, dans la définition de certaines lois, les  $\Psi_{x/m}$  par des  $\Psi_x$ .

## 3.5 Problématique de l'inférence

Cette section s'intéresse au développement de méthodes d'inférence exacte. Comme il a été mentionné dans le chapitre précédent, la question principale dans l'utilisation de l'algorithme d'élimination des variables est celui du choix d'un ordre d'élimination optimal. Ce choix est déterminant dans la complexité des calculs qui vont s'effectuer.

Le RBD défini et qui englobe à la fois le modèle de dégradation et le modèle de pronostic, contient de nombreux nœuds représentant des valeurs quantitatives. Par conséquent, la taille de leur domaine est très grande. il est donc nécessaire de développer des méthodes spécifiques au graphe de ce RBD pour conduire les calculs d'inférence avec une complexité acceptable. L'ensemble du graphe est repris sur la figure 3.14. L'interface gauche



**FIGURE 3.14:** RBD global représentant l'ensemble du réseau intervenant dans le calcul de la RUL

de ce réseau se compose des 9 nœuds suivants :

$$I_g = (C, S, X, D, T_{SEC}, T_{SR}, R, \tilde{C}) \quad (3.26)$$

Avec,  $n_M$  le nombre de modes détectés,  $N$  le nombre d'états du composant et  $T_{max}$  le temps maximal pendant lequel il est possible de rester sur un état.

La complexité spatiale d'un tel potentiel est

$$|I_g| = n_M^2 N^3 (T_{max})^4 \quad (3.27)$$

Si  $N=4$ ,  $n_M = 2$ ,  $T_{max} = 60$  ainsi que  $n_A=3$ , la complexité de l'interface est déjà :  $|I_g| \sim 6 \times 10^9$  soit environ 6 milliards de valeurs. Une telle complexité est trop importante en pratique pour être manipulée aisément sur un ordinateur standard. De plus, un calcul direct par l'algorithme d'interface ferait intervenir  $p(I_g^{(t+1)} | I_g^{(t)})$  ce qui correspondrait à une complexité de  $|I_g|^2$ . Il n'est donc pas possible d'effectuer ce calcul directement par bloc pour mettre à jour la loi de l'interface gauche.

L'objectif est donc ici de proposer une procédure d'inférence de la loi marginale du nœud  $RUL^{(t)}$ , spécifique au réseau, et exploitant au mieux les relations d'indépendances conditionnelles (RIC) existantes de façon à limiter au maximum la complexité des calculs qui vont s'effectuer sans faire aucune hypothèse particulière en plus de celles qui sont déjà contenues dans le modèle de la figure 3.14.

Sur le graphe de cette figure, les deux flèches reliant  $X^{(t)}$  à  $D^{(t)}$  et  $X^{(t+1)}$  à  $D^{(t+1)}$  séparent le réseau en deux blocs presque indépendants. L'idée est donc de mettre à jour le bloc  $(X^{(t)}, S^{(t)}, C^{(t)})$  presque indépendamment des autres nœuds.

Il s'agit donc de découper astucieusement l'interface gauche en 3 sous blocs qui pourront être mis à jour de façon semi-indépendante. Les raisons de ce découpage sont expliquées dans la section suivante. 3 groupes de nœuds sont distingués :

- groupe 1 :  $(X, S)$
- groupe 2 :  $(D, \tilde{C}, T_{SEC}, T_{SR})$
- groupe 3 :  $(D, R, \tilde{C})$

La notation  $n(X)$  désignera le potentiel du nœud  $X$  plutôt que de réécrire l'ensemble de la loi conditionnelle du nœud avec l'ensemble de ses parents. La notation  $\times$  désignera quant à elle l'opération de produit de potentiels,  $\overset{\Sigma}{\rightarrow}$ , le résultat à l'issue d'une opération de marginalisation de la loi et  $\overset{cond}{\rightarrow}$  le résultat à la suite d'une opération de conditionnement. La procédure d'inférence pour la mise à jour de la RUL et de chacun des 3 blocs est détaillée dans l'algorithme 3.

---

**Algorithme 3** Procédure d'inférence exacte spécifique au RBD pronostic :

---

**Tranche initiale à t=0 :**

- 1:  $n(X^{(0)}) \times n(S^{(0)}) \times n(C^{(0)}) = \mathbf{p}(\mathbf{X}^{(0)}, \mathbf{S}^{(0)}, \mathbf{C}^{(0)})$  (**Loi du Groupe 1**)
- 2:  $n(X^{(0)}) \times n(Act^{(0)}) \times n(D^{(0)}) \times n(\Delta^{(0)}) \xrightarrow{\Sigma} p(X^{(0)}, D^{(0)}, \Delta^{(0)})$
- 3:  $p(X^{(0)}, D^{(0)}, \Delta^{(0)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{D}^{(0)}, \mathbf{\Delta}^{(0)})$
- 4:  $p(X^{(0)}, D^{(0)}, \Delta^{(0)}) \xrightarrow{\Sigma} p(D^{(0)}, X^{(0)}) \xrightarrow{cond} \mathbf{p}(\mathbf{D}^{(0)} | \mathbf{X}^{(0)})$
- 5:  $\mathbf{p}(\mathbf{D}^{(0)}, \mathbf{\Delta}^{(0)}) \times n(T_{SEC}^{(0)}) \times n(\tilde{C}^{(0)}) \times n(T_{SR}^{(0)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{D}^{(0)}, \mathbf{T}_{SEC}^{(0)}, \tilde{\mathbf{C}}^{(0)}, \mathbf{T}_{SR}^{(0)})$  (**Loi du groupe 2**)
- 6:  $p(D^{(0)}, \Delta^{(0)}, T_{SEC}^{(0)}, \tilde{C}^{(0)}, T_{SR}^{(0)}) \xrightarrow{\Sigma, cond} p(T_{SR}^{(0)} | D^{(0)}, \Delta^{(0)}, \tilde{C}^{(0)})$
- 7:  $p(D^{(0)}, \Delta^{(0)}, T_{SEC}^{(0)}, \tilde{C}^{(0)}, T_{SR}^{(0)}) \xrightarrow{\Sigma} p(D^{(0)}, \Delta^{(0)}, \tilde{C}^{(0)})$
- 8:  $p(D^{(0)}, \Delta^{(0)}, \tilde{C}^{(0)}) \times n(R^{(0)}) \times p(T_{SR}^{(0)} | D^{(0)}, \Delta^{(0)}, \tilde{C}^{(0)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{D}^{(0)}, \tilde{\mathbf{C}}^{(0)}, \mathbf{R}^{(0)})$  (**Loi du groupe 3**) et  $p(T_{SR}^{(0)}, R^{(0)})$
- 9:  $p(T_{SR}^{(0)}, R^{(0)}) \times n(RUL^{(0)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{RUL}^{(0)})$

**Mise à jour des groupes de la tranche t à t+1**

- 1:  $p(X^{(t)}, S^{(t)}, C^{(t)}) \times n(X^{(t+1)}) \times n(S^{(t+1)}) \xrightarrow{\Sigma} p(X^{(t)}, C^{(t)}, X^{(t+1)}, S^{(t+1)})$
- 2:  $p(X^{(t)}, C^{(t)}, X^{(t+1)}, S^{(t+1)}) \times n(C^{(t+1)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{X}^{(t+1)}, \mathbf{S}^{(t+1)}, \mathbf{C}^{(t+1)})$   
(**Mise à jour du groupe 1**) et par marginalisation  $p(X^{(t)}, X^{(t+1)})$
- 3:  $p(X^{(t)}, X^{(t+1)}) \times \mathbf{p}(\mathbf{D}^{(t)} | \mathbf{X}^{(t)}) \times n(Act^{(t+1)}) \times n(D^{(t+1)}) \xrightarrow{\Sigma, cond} p(D^{(t+1)} | D^{(t)})$  et  $p(D^{(t+1)}, X^{(t+1)})$
- 4:  $X = \mathbf{p}(\text{groupe 2})^{(t)}$
- 5:  $X \times p(D^{(t+1)} | D^{(t)}) \times n(\Delta^{(t+1)}) = p(X, D^{(t+1)}, \Delta^{(t+1)})$
- 6:  $p(X, D^{(t+1)}, \Delta^{(t+1)}) \times n(T_{SEC}^{(t+1)}) \times n(\tilde{C}^{(t+1)}) = \mathbf{p}(\mathbf{X}, \mathbf{D}^{(t+1)}, \mathbf{\Delta}^{(t+1)}, \mathbf{T}_{SEC}^{(t+1)}, \tilde{\mathbf{C}}^{(t+1)})$  (Sommet en terme de complexité)
- 7:  $p(X, D^{(t+1)}, \Delta^{(t+1)}, T_{SEC}^{(t+1)}, \tilde{C}^{(t+1)}) \xrightarrow{\Sigma} p(T_{SR}^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, T_{SEC}^{(t+1)}, \tilde{C}^{(t+1)})$
- 8:  $p(T_{SR}^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, T_{SEC}^{(t+1)}, \tilde{C}^{(t+1)}) \times n(T_{SR}^{(t+1)}) \xrightarrow{\Sigma} p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}, T_{SR}^{(t+1)})$  et  $\mathbf{p}(\mathbf{D}^{(t+1)}, \mathbf{T}_{SEC}^{(t+1)}, \tilde{\mathbf{C}}^{(t+1)}, \mathbf{T}_{SR}^{(t+1)})$  (**Mise à jour du groupe 2**)
- 9:  $p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}, T_{SR}^{(t+1)}) \xrightarrow{cond} p(T_{SR}^{(t+1)} | D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)})$
- 10:  $\mathbf{p}(\mathbf{X}, \mathbf{D}^{(t+1)}, \mathbf{\Delta}^{(t+1)}, \mathbf{T}_{SEC}^{(t+1)}, \tilde{\mathbf{C}}^{(t+1)}) \xrightarrow{\Sigma} p(D^{(t)}, \tilde{C}^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}) \xrightarrow{cond} p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)} | D^{(t)}, \tilde{C}^{(t)})$
- 11:  $\mathbf{Y} = \mathbf{n}(\text{groupe 3})^{(t)}$
- 12:  $Y \times p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)} | D^{(t)}, \tilde{C}^{(t)}) = p(\tilde{C}^{(t)}, R^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)})$
- 13:  $p(\tilde{C}^{(t)}, R^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}) \times n(R^{(t+1)}) \xrightarrow{\Sigma} p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}, R^{(t+1)})$
- 14:  $p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}, R^{(t+1)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{D}^{(t+1)}, \tilde{\mathbf{C}}^{(t+1)}, \mathbf{R}^{(t+1)})$  (**Mise à jour du groupe 3**)
- 15:  $p(D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}, R^{(t+1)}) \times p(T_{SR}^{(t+1)} | D^{(t+1)}, \Delta^{(t+1)}, \tilde{C}^{(t+1)}) \xrightarrow{\Sigma} p(T_{SR}^{(t+1)}, R^{(t+1)})$
- 16:  $p(T_{SR}^{(t+1)}, R^{(t+1)}) \times n(RUL^{(t+1)}) \xrightarrow{\Sigma} \mathbf{p}(\mathbf{RUL}^{(t+1)})$

### 3.5.1 Explications et commentaires

La première idée pour réduire la complexité des calculs, est de ne pas stocker en un seul bloc l'interface gauche. Pour cela, l'interface intra-tranche qui existe entre les nœuds  $X$  et  $D$  est utilisée. En effet pour mettre à jour l'ensemble des nœuds de l'interface relatifs à l'algorithme de pronostic<sup>10</sup> à proprement parlé, il n'est pas nécessaire de connaître les lois de  $X, S$  et  $C$  mais simplement de connaître, à chaque instant, la loi conditionnelle de  $P(D^{(t+1)}|D^{(t)})$ . La loi de  $(X, S, C)$  va donc se mettre à jour indépendamment des autres nœuds, d'où la création du groupe 1.

De plus, la RUL dépend des nœuds  $T_{SR}$  et  $R$  tandis que le nœud  $T_{SR}$  dépend de  $T_{SEC}, \Delta$  et  $D$  de la tranche précédente ainsi que de la tranche courante. Il est donc nécessaire de considérer en bloc tous ces nœuds. Le groupe 2 est donc défini par :  $(D, T_{SEC}, \tilde{C}, T_{SR})$

Enfin, le nœud  $R^{(t)}$  possède un domaine ayant la plus grande taille. Il est faut le mettre à jour dans un bloc à part pour ne pas faire exploser la complexité spatiale des calculs liés au produit. Or  $R^{(t)}$  intervient dans le calcul de la RUL avec le nœud  $T_{SR}$ . Ces deux nœuds n'étant pas indépendants, il n'est pas possible de les mettre à jour indépendamment. Cependant, ils le deviennent conditionnellement à  $D$  et  $\tilde{C}$ . Le groupe 3 est donc défini par  $(D, \tilde{C}, R)$

Avec ce nouvel algorithme, la partie la plus critique en terme de complexité se trouve dans la mise à jour du groupe 2. Le plus gros potentiel manipulé étant  $\mathbf{W} = \mathbf{p}(\mathbf{D}^{(t)}, \mathbf{T}_{SEC}^{(t)}, \tilde{\mathbf{C}}^{(t)}, \mathbf{T}_{SR}^{(t)}, \mathbf{D}^{(t+1)}, \Delta^{(t+1)}, \mathbf{T}_{SEC}^{(t+1)}, \tilde{\mathbf{C}}^{(t+1)})$ . La complexité du calcul d'inférence, conduit avec la méthode ci-dessus, est donc de l'ordre de :

$$|W| = N^3 n_M^2 (T_{max})^3$$

Par rapport à une procédure standard de mise à jour de l'interface  $((n_M^2 N^3 (T_{max})^4)^2)$ , le gain est non négligeable et rend possible le lancement des calculs sur des machines standards. Cependant, le fait que la complexité du calcul soit liée à  $T_{max}^3$  rend la faisabilité de la méthode très dépendante de la borne max des lois de temps de séjour considérées. Par exemple, avec une machine de calcul disposant de 8 Go de mémoire RAM, il est nécessaire de considérer pour  $T_{max}$  une valeur qui soit inférieure ou égale à 65.

10. Soit l'ensemble des nœuds positionnés en dessous de  $D^{(t)}$  sur la figure 3.14

### 3.5.2 Calculs d'inférence pour les probabilités conditionnelles

Lorsque l'on effectue des calculs de pronostic, cela conduit en pratique à des problèmes de prédiction, c'est à dire des calculs de la forme  $P(X^{(t+h)}|D^{(1:t)})$ . Soit  $G$  un ensemble de nœuds dont la loi marginale  $P(G)$  est connue. Pour calculer  $P(G|D^{(1:t)} = D)$  où  $D = (d_1, d_2, \dots, d_t)$ , on a :

$$P(G|D^{(1:t)} = D) = \frac{P(G, D^{(1)} = d_1, \dots, D^{(t)} = d_t)}{P(D^{(1)} = d_1, \dots, D^{(t)} = d_t)} \quad (3.28)$$

Il n'est évidemment pas pensable de stocker le potentiel de l'ensemble de la loi jointe de  $D^{(1:t)}$  présente au numérateur ainsi qu'au dénominateur ; sa taille croit exponentiellement en fonction de la taille de la fenêtre d'observation. Il a donc été choisi en pratique d'introduire les variables de conditions ( $D^{(1:t)}$ ) de façon itérative à chaque mise à jour de la tranche  $t$  vers  $t+1$ . A chaque mise à jour de  $G^{(t)}$  vers  $G^{(t+1)}$ , le conditionnement  $D^{(t)} = d_t$  est introduit.<sup>11</sup> Ce processus est décrit en détail dans l'algorithme 4.

La notation  $K(:, :, :, :)$  désignera un potentiel à 4 dimension, et  $L = K(:, :, :, d)$  désignera le potentiel à 3 dimensions, la quatrième dimension de  $K$  étant fixée à  $d$ .

---

**Algorithme 4** Modification de la procédure d'inférence pour introduction d'un conditionnement itératif par  $D^{(t)}$  sur la loi jointe d'un ensemble de nœuds  $G$ .

---

Soit  $P(G^{(t)}|D^{(1)} = d_1, \dots, D^{(t-1)} = d_{t-1})$ . le but est de calculer  $P = P(G^{(t+1)}|D^{(1)} = d_1, \dots, D^{(t)} = d_t)$  c'est à dire introduire  $D^{(t)} = d$  dans le conditionnement.

- 1:  $P(G^{(t)}|D^{(1)} = d_1, \dots, D^{(t-1)} = d_{t-1}) \times n(G^{(t+1)}) \times P(D^{(t)}|G^{(t)}|D^{(1)} = d_1, \dots, D^{(t-1)} = d_{t-1}) \rightarrow P(G^{(t+1)}, D^{(t)}|D^{(1)} = d_1, \dots, D^{(t-1)} = d_{t-1}) = K(:, :, :)$
  - 2:  $P(G^{(t+1)}, D^{(t)}|D^{(1)} = d_1, \dots, D^{(t-1)} = d_{t-1}) \rightarrow P(D^{(t)}|D^{(1)} = d_1, \dots, D^{(t-1)} = d_{t-1}) = J(:)$
  - 3:  $S = \frac{K(:, d_t)}{J(d_t)} = \frac{P(G^{(t+1)}, D^{(t)}=d_t|D^{(1)}=d_1, \dots, D^{(t-1)}=d_{t-1})}{p(D^{(t)}=d_t|D^{(1)}=d_1, \dots, D^{(t-1)}=d_{t-1})}$  et donc  $S = G^{(t+1)}|D^{(1)} = d_1, \dots, D^{(t)} = d_t$
- 

De cette façon, chaque terme de l'ensemble de conditionnement est introduit de façon itérative sans augmenter la complexité des calculs. Ceci peut permettre de mettre à jour un certain nombre d'indicateurs probabilistes de façon dynamique en fonction de nouvelles données entrantes.

---

11. On passe donc par exemple de  $P(G^{(t)}|D^{(1:t-1)})$  à  $P(G^{(t+1)}|D^{(1:t-1)}, D^{(t)})$ .

### 3.5.3 Perspectives

La limitation liée à la complexité des calculs est contraignante. Plusieurs pistes peuvent être envisagées pour surmonter cette contrainte.

Une idée pourrait être de remplacer le nœud discret du modèle graphique de durée  $S^{(t)}$  par un nœud à domaine continu. La principale difficulté de cette approche est que, d'une part, le fait d'utiliser une loi continue paramétrée implique des contraintes sur la forme de la loi qui n'existait pas dans la version discrète. D'autre part, rendre le nœud  $S^{(t)}$  continu suppose, par propagation, de rendre continus l'ensemble des nœuds du réseau pronostic. Le choix de lois continues, pour des raisons de stabilité liées aux opérations de produit, pourraient contraindre à n'utiliser que des distributions de famille exponentielles. Enfin les méthodes d'inférence sur les lois continues sont beaucoup moins développées que dans le cas discret.

Le caractère défini par morceau de nombreuses lois de l'algorithme de pronostic pourrait conduire à des relations d'indépendance conditionnelle au niveau non pas du graphe mais des lois d'un point de vue de leur définition mathématique. Il pourrait être utile de s'intéresser à cet aspect là.

Une autre piste à envisager consiste à exploiter le caractère creux des potentiels manipulés. La question principale est de trouver un format de stockage pour les matrices multidimensionnelles creuses qui accélère les 2 opérations élémentaires des calcul d'inférence : la marginalisation et le produit de 2 potentiels. A notre connaissance, il n'existe pas de bibliothèques de calcul d'inférence opérant sur des variables sparse. La piste que nous avons commencé à explorer et qui nous paraissait la plus adaptée est l'utilisation d'un format de stockage dans lequel chaque colonne contiendrait la valeur non nulle ainsi que l'ensemble des coordonnées de la valeur indices dans chaque dimension. Un potentiel de dimension  $Q$  serait donc codé par une matrice de taille  $N_{val} \times (Q + 1)$  où  $N_{val}$  est le nombre de valeurs non nulles du potentiel. La définition d'un codage optimal pour les opérations de produit et de marginalisation dans ce format n'est pas évidente puisque l'utilisation du format creux transforme un problème de complexité spatiale en problème de complexité algorithmique.

Une autre perspective intéressante pourrait être de simplifier le graphe du RBD pronostic en supprimant certaines dépendances. Pour expliquer cela il est nécessaire de revenir à la façon dont l'algorithme de pronostic a été créé. Dans sa version initiale l'algorithme de pronostic calculait la RUL et la mettait à jour en procédant par une succession de tirages aléatoires sans aucun test de compatibilité. Il s'agissait donc de l'algorithme 1 sans les lignes 9 à 11, 17 à 19 ainsi que 22 à 24. L'inconvénient majeur était que la trajectoire de la RUL était très aléatoire et par conséquent la convergence de la RUL estimée vers la RUL

réelle fortement non monotone. Les tests de compatibilité (consistant à ne refaire de tirage aléatoire que lorsque cela est indispensable) ont donc été ajoutés dans un second temps pour réduire le coté aléatoire des trajectoires de RUL obtenues et se rapprocher d'une convergence monotone. L'algorithme de pronostic est supposé fournir au mainteneur une information lui permettant de définir ou non une action de maintenance prévisionnelle. Pour des raisons déjà évoquées, celui-ci fournit une instanciation de la RUL non une distribution. Il n'est pas pensable de définir une politique de maintenance à partir d'une valeur de RUL aléatoire changeant à chaque lancement de l'algorithme. Pour cette raison, nous avons remplacé les tirages aléatoires par la méthode variante reposant sur le choix des moyennes. Cela permet d'avoir en sortie une valeur fixe de référence de la RUL à partir de laquelle, on peut ensuite envisager une règle de décision.

Or, le fait de procéder par choix de la moyenne, ne conduit plus à un comportement aléatoire du calcul de la RUL. Par conséquent les tests de compatibilité ne sont plus forcément nécessaires. Or ceux-ci sont à l'origine des dépendances suivantes ( $T_{SR}^{(t)} \rightarrow T_{SR}^{(t+1)}$  et  $T_{SR}^{(t)} \rightarrow T_{SEC}^{(t+1)}$ ). La suppression de ces tests, et donc des flèches associées dans le graphe, pourrait simplifier grandement la complexité de la procédure d'inférence car cela réduirait alors la taille du bloc 2 et simplifierait sa procédure de mise à jour déjà identifiée comme la partie de la procédure d'inférence la plus complexe. Le plus gros potentiel manipulé ne serait alors que de taille :

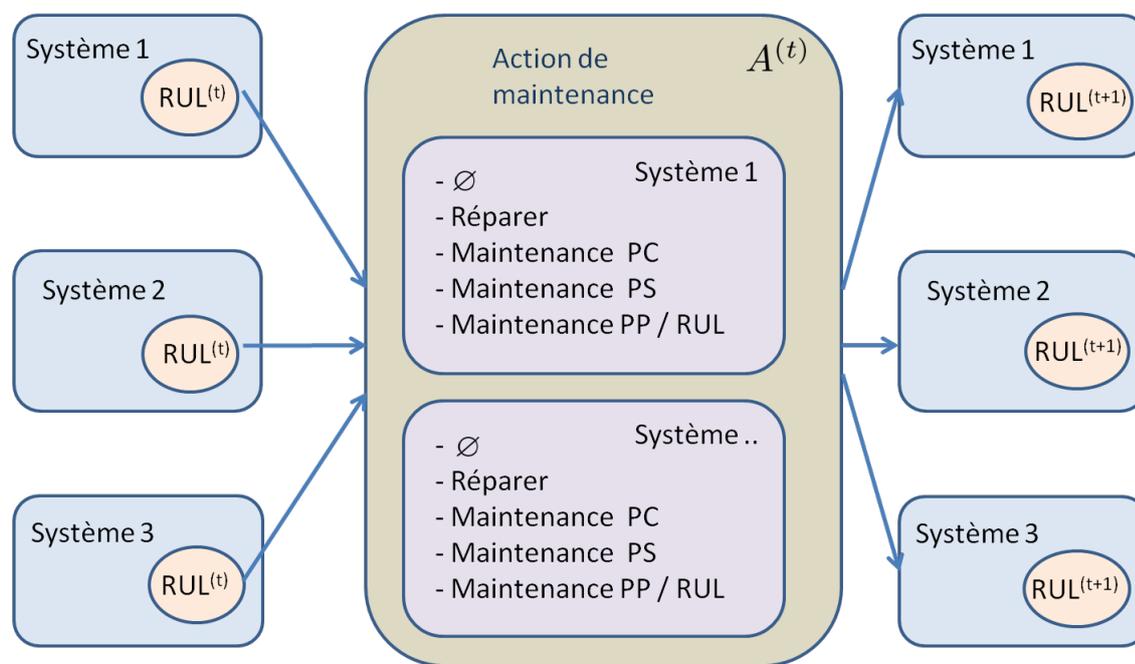
$$|W| = N^3 n_M^2 (T_{max})^2$$

ce qui réduit la complexité d'un facteur  $T_{max}$

Enfin, nous tenons à donner les perspectives générale du modèle graphique développé en illustrant précisément ce vers quoi on veut aller. Comme il a été dit, le but à terme est de modéliser un système multi-composants ou encore une flotte de systèmes sur lesquels on puisse déployer une politique de maintenance basée sur la RUL. Pour cela, il sera nécessaire de connecter la RUL de chaque composant ou système à une boîte de décision générale qui piloterait les actions de maintenance effectuées sur les composants ou la flotte à chaque instant. Une représentation d'un tel modèle est donnée sur la figure 3.15. Chaque système (représenté par un cadre bleu) est décrit par un RBD tel que celui que nous avons représenté à la figure 3.14. Sont représentés ici uniquement le nœud de sortie  $RUL^{(t)}$ . La boîte de décision de maintenance générerait plusieurs décisions possibles pour chaque système présent et selon les stratégies adoptées :

- $\emptyset$  : pas d'intervention
- Réparation curative immédiate
- Maintenance préventive conditionnelle
- Maintenance préventive systématique
- Maintenance prévisionnelle planifiée à un horizon dépendant de la RUL

L'ensemble de ces décisions influencerait sur l'état de chaque système à l'instant suivant ( $t+1$ ).



**FIGURE 3.15:** Types de modèle vers lesquels on veut aller à terme dans le cas par exemple d'une flotte à 3 systèmes

Le choix des règles de décision est un problème de recherche opérationnelle ou d'optimisation avec des critères qui restent à définir (coût, nécessité, criticité des pannes, disponibilité, ...). Une politique de décision est facilement implémentable dans une procédure de calcul reposant à la version algébrique l'algorithme de pronostic. Un exemple de programmation d'une telle politique sur une flotte sera d'ailleurs conduit dans le chapitre suivant. Une représentation graphique par RBD du processus de décision de la maintenance pour une flotte de systèmes, est représentée sur la figure 3.15. Ce RBD est nécessaire si l'on souhaite évaluer voir optimiser une politique décisionnelle de type prévisionnelle. Or la connexion d'un tel réseau avec le réseau pronostic décrit sur la figure 3.14 générerait probablement de nouvelles complexités pour les calculs d'inférence. Explorer les différentes pistes évoquées dans les perspectives pour réduire la complexité des calculs est donc incontournable si l'on souhaite continuer à développer ce type d'approche avec le formalisme des réseaux bayésiens dynamiques.

### 3.5.4 Conclusions

Dans ce chapitre, un algorithme de pronostic a été défini. Celui-ci fournit une valeur numérique de la RUL à chaque instant  $t$ . Dans le but d'améliorer la finesse du modèle de dégradation et donc la qualité des calculs de pronostic, un nouveau modèle de dégradation a été proposé, basé sur les lois de temps de séjours conditionnelles (LTSC). Une possible interprétation des LTSC est la notion de mode de dégradation. Une représentation graphique de l'algorithme de pronostic a ensuite été introduite et pouvant prendre en compte l'utilisation des LTSC. Le choix d'un modèle de dégradation basée sur le modèle graphique de durée, a nécessité de positionner de manière explicite un grand nombre de nœuds discrets de séjour dans le graphe. Cela a conduit à des domaines de grande taille et par voie de conséquence à une forte complexité des calculs d'inférence exacte ; il est possible de la réduire en choisissant un ordre d'élimination de variables et de déplacement dans le graphe utilisant astucieusement les relations d'indépendances conditionnelles existantes. Ce problème de complexité est le principal verrou identifié lié à l'utilisation des réseaux bayésiens comme formalisme ainsi que l'utilisation du MGD. D'un autre côté, le modèle graphique de durée impose peu de contraintes sur la forme des lois apprises. La contrepartie est un stockage numérique de toutes les valeurs possibles ce qui induit par conséquent une forte augmentation de la complexité spatiale du modèle. Le choix des réseaux bayésiens dynamiques permet de regrouper dans un modèle unifié un grand nombre de variables de natures différentes qualitatives et quantitatives. En contrepartie, les calculs d'inférence dépendent de la structure de l'objet mathématique "potentiel" qui est un tableau multidimensionnel dont la taille croît rapidement en fonction du nombre de parents et de la taille des domaines des nœuds du réseau. Les développements qui ont été présentés vont être maintenant testés sur des exemples applicatifs. Ceci est l'objet du prochain chapitre.



# Chapitre 4

## Application des modèles développés sur des données simulées

Initialement, des données de diagnostic embarqué de l'état des portes d'une flotte de véhicules ferroviaires devaient être fournies par un des partenaires industriels du projet DIADEM pour permettre une application sur des données réelles. Ces données n'ont malheureusement pas pu être fournies à temps pour cette thèse. ces données n'ont pas pu être obtenues. En conséquence, l'ensemble de tests conduits dans ce chapitre se fera sur des données simulées. Dans ce chapitre, l'ensemble des algorithmes définis dans le chapitre précédent sont appliqués de façon à démontrer concrètement le type de résultats qu'ils permettent d'obtenir. L'algorithme de pronostic sera testé dans ses deux versions (avec et sans lois conditionnelles) pour comparer les résultats obtenus en terme de qualité du pronostic et de vitesse de convergence. Des calculs d'inférence seront effectués de façon à extraire des distributions de la RUL et donc des intervalles de confiance sur les prédictions obtenues. Enfin, dans une seconde partie, un cas plus applicatif sera présenté en considérant un exemple pouvant s'apparenter à une flotte de véhicules pour laquelle un processus de décision d'actions de maintenance prévisionnelle sera proposé.

### 4.1 Tests sur des données simulées

#### 4.1.1 Création d'une base de données d'observation

Ce paragraphe décrit le processus de création d'une base d'observation discrète d'un système en vue de l'ajustement de sa maintenance. Soit un système possédant  $n$  états  $(x_1, \dots, x_n = x_P)$  et  $m$  modes  $(m_1, \dots, m_M)$ .  $\Delta t$  désigne le pas entre deux diagnostics<sup>1</sup>. Dans une première étape, les temps de séjour dans chaque état sont générés par une loi

---

1. Le pas de diagnostic est supposé constant car cela permet de simplifier le processus de simulation.

de Weibull. Puis, à partir de ces valeurs, les instants de transition entre chaque état sont reconstitués. Enfin, la dernière étape, consiste à discrétiser le temps de chaque séquence selon le pas  $\Delta t$ . Cette procédure générale est illustrée sur la figure 4.1.

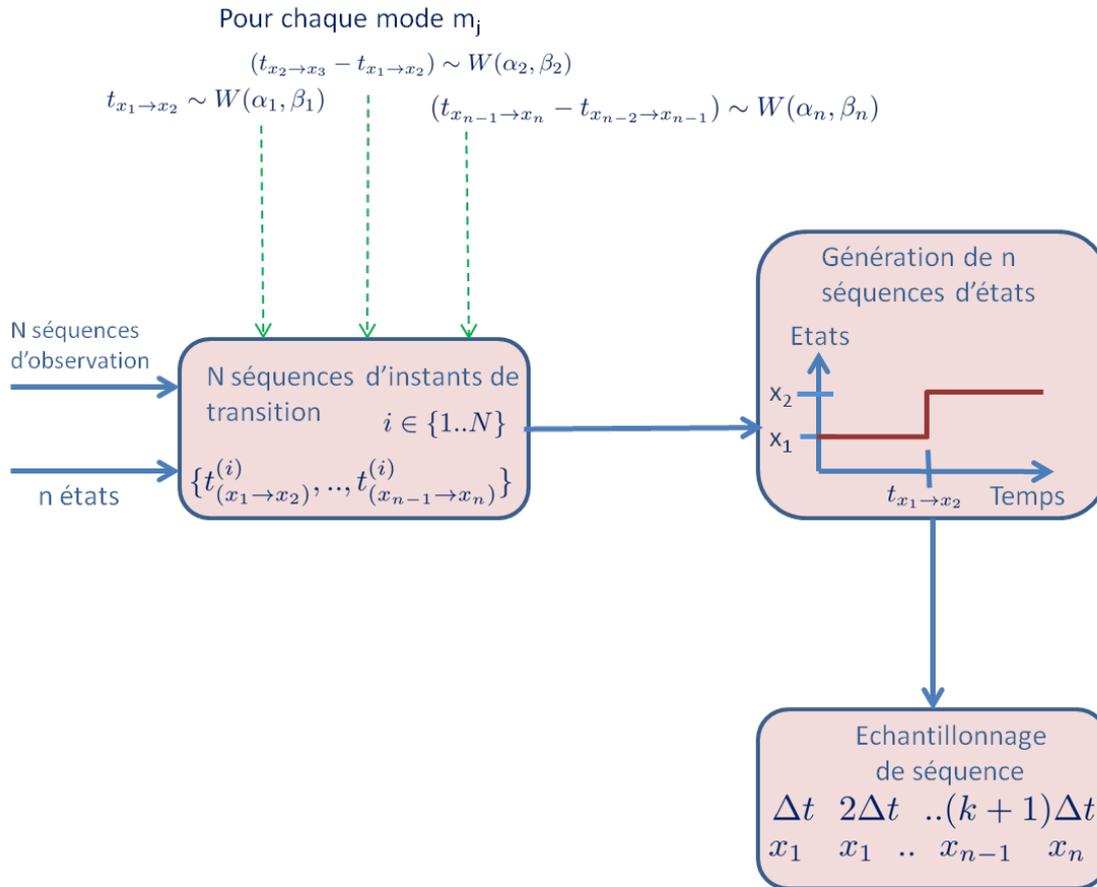


FIGURE 4.1: Processus général de simulation d'une base de données

L'objectif est de tester l'algorithme de pronostic développé dans le chapitre 3. Pour pouvoir tester l'efficacité du modèle intégrant les lois de temps de séjour conditionnelles, plusieurs modes de dégradation seront considérés simultanément. Afin de pouvoir suivre en détail l'évolution de la convergence dans un grand nombre de situations, le système testé contiendra plusieurs états intermédiaires entre les états "neufs" et l'état de "panne". Un système à six états et trois modes de dégradation a donc été simulé. On suppose que celui-ci est périodiquement observable tous les 5 pas de temps. Chaque séquence d'observations à valeur dans l'espace d'états  $\{1, \dots, 6\}$  (nommée aussi trajectoire) va être générée à partir d'un ensemble de temps de séjours (TS) dans chaque état. Ceux-ci vont être simulés à partir de modèles de mélanges à base de lois de Weibull.  $\alpha$  désignera

le paramètre de forme et  $\beta$  le paramètre d'échelle de chaque loi de Weibull associée à un mode de dégradation particulier. Les paramètres utilisés pour simuler les lois sont rassemblés dans le tableau 4.1 et les courbes obtenues sur la figure 4.2.

Loi de Weibull	Mode 1	Mode 2	Mode 3
Poids	0.3	0.3	0.4
	$(\alpha, \beta)$	$(\alpha, \beta)$	$(\alpha, \beta)$
Etat 1	(20,8)	(40,10)	(55,15)
Etat 2	(20,13)	(30,15)	(40,18)
Etat 3	(15,7)	(27,10)	(35,18)
Etat 4	(10,10)	(18,10)	(25,15)
Etat 5	(5,7)	(10,8)	(15,10)

**TABLE 4.1:** Paramètres des lois de Weibull utilisés pour la simulation des données

La taille choisie pour la base d'observation est de 1000 séquences. A partir des lois de Weibull précédemment définies, des temps de séjour à valeurs continues sont tirés pour chaque état et, à partir de cela, sont déduits les instants de transition entre les états. La base d'observation est ensuite divisée en 3 sous ensembles. Le premier tiers des observations est généré avec le mode de dégradation 1, le second tiers avec le mode 2 et le dernier avec le mode 3. Il en résulte un ensemble de trajectoires décrit par le tableau 4.2.

Instants de transition	e1→e2	e2→e3	e3→e4	e4→e5	e5→e6
trajectoire 1	16,36	34,04	46,81	57,48	61,89
...	...	...	...	...	...
trajectoire 550	30,41	59,61	87,58	106,12	114,64
...	...	...	...	...	...
trajectoire 1000	59,22	99,121	134,15	159,523	172,30

**TABLE 4.2:** Caractérisation de quelques trajectoires simulées

A partir des ces instants de transition continus obtenus par sommes cumulées des temps de séjour, il est possible de reconstituer des suites d'observations périodiques

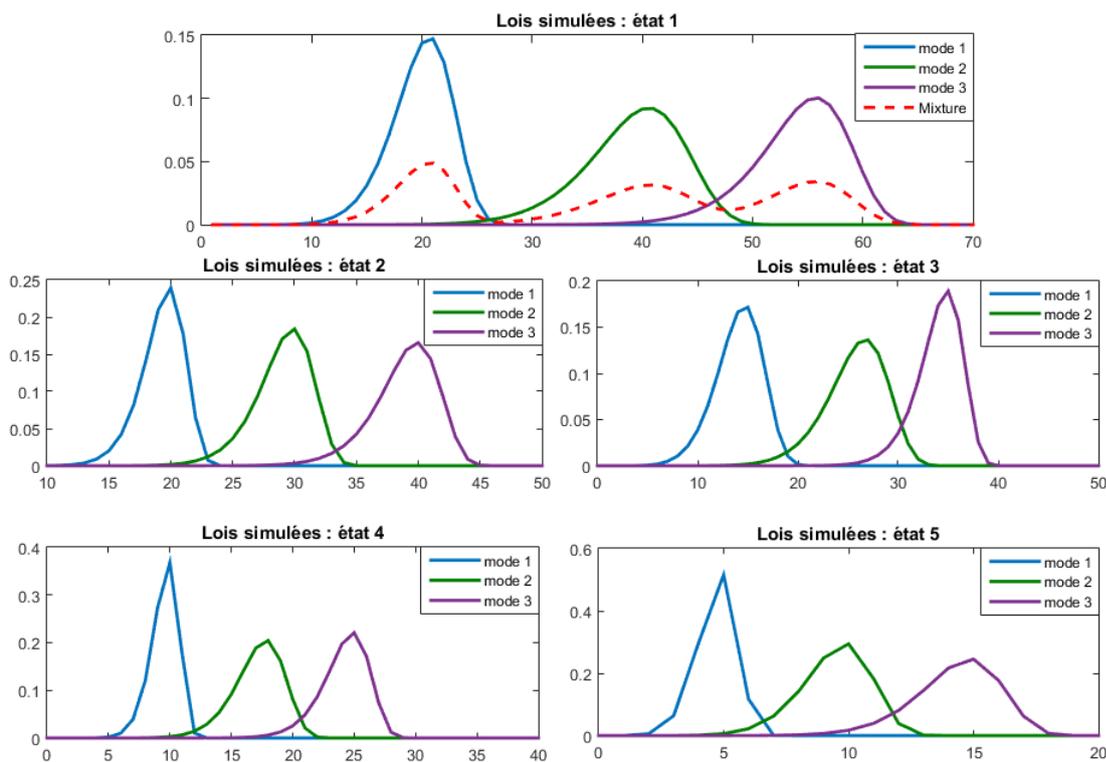


FIGURE 4.2: Loi simulées à partir de mélanges de Weibull

discrètes que l’on obtiendrait avec un pas d’observation égale à 5. Ce processus est illustré dans le tableau suivant 4.3 montrant le résultat obtenu sur la trajectoire 1 de la base REX.

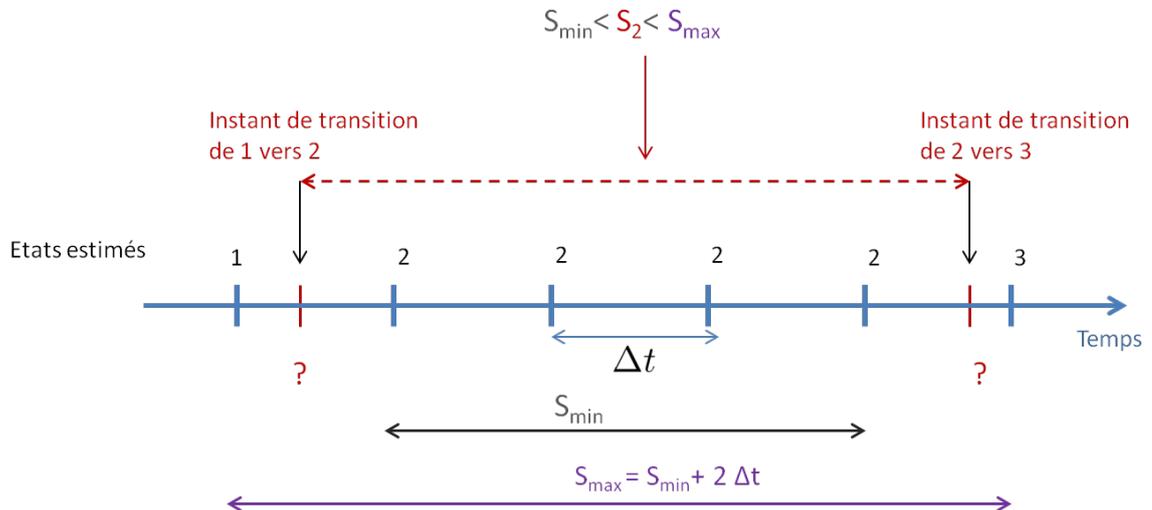
Temps	0	5	10	15	20	25	30	35	40	45	50	55	60	65
Etat observé	1	1	1	1	2	2	2	3	3	3	4	4	5	6

TABLE 4.3: Séquence d’observations discrètes associée à la trajectoire 1

Une fois la base d’observation discrète obtenue, il faut apprendre la loi du mélange de l’état initial en utilisant l’algorithme EM. Pour fixer le nombre de composants du mélange considéré, un apprentissage simple du modèle de mélange de l’état initial est effectué par la méthode MV. Cette étape permet de visualiser la distribution globale des TS dans l’état initial et d’après la forme de la courbe de conjecturer un nombre de

composantes existantes.

Utiliser la méthode MV sur des séquences d'observation discrètes nécessite par ailleurs de reconstituer l'instant de transition manqué lors de chaque changement d'état, de façon à attribuer un temps de séjour dans chaque état pour chaque observation. Par exemple dans le cas de la trajectoire 1 représentée dans la table 4.3, il n'est pas possible d'associer précisément le temps de séjour dans l'état 1. On peut au mieux, donner un encadrement de celui-ci qui est dans ce cas compris entre 16 et 20 unités de temps.<sup>2</sup> Lors du processus d'apprentissage par MV, cet instant de transition est reconstitué par un tirage **uniforme** dans l'intervalle concerné pour associer une valeur de temps de séjour à cette trajectoire. Cette étape est illustrée sur la figure 4.3. Sur cet exemple, le pas de diagnostic est constant. Le temps de séjour  $S_2$  dans l'état 2 est forcément tel que  $S_{min} < S_2 < S_{max}$ . L'erreur commise à l'apprentissage est donc au maximum égale à  $2\Delta t - 1$ .



**FIGURE 4.3:** Illustration du processus d'apprentissage de TSJ à partir d'une trajectoire discrète

La distribution obtenue par EMV sur l'ensemble des 1000 TS reconstitués dans l'état 1 est représentée sur la figure 4.4. Etant donné la forme de la distribution obtenue, il semble raisonnable de postuler l'existence de 3 composantes. L'algorithme EM est donc lancé sous cette hypothèse. A partir du mélange obtenu, un mode actif est associé à chaque observation. Puis les LTSC dans les autres états sont apprises d'après la méthodologie décrite dans le chapitre précédent à savoir par la méthode EMV sur la sous base d'observation associée à chaque mode. L'ensemble des distributions obtenues est

<sup>2</sup> Car d'après une des hypothèses énoncées dans le chapitre 3, si la transition avait eu lieu à  $t=15$ , l'état visible à cet instant aurait été l'état 2.

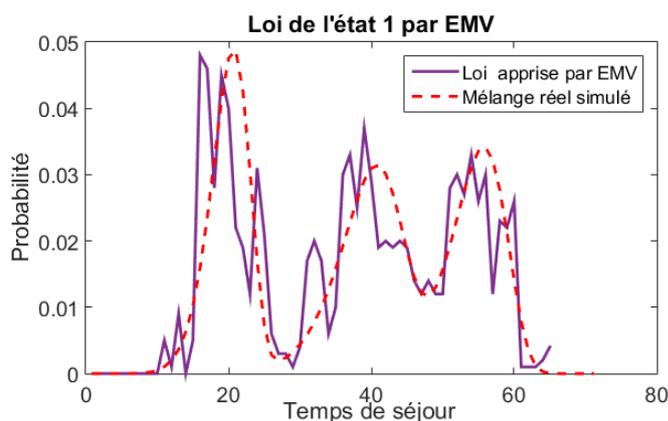


FIGURE 4.4: Apprentissage de la loi de l'état initial par EMV

représenté sur la figure 4.5. Dans ce cas précis, l'apprentissage de chaque mélange se fait de façon relativement correcte pour les états 1 à 4 par rapport aux mélanges originaux qui ont servis à simuler les données. Les lois conditionnelles apprises à l'état 5 ont des supports fortement non disjoints ce qui peut induire des erreurs de recalage comme nous le verrons dans les simulations qui vont suivre. Dans le choix des lois utilisées pour simuler les données, il a fallu s'assurer que la largeur des supports de chaque composante du mélange soit largement supérieure au pas de diagnostic choisi. Dans le cas contraire le processus de discrétisation peut lisser les courbes et faire disparaître les composantes du mélange, rendant l'algorithme EM inopérant. Les mélanges ont donc été volontairement choisis de façon à limiter l'intersection de leurs supports.

### 4.1.2 Test de l'algorithme de pronostic

Disposant maintenant de l'ensemble des lois de temps de séjour conditionnelles nécessaires, l'algorithme de pronostic peut être testé. Pour les tests, 3 trajectoires (chacune générée par un mode différent) ont été considérées.

La première séquence d'observations est générée par le mode 1. L'algorithme de pronostic est testé dans sa version MGD puis dans sa version MGDC.<sup>3</sup> Dans les deux cas, les tirages aléatoires sont remplacés par le choix des moyennes des distributions. La suite des observations, l'ensemble des RUL calculées ainsi que la RUL réelle sont représentés sur le tableau 4.4.

3. MGDC : Modèle graphique de durée intégrant les lois de temps de séjour conditionnelles

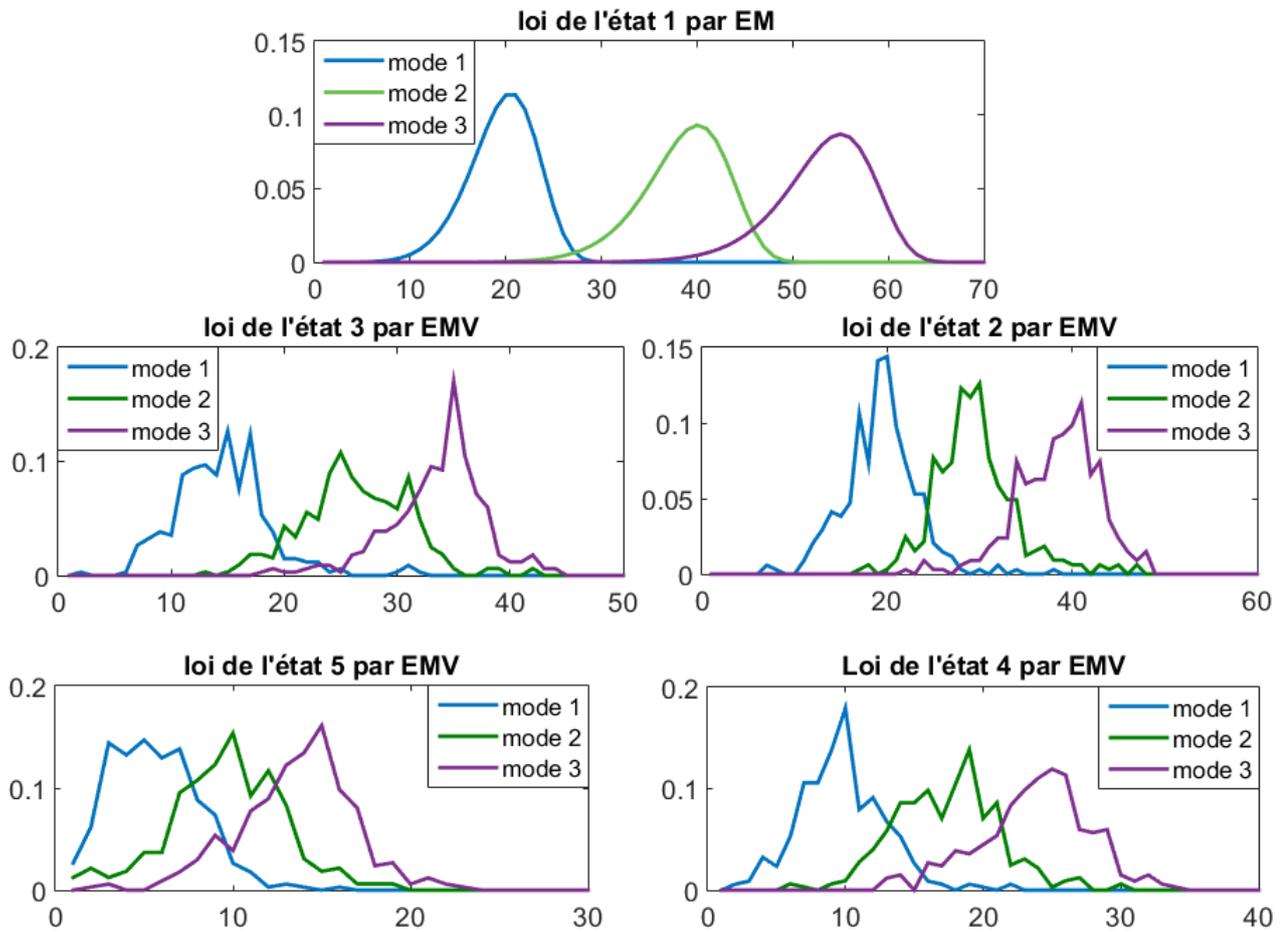
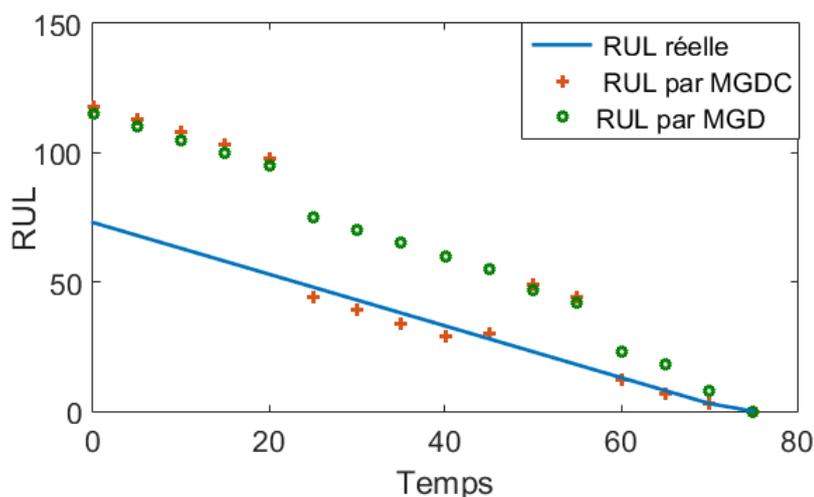


FIGURE 4.5: Apprentissage de la LTSC pour chaque état

Temps	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
État observé	1	1	1	1	1	2	2	2	2	2	3	3	4	4	5	6
RUL réelle	73	68	63	58	53	48	43	38	33	28	23	18	13	8	3	0
RUL GDM	115	110	105	100	95	75	70	65	60	55	47	42	23	18	8	0
RUL GDMC	118	113	108	103	98	44	39	34	29	30	49	44	12	7	3	0

TABLE 4.4: Séquence d'observations discrète associée à la trajectoire 1

Pour comparer les deux versions de l'algorithme, les valeurs obtenues sont représentées sur le graphe 4.6.



**FIGURE 4.6:** Comparatif des résultats obtenus par l'algorithme de pronostic sur une trajectoire suivant le mode 1

On constate que la RUL calculée avec le MGDC donne globalement de meilleures estimations que le MGD hormis aux instants de diagnostic  $t=50$  et  $t=55$ . Pour comprendre cet écart soudain, il convient de décomposer le processus pas à pas dans le cas du MGDC. Les temps de séjours réels utilisés pour simuler la trajectoire dans les états de 1 à 5 sont : (24,21,12,9,5). Le mode initial choisi par l'algorithme de pronostic à  $t=0$  est le mode 2. Lors du premier changement d'état à  $t=25$ , le temps de séjour reconstitué à l'état 1 est 22. Par conséquent, le mode le plus probable est recalé sur le mode 1. Lors du changement d'état de 2 vers 3, le temps de séjour reconstitué à l'état 2 est 25 alors que le temps réel est 21. Cela est lié au biais induit par le pas de diagnostic. On se trouve alors à l'intersection des supports des distributions 1 et 2 et le mode 2 est choisi comme étant le plus probable. Par conséquent les temps de séjour sont recalculés sous cette hypothèse de mauvais mode, ce qui conduit à un calcul de RUL plus éloigné de la RUL réelle. Il faut alors attendre la transition vers l'état suivant (état 3 vers état 4) avec un temps de séjour reconstitué dans l'état 3 égal à 11 pour reconsidérer le mode 1 comme le plus probable. Les dernières prédictions sont alors très précises puisque le recalage s'est fait dans le bon mode. La précision des calculs est donc dépendante de la largeur de l'intersection des supports des lois de chaque mode avec le mode voisin et du rapport de celle-ci avec la marge d'erreur commise dans la reconstitution des temps de séjour, c'est à dire  $2\Delta t - 1$ . Cette erreur d'estimation est la cause du mauvais changement de mode qui s'est produit dans le 1<sup>er</sup> cas test que nous venons de présenter.

L'algorithme est ensuite testé sur une deuxième trajectoire issue de la base et générée par

le mode 2. Les courbes obtenues sont tracées sur la figure 4.7. Dans cet exemple, le mode initialement tiré (mode 2) étant le bon, comme il n'y a pas eu recalage de mode effectué, le pronostic calculé avec le MGDC est toujours un peu meilleur que celui obtenu avec le MGD.

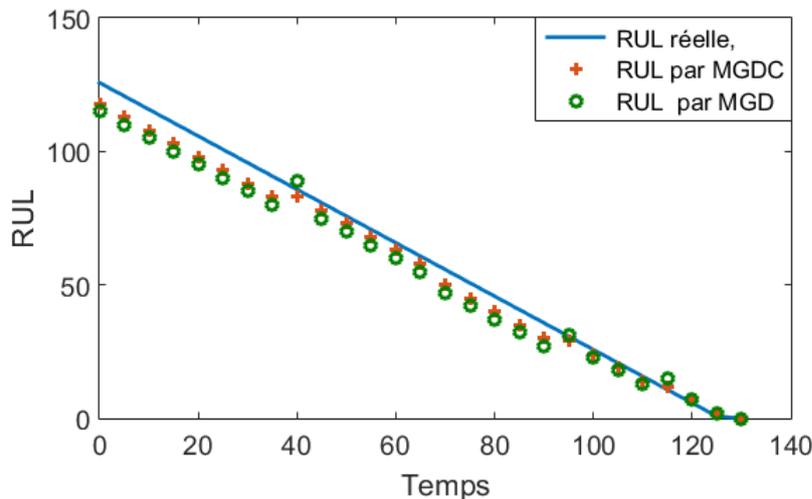


FIGURE 4.7: Comparatif des résultats obtenus sur une trajectoire en mode 2

Enfin la dernière trajectoire est celle générée par le mode 3. Il s'agit du mode le plus lent donc le plus adapté à un pronostic de qualité, puisque laissant beaucoup de temps pour réévaluer correctement les temps de séjour avant la panne. Les résultats obtenus sont présentés sur la figure 4.8. Avec le modèle MGD simple, la convergence s'effectue assez lentement, avec un recalage progressif devenant de bonne qualité (c'est à dire proche de la RUL réelle) à partir de  $t=135$ . Le calcul effectué à partir du MGDC converge beaucoup plus vite puisque le recalage du mode 2 initial vers le mode 3 réel s'effectue à  $t=80$ .

Il est possible de calculer la distribution de la RUL en terme probabiliste en effectuant un calcul d'inférence sur le modèle de dégradation caché montré sur la figure 3.1 du chapitre 3. Ces calculs sont effectués sur certains instants de diagnostic de la première trajectoire. Les résultats obtenus sont présentés sur la figure 4.9.

Chaque figure représente la distribution de la RUL calculée par la GDM ainsi que par le GDMC. La RUL réelle, calculée à partir des temps de séjour qui ont servi à simuler les données, est aussi représentée. Ces courbes permettent de voir pourquoi le MGDC est plus efficace que le simple MGD. En effet, la distribution de la RUL estimée par le MGDC a comme prévu globalement une plus faible variance que la RUL estimée par le MGD et est également en général plus proche de la RUL réelle associée à la

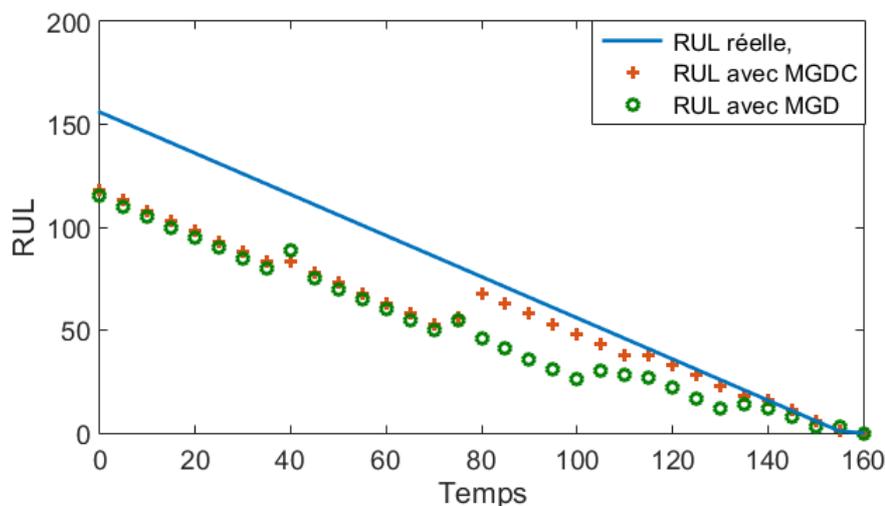
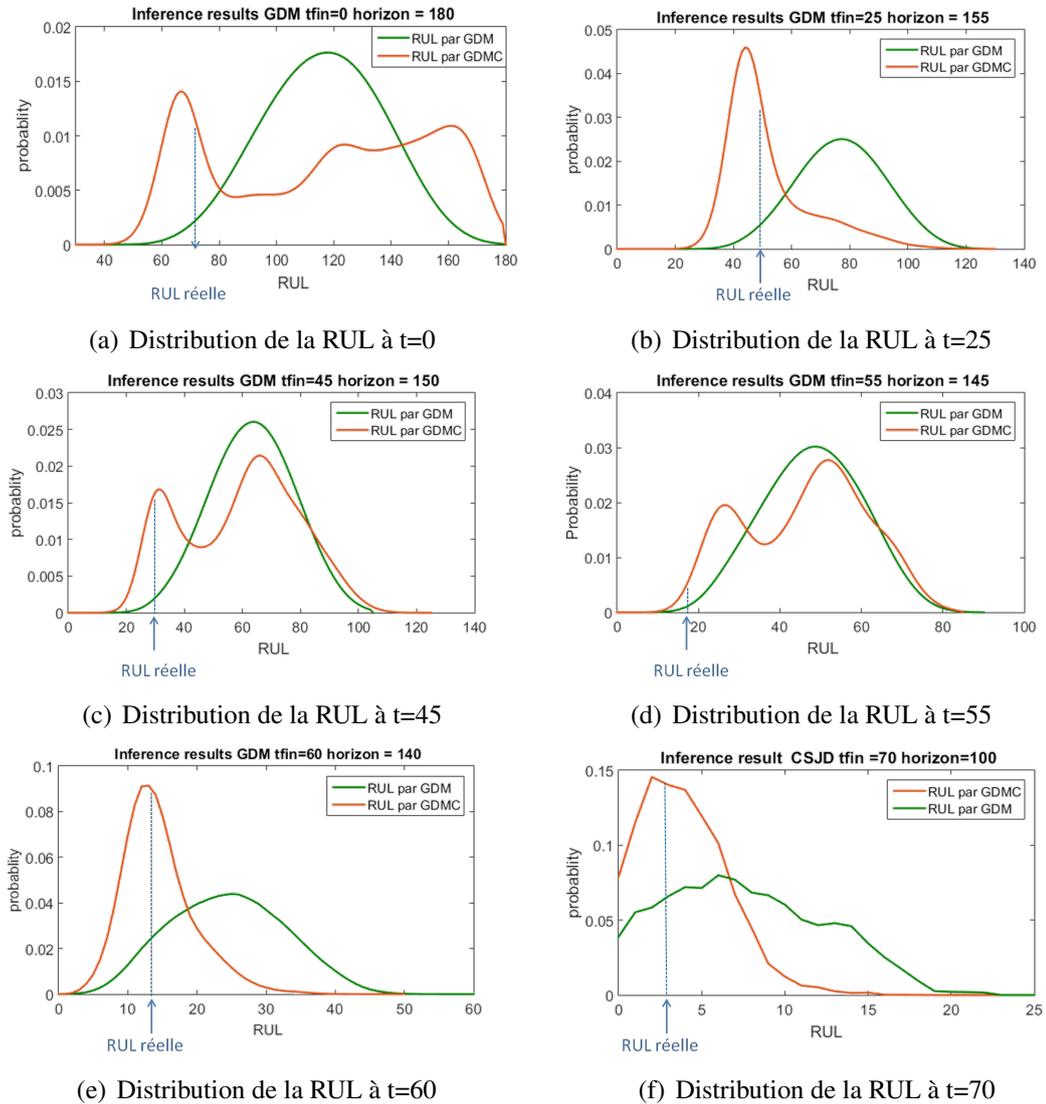


FIGURE 4.8: Comparatif des résultats obtenus sur une trajectoire en mode 3

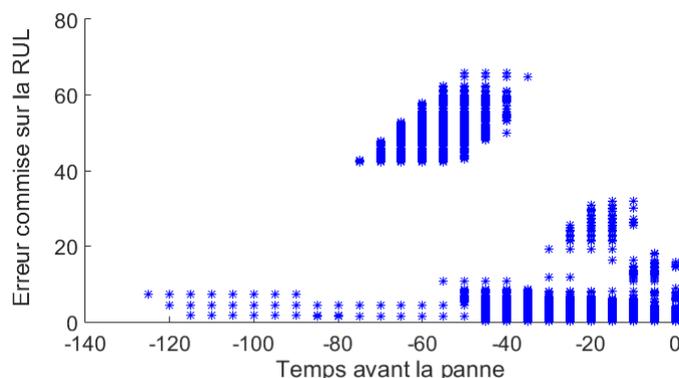
trajectoire considérée. Il est intéressant d'observer l'origine du mauvais recalage du mode effectué à  $t=50$  que l'on avait remarqué précédemment dans les résultats. On constate en effet à l'instant précédent le changement d'état ( $t=45$ ) que la courbe obtenue par le MGDC contient deux pics. Le premier est centré sur 25 et décrit une RUL calculée sous l'hypothèse que le système se dégrade en mode 1 et le second pic centré sur 65 suppose un système dans le mode 2. Cela signifie qu'il y a potentiellement à cet instant deux scénarios différents fortement probables. Or c'est le second qui est choisi car sa probabilité est plus élevée que le premier (0.021 contre 0.0165), il y a donc un recalage vers le mode 2.

La qualité du pronostic dépend de la trajectoire considérée, mais aussi pour ce qui concerne le MGDC, du processus de recalage des modes. Pour avoir une vision globale du comportement de l'algorithme, celui-ci est lancé sur les 1000 trajectoires simulées de la base REX. Pour chaque ensemble de trajectoires dans chaque mode, l'erreur commise sur les prédictions de RUL est tracée à chaque diagnostic. Les résultats obtenus sont présentés sur les figures 4.10 4.11 et 4.12. Concernant les trajectoires du mode 1, on peut constater que, à partir de 40 unités de temps avant la panne, la majorité des estimations ont une erreur inférieure à 10 même s'il existe certains cas où un mauvais recalage du mode induit une grande erreur. Les erreurs importantes sont liées au fait que, les temps de séjour étant très courts, il y a très peu de temps pour le recalage du temps de séjour, qui se fait bien souvent au dernier moment. Pour les trajectoires du mode 2, la convergence est plus stable. La majorité des prédictions se situent dans une marge d'erreur inférieure

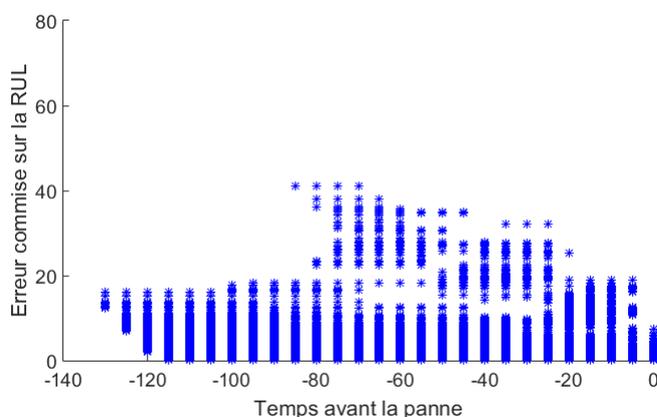


**FIGURE 4.9:** Distribution de la RUL à certains instants de diagnostic

à 10. Cela s'explique par le fait qu'initialement dans l'algorithme, le mode 2 est choisi comme mode initial. Par conséquent, les prédictions sont, dès le début, de bonne qualité. Enfin, dans le cas de trajectoires générées par le mode 3, la convergence générale est plus progressive dans le temps. Comme le temps de dégradation est plus long, les processus ont le temps de se recaler correctement avant la panne, ce qui explique le comportement global observé sur la figure 4.12.

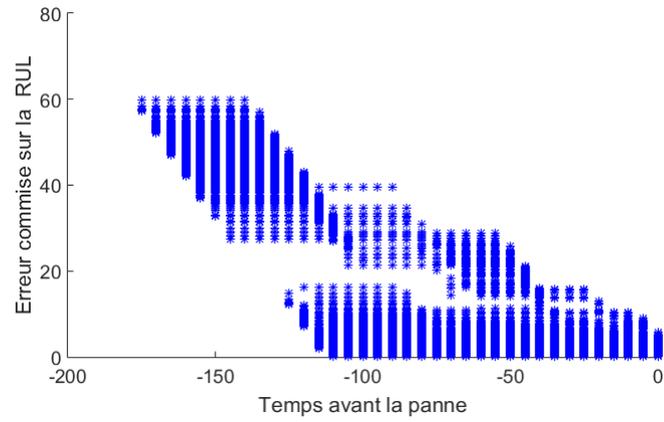


**FIGURE 4.10:** Repartition de l'erreur de RUL sur les trajectoires du mode 1 avec le MGDC



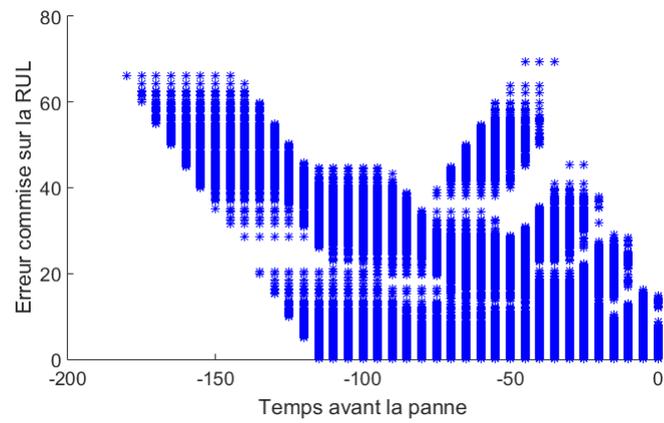
**FIGURE 4.11:** Repartition de l'erreur de RUL sur les trajectoires du mode 2 avec le MGDC

Enfin, le même calcul avec le MGD donne le graphe représenté sur la figure 4.13. La variance de la distribution de l'erreur est plus forte qu'avec le MGDC. Cela est vrai notamment pour les instants qui sont proches de l'instant de panne (de -60 à -20). L'intérêt de MGDC pour ce type de données est donc avéré. Comme il a été mentionné, les données ont été simulées à partir de distributions de temps de séjour composées de mélange de Weibull. Ces distributions auraient pu être choisies différemment, par exemple les composantes de chaque modes auraient pu être ordonnées différemment en fonction des états. Ce type de modifications aurait changé les valeurs de la RUL obtenues mais non le comportement global de l'algorithme de pronostic. Comme il a été dit, les mauvaises estimations surviennent lorsque le mode actif est déterminé à partir d'un temps



**FIGURE 4.12:** Repartition de l'erreur de RUL sur les trajectoires du mode 3 avec le MGDC

de séjour se situant à l'intersection de plusieurs composantes d'un mélange rendant le choix d'un mode actif sujet à un fort risque d'erreur.



**FIGURE 4.13:** Repartition de l'erreur de RUL sur les trajectoires avec le GDM

## 4.2 Application à la gestion d'une flotte de véhicules

L'objectif de cette section est de montrer comment la méthodologie développée dans cette thèse peut être utilisée pour définir une politique de gestion de la maintenance à l'échelle d'un ensemble de systèmes. Une flotte de  $n$  matériels roulants est considérée. A chaque diagnostic sur l'état d'un véhicule, l'algorithme de pronostic recalcule la RUL. Un ensemble de RUL noté  $(RUL_1^{(t)}, RUL_2^{(t)}, \dots, RUL_n^{(t)})$  est donc disponible à chaque instant  $t$ . La maintenance de chaque système génère une période d'indisponibilité. Ces durées sont notées  $(I_1, I_2, \dots, I_n)$ . L'objectif est de mettre en place un algorithme de décision, prenant, en entrée, l'ensemble des RUL connues à un instant  $t$  et qui renvoie une estimation de la durée avant la prochaine action de maintenance prévisionnelle sur chaque système. En d'autres termes, on souhaite disposer d'un calendrier de maintenance qui puisse être mis à jour de façon dynamique en fonction des pronostics calculés. Par hypothèse, l'atelier de maintenance ne disposera que d'une seule voie. Il n'est donc pas possible de réparer deux rames en même temps. L'algorithme 5 présenté ci-dessous prend en entrée, l'ensemble des RUL de chaque système  $(RUL_1^{(t)}, \dots, RUL_n^{(t)})$  et calcule les dates d'intervention à prévoir  $(A_1^{(t)}, \dots, A_n^{(t)})$  en fonction des RUL obtenues. Un matériel en réparation aura une RUL ainsi qu'une action de maintenance (notée  $A$ ) ayant pour valeur  $\emptyset$ .

---

**Algorithme 5** Algorithme de décision de maintenance pour une flotte multisystème.

---

```

1:  $(R_1, R_2, \dots, R_n) = \sigma(RUL_1^{(t)}, RUL_2^{(t)}, \dots, RUL_n^{(t)})$  ( $\sigma$  est la permutation ordonnant
   les RUL dans l'ordre croissant)
2:  $\forall i, A_i = \sigma(A_i)$ 
3: for  $i=n : 1 : -1$  et  $RUL_i \neq \emptyset$  do
4:    $A_i = R_i - I_i$ 
5:   if  $(I_i > R_i - R_{i-1})$  then
6:      $R_{i-1} = A_i - 1$ 
7:   end if
8: end for

```

---

### Explication

Au préalable, seuls les systèmes qui ne sont pas déjà en réparation et donc possédant une RUL sont considérés. Les systèmes  $R_i$  sont dans un premier temps ordonnés dans l'ordre croissant, de celui qui possédant la RUL la plus courte jusqu'à celui possédant la RUL la plus élevée.(L1)

L'algorithme affecte ensuite une date de maintenance en commençant par le dernier système (le moins critique) et en remontant le temps vers le plus critique. L'indice  $i$  de la

boucle remonte de  $n$  jusqu'à 1. Le principe est détaillé pour les deux derniers systèmes de la liste. Pour le dernier système, l'action de maintenance est lancée par  $I_n$  unités de temps avant sa fin de vie. Pour le système précédent  $n-1$ , le calcul de sa date de maintenance se fait en fonction de la date de maintenance du système  $n$ . Le test (sur la L5) détermine si le début de l'action de maintenance de système  $n$  se trouve avant ou après la fin de vie estimée de système  $n-1$ . Si il se trouve après alors la maintenance du système  $n-1$  se fait naturellement dans  $RUL_{i-1} - I_{n-1}$  unités de temps et dans le cas contraire il se fait  $I_{n-1}$  unités de temps avant le début de la maintenance du système  $n$ . De cette façon, par décalage successifs, une date de maintenance inférieure à la RUL est réglée pour chaque système. Une illustration du résultat obtenu par l'algorithme de décision est donnée sur la figure 4.14. Une flotte à 4 systèmes ( $S_1, S_2, S_3, S_4$ ) est considérée et comme variable d'entrée les 4 RUL représentées en rouge ( $RUL_1, RUL_2; RUL_3, RUL_4$ ).

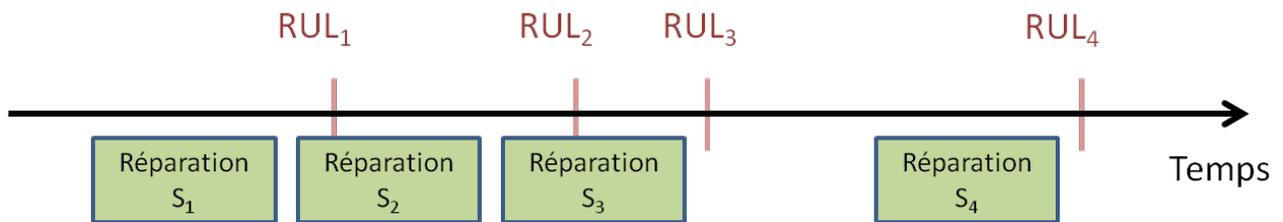


FIGURE 4.14: Répartition des tâches sur un exemple

L'algorithme précédemment énoncé est maintenant testé. Pour cela, une flotte de 4 matériels roulants est considérée. Une telle flotte ainsi que le processus de décision qui s'y applique sont représentés par un RBD sur la figure 4.15. Ce graphe décrit le fait que le calcul des futures dates d'intervention dépend de l'ensemble des 4 RUL calculées à chaque instant. Ces dates sont donc mises à jour à chaque mise à jour des RUL. La variable  $A^{(t)}$  qui contient le temps prévu avant la prochaine action de maintenance a une influence sur l'état réel  $X^{(t)}$  lorsque l'action de maintenance se déclenche à l'instant suivant, c'est à dire quand  $A^{(t)} = 1$ .

Pour chacun des 4 systèmes considérés, une trajectoire est simulée. Les systèmes possèdent chacun 4 états de dégradation qui suivent la même loi. Les LTS de chaque état suivent une distribution de Weibull dont les paramètres sont donnés dans le tableau 4.5. Dans ce cas, la dégradation est modélisée par un simple GDM. Le pas de diagnostic est de 4 unités de temps. Enfin, par hypothèse la réparation d'un système dure 2 unités de temps. Les RUL de chaque système ainsi que les estimations des prochaines dates de maintenance à chaque instant sont représentées dans le tableau 4.6 sur les 48 premiers instants.  $A_i$  désigne l'estimation de la durée restante avant la prochaine action de maintenance prévisionnelle sur le système  $i$ .

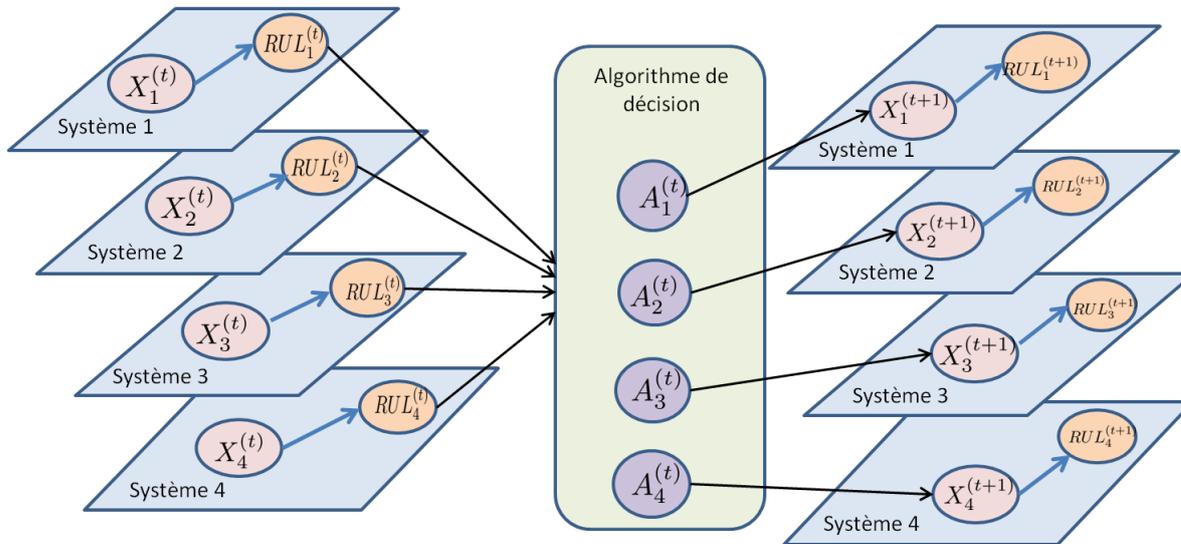


FIGURE 4.15: Représentation de la flotte et du processus de décision par un RBD

Loi de Weibull	$\alpha$	$\beta$
Etat 1	20	12
Etat 2	15	6
Etat 3	8	6

TABLE 4.5: Paramètres des lois de Weibull utilisés pour la simulation des données

D’après les calculs de l’algorithme de décision, l’action de maintenance s’effectue sur le système 1 à  $t=40$ . L’état visible est donc l’état neuf (état 1) à  $t=44$  et la nouvelle RUL est alors calculée par l’algorithme de pronostic avec une valeur pour  $T_{SEC}^{(0)}$  égale à 2 car la maintenance se termine alors à  $t=42$ . Les autres actions de maintenance prévisionnelle ont lieu sur les 3 autres systèmes (2,3 et 4) respectivement aux dates  $t=36$ ,  $t=34$  et  $t=38$ . Chaque système réparé voit alors son état remis à neuf (état 1).

L’application de l’algorithme de pronostic sur des données simulées a donc permis de mettre en évidence, l’intérêt du GDMC dans le cas où les données contiennent plusieurs modes de dégradation induisant des relations de dépendances conditionnelles entre les états. Des calculs d’inférence ont été effectués sur le RBD associé pour obtenir des intervalles de confiance sur les RUL calculées. Enfin, un premier algorithme de décision a été défini de façon à planifier automatiquement des dates d’intervention en fonction

	Temps	0	4	8	12	16	20	24	28	32	36	40	44	48
<b>Système 1</b>	<b>Etat</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>3</b>	$\emptyset$	<b>1</b>	<b>1</b>
	$RUL_1$	39	35	31	27	23	19	15	11	10	3	$\emptyset$	35	31
<b>Système 2</b>	<b>Etat</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	$\emptyset$	<b>1</b>	<b>1</b>	<b>1</b>
	$RUL_2$	34	30	26	22	22	18	14	10	10	$\emptyset$	37	33	29
<b>Système 3</b>	<b>Etat</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
	$RUL_3$	37	33	29	25	21	21	17	13	9	39	35	31	27
<b>Système 4</b>	<b>Etat</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>1</b>	<b>1</b>
	$RUL_4$	36	32	28	24	22	18	14	10	10	6	37	33	29
<b>Durée estimée avant la prochaine action de maintenance prévisionnelle</b>	$A_1$	37	33	29	25	21	17	13	9	8	4	$\emptyset$	33	29
	$A_2$	31	27	23	19	17	13	9	5	4	$\emptyset$	35	31	27
	$A_3$	35	31	27	23	15	19	15	11	2	37	31	27	23
	$A_4$	33	29	25	21	19	15	11	7	6	2	33	29	25

**TABLE 4.6:** Evolution des dates de maintenance prévisionnelles estimées par l'algorithme de décision

des RUL calculées à chaque diagnostic. Une première représentation graphique par RBD a été donnée de ce calcul. Ceci n'est qu'une première étape sur une problématique assez vaste. En effet le codage d'un RBD décrivant exactement le processus de décision décrit dans l'algorithme 5 ferait intervenir un grand nombre de variables quantitatives ( $A_1, \dots, A_n$ ) posant de nouveaux problèmes de complexité notamment lors des calculs d'inférence.

Cet algorithme n'est qu'une première version développée pour l'illustration de la partie applicative de cette thèse. On pourrait imaginer par exemple de tenir compte de l'intervalle de confiance sur la RUL, pour planifier l'action de maintenance non pas juste avant la RUL estimée, mais avec un intervalle de sécurité. La problématique plus générale d'une méthode d'ordonnancement des tâches pour de la maintenance prévisionnelle nécessiterait bien entendu des travaux plus approfondis.

### 4.2.1 Conclusions

Dans ce chapitre, un exemple simple d'application de l'algorithme de pronostic a été donné sur une base d'observation simulée à partir de modèles de mélange de Weibull. Les résultats obtenus ont permis d'illustrer l'intérêt de l'utilisation du modèle graphique de durée avec les lois conditionnelles pour améliorer la qualité des estimations de pronostic dans certains cas. Les calculs d'inférence effectués sur le réseau bayésien ont permis de suivre l'évolution du calcul de pronostic en terme de distribution de probabilité. Dans une seconde partie, une simulation de gestion d'une flotte de véhicule a été présentée. Un algorithme simple de décision a été défini de façon à illustrer une des applications possibles de l'algorithme de pronostic sur des problématiques pratiques. A première vue, il semble que la représentation d'une politique de décision de maintenance prévisionnelle par un RBD soit quelque chose de compliqué à mettre en place. En effet, cela fait intervenir l'ensemble des RUL de chaque système. La loi de probabilité conditionnelle d'un tel nœud de décision serait donc codée avec un potentiel de taille au minimum égale à  $((n - 1)t_{max})^n$ . Il n'est pas pensable de représenter un nœud ayant une telle complexité sous cette forme. Il est donc indispensable d'envisager, d'autres modèles ou d'autres forme de représentation des nœuds pour la partie décision.

# Conclusion générale

Cette thèse avait pour objectif principal de développer des outils permettant de mettre en place une politique de maintenance prévisionnelle pour des systèmes à états discrets périodiquement observables, et pour lesquels des bases de données d'observation étaient disponibles. Ce cadre, assez contraignant, se justifiait par le fait que de nombreux systèmes du domaine ferroviaire possèdent ces caractéristiques. Tout au long de ce travail, nous nous sommes concentrés sur le calcul d'indicateurs de pronostic performants ayant vocations à déterminer une décision de maintenance prévisionnelle ainsi que de leurs représentation par réseaux bayésiens dynamiques.

Un état de l'art des méthodes de pronostic nous a conduit à privilégier une approche fiabiliste compte tenu du type de données disponibles concernant les systèmes étudiés (base REX, avis d'expert). L'objectif principal étant d'intégrer ces outils au modèle générique VirMaLab, il a été décidé de garder le même formalisme de représentation, à savoir les RBD. Ce choix des RBD présentait l'avantage de pouvoir intégrer des données de diagnostic discrètes aux estimations de pronostic par le calcul de probabilités conditionnelles et donc de lier en quelques sortes l'approche fiabiliste se concentrant sur une description stochastique du processus de dégradation et l'approche basée sur les données s'attachant à décrire la tendance (en terme de dynamique) contenue dans les séquences d'états observées.

Le formalisme relatif au modèle générique VirMaLab a ensuite été présenté. Les RB et RBD ont été introduits, en insistant particulièrement sur la notion d'indépendance conditionnelle qui est déterminante dans la complexité des calculs d'inférence. Puis, l'approche choisie par l'équipe Data et Mobilité du GRETTIA pour modéliser la dégradation a été détaillée. Le choix de considérer un processus de dégradation comme un ensemble de temps de séjour a conduit à formaliser ce processus par un modèle semi-Markovien introduit par Donat nommé "modèle graphique de durée". Ce modèle est un RBD modélisant la dégradation à partir d'un ensemble de lois de temps de séjour suivant n'importe quelle distribution probabilité discrète. Le modèle générique VirMaLab est enfin décrit avec ces différents modules de diagnostic, de maintenance et d'optimisation.

Ce cadre de travail étant posé, la première contribution de cette thèse a consisté à

définir un algorithme de pronostic basé sur le formalisme des modèles graphiques de durée, c'est à dire calculant et mettant à jour la RUL à partir des distributions de temps de séjour de chaque états et de l'ensemble des diagnostics disponibles. Une analyse du fonctionnement de cet algorithme a montré que l'erreur commise sur les prédictions de RUL est fortement déterminée par la variance des lois de temps de séjour caractérisant le processus de dégradation considéré.

Nous avons donc cherché à développer un nouveau modèle de dégradation (le MGDC) rejetant l'hypothèse d'indépendance entre les loi de temps de séjour en cherchant à détecter l'existence de divers modes de dégradations coexistants pour en capter la dynamique. Ce nouveau modèle repose sur l'existence d'un modèle de mélange dont chaque composante est associée à un mode de dégradation. L'identification d'un mode actif permet de faire intervenir des distributions de temps de séjour conditionnelles de variance plus faible et donc réduisant le risque d'erreur. L'introduction de lois de temps de séjour conditionnelles peut être vue comme une volonté de mélanger de façon plus profonde l'approche de pronostic fiabiliste à celle basée sur les données. En effet, l'idée principale est de parvenir à capter la dynamique de dégradation à l'origine des observations en ligne, pour croiser cette information avec une connaissance stochastique de cette dégradation apprise hors-ligne. De mauvais résultats peuvent apparaître lorsqu'il y a un mauvais recalage des modes dans le calcul. Ce type de situation survient lorsque le temps de séjour considéré pour l'estimation du mode le plus probable est situé à l'intersection de plusieurs composantes d'un mélange. De plus, il est nécessaire que des mélanges soient visibles dans les distributions pour être appris correctement par l'algorithme EM considéré dans la démarche proposé. Or le pas de discrétisation du diagnostic peut induire un biais dans l'apprentissage des temps de séjour et rendre difficile la détection d'un mélange. Enfin, la procédure d'apprentissage repose sur l'hypothèse que chaque mode peut être identifié uniquement par le temps de séjour à l'état initial et qu'un système ne peut pas changer de mode même si, dans la pratique, l'algorithme de pronostic ne tient pas compte de cette hypothèse et peut détecter un changement de mode. Une perceptive envisagée pour remédier à ces inconvénients serait de changer la procédure de sélection de mode dans le cas de plusieurs modes concurrents voir de tenir compte d'une pondération des lois de chacun des modes concurrents pour le calcul de pronostic en lui même, sans raisonner à partir d'un seul mode le plus probable comme cela est fait actuellement. Une autre idée consisterait à définir une méthode qui ne soit pas dépendante de l'existence de mélanges visibles dans les lois de temps de séjour en cherchant des dépendance entre les lois de temps de séjour des différents états sans se baser sur des mélanges. Le MGDC a donné lieu à une nouvelle version de l'algorithme de pronostic intégrant les lois de temps de séjour conditionnelles. Une représentation par un RBD spécifique a de plus été proposée. Enfin, la dernière contribution fut de donner une représentation par RBD de l'ensemble du processus de calcul de pronostic dans le but d'intégrer cet outil au modèle VirMaLab. Ceci a fait apparaître à nouveau les questions de complexité de calcul liées à l'utilisation

des modèles de durée dont le principe est de transférer la discrétisation du temps du processus de dégradation dans le domaine d'un nœud de durée de façon explicite pour pouvoir en contrôler la distribution. Le principe d'un RBD discret repose sur le stockage de l'ensemble des configurations d'événements possibles entre les différentes variables du graphe. Cela entraîne de grandes complexités spatiales et algorithmiques pour les calculs d'inférence ce qui rend inutilisable l'algorithme standard d'interface sur le RBD de pronostic dans sa version de base. Une procédure d'inférence spécifique basée sur l'algorithme d'interface et exploitant certaines relations d'indépendance conditionnelle a donc été proposée pour réduire cette complexité. Cela a permis de lancer des calculs mais avec une certaine limitation concernant la borne supérieure des lois de temps de séjour de chaque état. Ce problème de complexité avait déjà été évoqué durant la thèse de Roland Donat. Le choix des RBD présente des avantages très intéressants, particulièrement dans la possibilité de regrouper dans un modèle unifié des variables de natures très différentes. Par conséquent, pour tenter de résoudre ces difficultés, les perspectives à envisager sont les suivantes :

- Travailler au niveau du RBD en lui-même en modélisant la durée par des variables continues pour ne plus être dépendant, en terme de complexité, de la taille des domaines des nœuds de durée
- Améliorer la procédure d'inférence en optimisant le codage par l'exploitation des structures semi-creuses des matrices représentant l'algorithme de pronostic
- Exploiter mieux les relations d'indépendance conditionnelle aussi bien au niveau du graphe que des lois (chercher par exemple les dépendances faibles).
- Considérer des méthodes d'inférence approchées par exemple basées sur des approches de type Monte-Carlo qui ne sont que très peu dépendantes de la taille des nœuds du réseau.

Enfin, les algorithmes de pronostic précédents ont été testés (faute de données réelles disponibles) sur des données simulées pour faire ressortir l'intérêt du MGDC. Le système simulé possédait 6 états discrets et 3 modes de dégradation ordonnés de la même façon dans chaque état avec un pas d'observation de 5 unités de temps. Les résultats ont montré que le MGDC est globalement plus précis que le MGD dans ce type de situation faisant effectivement intervenir plusieurs modes de dégradation. Ceci s'explique par l'utilisation de lois de temps de séjour à variance plus faible lorsque le bon mode de dégradation actif est détecté par l'algorithme de pronostic. Un algorithme de décision a ensuite été introduit dans le but de mettre en place une politique de maintenance prévisionnelle sur une flotte de matériels roulants à partir de l'ensemble des RUL disponibles. Représenter graphiquement un processus de décision basée sur les RUL de chaque sous système par un RBD ne semble pas faisable avec des nœuds discrets car la taille du nœud de décision croît de façon exponentielle avec le nombre de systèmes considérés. Il semble donc nécessaire de passer soit à des nœuds à domaine continu, soit d'envisager d'utiliser un autre formalisme pour traiter la partie décisionnelle en terme probabiliste.

L'ensemble des travaux présentés dans cette thèse permettent à partir de l'algorithme de pronostic et de l'algorithme décision présentés, de définir une politique de maintenance prévisionnelle pour une flotte de systèmes à états discrets périodiquement observables. Le MGDC donne, quant à lui, la possibilité d'améliorer les indicateurs de fin de vie considérés en prenant en compte l'existence de plusieurs modes de dégradation coexistants, ce qu'il n'était pas possible de faire jusque là. La représentation par RBD permet de fournir quant à elle des intervalles de confiance associés aux estimations de pronostic et d'effectuer des calculs de fiabilité. La complexité induite par le RBD pronostic, ne permet pas encore d'intégrer les noeuds de décision dans le calcul de ces indicateurs et nécessite des limitations concernant la taille des lois de temps de séjour considérées. Toutefois, ces travaux ont permis de cerner précisément les causes de la complexité des calculs d'inférence exacte et de proposer plusieurs perspectives pour surmonter ces difficultés.

## Remerciements

Cette thèse fait partie du projet DIADEM ANR -13 -TDMO -04, financé par l'Agence Nationale pour la Recherche (ANR), en partenariat avec l'université de technologie de Troyes, Faiveley Transport, Keolis Rennes et l'université de technologie de Compiègne.

# Bibliographie

- [1] Adeline R., Gouriveau R., Zerhouni N, *Pronostic de défaillance : maîtrise de l'erreur de prédiction*, MOSIM (2008)
- [2] Allen D., Darwiche A., *New advances in inferennce by recursive conditioning* in UAI, pp2-10 (2003)
- [3] Norme AFNOR X 60-000 (2002)
- [4] D'Ambrosio B, *Incremental probabilistic inference*. Proceeding of the ninth Conference on Uncertainty in Artificial Intelligence,pp 116-125 (1999)
- [5] Arnborg S.A., *Efficient algorithms for combinatorial problems on graphs with bounded decomposability—A survey*, BIT 25 2–23 (1985)
- [6] Barros A., *De l'information de surveillance à la construction d'indicateurs de décision en fiabilité et en maintenance : contribution par approches stochastiques.*, HDR thesis, Université de technologie de Compiègne, France (2010)
- [7] Baruah P., Chinnam R.B , *HMMs for diagnostic and pronostic in machining process ; International Journal of Producing Research*, vol 43-n6, 1275-1293 (2004)
- [8] Bensi M., Kiureghian D., Straub D., *Efficient bayesian network modeling of systems*, "Relibaility engineering and System Safety" vol 112, pp. 200-213 (2013)
- [9] Boudali H., Dugan J., *A new bayesian network approach to solve dynamix trees*, in IEEE Relibaility and mainteability Symposium, pp 451-456 (2005)
- [10] Bouillaut L., *Les modèles graphiques probabilistes : de la modélisation de la dégradation à l'optimisation de la maintenance de systèmes complexes*, Thèse HDR, Université paris-Est (2014)
- [11] Bouillaut L., Donat R., Neji A., Aknin P. , *Modelling the maintenance of multi-components system : Comparison of two Graphical Models approaches*, 13th IFAC Symposium on Information Control Problems in Manufacturing, Moscow, Russia (2009)
- [12] Bouillaut L., Aknin P., Donat R., Bondeux S. , *VirMaLab : A generic approach for optimizing maintenance policies for complex systems*, 9th World Congress on Railway Research (WCRR) (2011)

- [13] Bouillaut L., Aknin P., Donat R., Bondeux S., Dubois S., *VirMaLab (Virtual Maintenance Laboratory) : a decision support tool for optimization of maintenance strategies* Recherche Transport Sécurité (RTS), Volume 27, Numéro 107, pp. 241-257 (2011)
- [14] Byington, C.S, Roermer, M.J, Kacprzyński, G.J, *Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance*, IEEE Aerospace Conference (2002)
- [15] Cho H.C., Fadali S., *Online estimation of dynamic bayesian network parameters ;* In international joint conference on neuronal network (2006)
- [16] Dechter R., *Bucket elimination : A unifying framework for reasoning*, Artificial Intelligence 113(1-2), 41-85 (1999)
- [17] Dempster A.P., Laird N. M., Rubin D. B., *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal statistical Society, B(39) pp. 1-38 (1977)
- [18] Donat R., Leray P., Bouillaut L., Aknin P., *A dynamic Bayesian network to represent discrete duration models*, Neurocomputing, Volume 73, Issues 4–6, January, Pages 570-577 (2010)
- [19] Donat R., *Modélisation de la fiabilité et de la maintenance par modèles graphiques probabilistes : Application à la prévention des ruptures de rails*, Thèse : Institut national des sciences appliquées de Rouen (2009)
- [20] Fay A, Jaray J., *A justification of local conditioning in bayesian networks*, "International Journal of Approximate Reasoning, vol 24 no 1, pp59-81 (2000)
- [21] Farrar C.R., Lieven N.A.J., *Damage prognosis : the future of structural health monitoring Philosophical transaction of the royal society*, vol 365, pp 623-632. (2007)
- [22] Fisher, R., *On the mathematical foundations of theoretical statistics*. transactions of the Royal Society of London. Series A, Containing papers of a Mathematical or Physical Character, Volume 222, pp 309-368 (1922)
- [23] Gorjian, N., Ma, L., Mittinty, M., Yarlalagadda, P., and Sun, Y. *A review on degradation models in reliability analysis*. In Proceedings of the 4th World Congress on Engineering Asset Management (2009)
- [24] Horenbeek V., Scarf P.A, Cavalcante C.A.V., *On the use of predictive information in a joint maintenance and inventory policy*, Safety, Reliability and Risk Analysis : Beyond the Horizon – Steenbergen et al. (Eds) Taylor & Francis Group (2014)
- [25] Henrion M., *Propagating uncertainty in Bayesian networks by probabilistic logic sampling ;* Uncertainty in Artificial Intelligence 2, pp 149-163 Elsevier Science Publishers B. V. Amsterdam (1988)

- [26] Jardine A., Lin D., Banjevic D., *A review on machinery diagnostics and prognostics implementing condition based maintenance*, Mechanical System and signal processing (2005)
- [27] Jensen F.V., *Bayesian Networks and Decision Graphs*, Springer (2007)
- [28] Kjaeruff U., *Approximation of Bayesian networks through edge removals*, Research Report IR-93-2007, Department of Computer Science, Aalborg University Denmark (1993)
- [29] Kothamasu, R., Huang, S. & Verduin, W., *System health monitoring and prognostics : a review of current paradigms and practices*, International Journal of advanced manufacturing technology, 28(9-10), pp. 1012-1024 (2006).
- [30] Langseth H., Portinale L., *Bayesian networks in reliability*, *Reliability Engineering and System Safety*, vol 92, no 1, pp 92-108 (2007)
- [31] Lauritzen S.L, Spiegelhalter D.J, *Local computations with probabilities on graphical structures and their applications to expert systems*. Journal of Royal Statistical Society B,50 : 2,pp 157-224 (1988)
- [32] Lebold, M., Thurston, M., "Open standard for condition based maintenance and prognostic system", In 5th Annual Maintenance and reliability modeling (2001)
- [33] Lorton A., Fouladirad M., Grall A. *Pronostic d'un système complexe pour la maintenance conditionnelle*, Proceedings of Lambda Mu 17, Innovation et Maitrise des risques, 2 934-943 (2013)
- [34] Luo J., Namburu M., Pattipati K., Qiao L., Kawamoto M., Chigusa S, *Model-based prognostic techniques*. AUTOESTCON proceedings, IEEE Systems Readiness technology Conference, Anaheim California (2003)
- [35] Luo J., pattipati K. R., Qiao L., Chigusa S., *Model-based prognostic techniques applied to a suspension system*. IEEE transactions on systems, man, and cybernetics–part a : systems and humans, vol,38 (5), pp 1156-1168 (2008)
- [36] Mahadevan S., Zhang R., Smith N., "Bayesian networks for system reliability reassessment", *Structural Safety*, vol 23, no 3, pp. 231-251 (2001)
- [37] McNaught K.R., Zagorecki A., *Using Dynamic Bayesian network for Prognostic Modelling to inform maintenance Decision Making* (2009)
- [38] Mathur A., Cavanaugh K.F, Kattipati K.R., Willett P.K and Galie G.R., *Reasoning and modeling systems in diagnostic and prognosis*, The international Society for optical engineering , vol 4389 , pp 194-203 (2001)
- [39] Medjaher K., Tobon-Mejia D.A., Zerhouni N., *Remaining useful life Estimation of Critical Components With Application to Bearings*, *Reliability, IEEE Transactions on* , vol.61, no.2, pp 292-302 (2012)

- [40] Mihajlovic V., Pekovic M., *Learning Dynamic bayesian network, a state of the art* ; Computer science department, University of Twente (2001)
- [41] Moncay F., Vernier J.P, Maintenance, *Méthode et organisation pour une meilleure productivité, 3 ème édition* : Dunod (2012)
- [42] Muller A., Suhner M., Iung B., *Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system* ; Reliability and engineering system safety 234-253 (2008)
- [43] Muller A., *Contribution a la maintenance prévisionnelle des systèmes de production par la formalisation d 'un processus de pronostic* ; Université Henri Poincaré (2005)
- [44] Murphy K.P, *Dynamic Bayesian network : Representation, inference and Learning*, PhD Thesis (2002)
- [45] Naim P., Wuillemain P-H., Leray P., Pourret O., Becker A., Réseaux Bayésiens 3ème édition, Eyrolles (2008)
- [46] Neves M., Santiago L., Maia C., *A condition-based maintenance policy and input parameters estimation for deteriorating systems under periodic inspection*, Computer & Industrial engineering (2011)
- [47] Nguyen K., Do Van P., Grall A., *A predictive maintenance strategy for multi-component systems using importance measure* , Safety, Reliability and Risk Analysis : Beyond the Horizon – Steenbergen et al. (Eds) Taylor & Francis Group (2014)
- [48] Van Noortwijk J.M, *A survey of application of Gamma processes in maintenance*, Reliability engineering and system safety (2009)
- [49] Norme ISO, 13381-1 (2004)
- [50] Paillat N., , *Méthodologie de surveillance dynamique à l'aide de réseaux neuroflous temporels*, thèse Université de Franche comté (2006)
- [51] Pearl J., *Fusion , propagation an destructuring in beloef networks* Artificial Intelligence, 29(3), pp.241-288 (1986)
- [52] Pecht M., Jaal R., *A prognostic and health management roadmap for information and electronics-rich systems*, Microelectronic reliability, vol 50, pp 317-323 (2010)
- [53] Peysson F. *Contribution au pronostic de systèmes complexes* ;Thèse Université Paul Cézanne (2009)
- [54] Rausand M. , Hayland A. , *System Reliability Theory - Models, Statistical methods and Applications*, Wiley Series in Probability and Statistics (2004)
- [55] Ribot P. ; *Towards Diagnosis/Prognosis integration for the maintenance of complex systems*, PhD Thesis, Université de Toulouse (2008)
- [56] Rozas R., *Intégration du retour d'expérience pour une stratégie de maintenance dynamique*, Thèse de l'université Paris-Est (2014)

- [57] Saxena A., Celaya J., Saha, B., Saha S., goebel K., *Metric for offline Evaluation of prognostic Performance*, International Journal of prognostic and health Management ISSN 2153-2648 (2010)
- [58] Sankararaman S., Saxena A.J., Goebel K., *Are Current Prognostic Performance Evaluation Practices Sufficient and Meaningful*, Annual Conference of the Prognostics and Health Management Society (2014)
- [59] Schwabacher M., Goebel K., *A survey of Artificial intelligence for prognostics*, Proceeding of AAAI Fall Symposium (2007)
- [60] Schwarz G.E, *Estimating the dimension of a model* , Annals of Statistics, vol 6, no 2, p461-464 (1978)
- [61] Tobon-Mejia D.A. , Medjaher K., Zerhouni N., Tripot G. ; *Hidden Markov models for failure diagnostic and prognostic*, International Conference on Prognostics and Health Management - PHM (2011)
- [62] Vachtsevanos G., Lewis F.L., Roermer M., Hess A., Wu B. *Intelligence fault diagnosis and prognosis for engineering systems*, Wiley (2006)
- [63] Wang H., A survey of maintenance policies of deteriorating systems. European Journal of Operational Research 139, 469–489 (2002)
- [64] Weber P., Simon C. , *Méthode de modélisation par Réseaux bayésien de la fiabilité d'un système multi-état* , 11ème Congrès international en qualité , Sureté de Fonctionnement de Développement Durable QUALITA (2015)
- [65] Weber P., Medina-Oliva G., Simon C, Iung B., *Overview on bayesian networks applications for dependability, risk analysis and maintenance areas*, "Engineering Applications of Artificial Intelligence, vol 25, no 4, pp 671-682 (2012)
- [66] G.Wiedl, A.L Madsen, E.Dahlquist , *Object oriented bayesian network for industrial process operation*, In bayesian modelling application workshop associated with the 19th conference of uncertainties in artificial intelligence (2003)
- [67] Willemin P., Torti L., *Structured probabilistic inference*, International journal of Approximate Reasoning, vol. 53, no. 7 pp 946-968 (2012)
- [68] Zhang N., Poole D., *A simple approach to bayesian network computations*, proceedings of the tenth Canadian Conference on Artificial Intelligence (pp. 171-178) (1994)
- [69] Zio E., *An introduction to the basic of Reliability and Risk Analysis, Series on Quality, Reliability and Engineering Statistics*, ed Singapore : World scientific publishing (2007)