



**HAL**  
open science

# Diagnostic des défauts dans les systèmes à évènements discrets soumis à des contraintes temporelles

Sara Rachidi

► **To cite this version:**

Sara Rachidi. Diagnostic des défauts dans les systèmes à évènements discrets soumis à des contraintes temporelles. Automatique / Robotique. Normandie Université, 2019. Français. NNT : 2019NORMLH24 . tel-02441760

**HAL Id: tel-02441760**

**<https://theses.hal.science/tel-02441760>**

Submitted on 16 Jan 2020

**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.



Normandie Université

## THESE

**Pour obtenir le diplôme de doctorat**

**Spécialité Automatique, Signal, Productique, Robotique**

**Préparée au sein de « Université du Havre Normandie »**

### **Diagnostic des défauts dans les systèmes à événements discrets soumis à des contraintes temporelles**

**Présentée et soutenue par  
Sara RACHIDI**

<b>Thèse soutenue publiquement le 22/11/2019 devant le jury composé de :</b>		
Mme Audine SUBIAS	Maitre de Conférences - HDR / INSA / Université de Toulouse	Rapportrice
M. Armand TOGUYENI	Professeur des Universités / Ecole centrale de Lille	Rapporteur
M. Houcine CHAFOUK	Enseignant chercheur - HDR / ESIGELEC	Examineur
M. Lotfi NABLI	Professeur des Universités / ENIM / Monastir Tunisie	Examineur
M. Édouard LECLERCQ	Maitre de Conférences / Université le Havre Normandie	Encadrant de thèse
M. Dimitri LEFEBVRE	Professeur des Universités / Université le Havre Normandie	Directeur de thèse

**Thèse dirigée par Dimitri LEFEBVRE, laboratoire GREAH (EA 3220)**





---

## Dédicaces

*A celle qui a consacré sa vie pour l'éducation de ses enfants*

*A celle qui m'a élevé avec amour et tendresse*

*A celle qui a toujours cru en moi*

*A ma maman chérie.*

*A celui pour qui éducation rime avec rigueur et travail*

*A celui qui m'a toujours guidé vers la bonne voie*

*A celui qui s'est sacrifié pour ma réussite*

*A mon cher papa.*

*A ma chère sœur **Houda** et mon beau-frère **Zakaria** qui ont toujours soutenu leur petite sœur avec tant d'amour et de tendresse.*

*A mon cher oncle **Khalid** ainsi qu'à sa femme **Ruth-Maria** qui m'ont toujours soutenu et à qui je témoigne mon affection et ma profonde reconnaissance.*

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être.*

*A toute ma famille.*

*A tous mes ami(e)s.*

*A ceux que j'aime et qui m'aiment ...*

*Pasar27*



---

# Remerciements

Mes trois années de thèse furent une immersion dans une aventure passionnante aux révélations enrichissantes, aussi bien sur le plan de la recherche intellectuelle que sur le plan personnel. Cette thèse représente l'aboutissement d'un chapitre de ma vie. Nombreuses sont les personnes qui ont contribué à son accomplissement. Je souhaite les remercier.

Mes premiers remerciements s'adressent tout naturellement à mon directeur de thèse : M. Dimitri LEFEBVRE ainsi qu'à mon encadrant M. Édouard LECLERCQ, d'abord pour la confiance qu'ils m'ont accordée en m'acceptant dans cette aventure, ensuite pour leur exceptionnelle disponibilité, leur rigueur, leur remarquable encouragements et la richesse de leurs remarques et conseils toujours constructifs. De plus, ils ont été un soutien permanent sur le plan personnel et professionnel. Je veux avec ces quelques mots leur exprimer ma gratitude. Ce travail est aussi le vôtre !

Je tiens aussi à remercier Mme Audine SUBIAS et M. Armand TOGUYENI pour avoir accepté d'être les rapporteurs de ma thèse et pour le temps qu'ils ont consacré, malgré leurs charges et leurs responsabilités, à l'évaluation de mes travaux avec soin. Je remercie également M. Lotfi NABLI pour avoir accepté de présider ma soutenance de thèse et M. Houcine CHAFOUK d'en être l'examineur. Merci pour le temps et l'effort que vous m'avez consacrés et de m'avoir fait l'honneur de faire partie de mon Jury de thèse.

Cette thèse s'est déroulée au sein du Groupe de Recherche en Electrotechnique et Automatique du Havre (GREAH). Je remercie le directeur du GREAH : M. Georges BARAKAT de m'avoir accueilli au sein de son laboratoire. Je tiens à remercier tous mes ami(e)s, collègues, enseignants et personnels du laboratoire. Mes remerciements s'adressent également aux membres du département Génie Mécanique et Productique (GMP) de l'IUT du Havre qui m'ont chaleureusement accueilli au sein de l'équipe enseignante.

Cette thèse a engagé ma famille. Je ne saurais trouver les mots pour remercier suffisamment mes parents, ma sœur, mon oncle et ma tante. Depuis toujours, ils m'accordent leur confiance, me prouvent leur soutien absolu et leur disponibilité sans faille. Leurs amour et présence inconditionnels m'ont permis de venir à bout de ce voyage parsemé de difficultés et d'imprévus. Pour toute l'attente qui a rythmé leur vie depuis 3 ans, je leur dédie cette thèse et espère avoir été à la hauteur de la fierté qu'ils ont toujours nourrie à mon égard.

Il me tient aussi très à cœur de remercier mes amies : Safa BELMESDAL, Amal ARFAOUI, Amina BENSALAH, Hajar SLIMANI et Marwa GAM pour avoir été présentes à des moments clés de cette thèse, et pour m'avoir encouragée par des gestes d'amitié et de bienveillance !



---

# Table des matières

<b>Introduction Générale</b> .....	1
<b>Chapitre 1. Modélisation des SED à contraintes temporelles</b> .....	5
1.1. Introduction.....	7
1.2. Modélisation des SED.....	7
1.2.1. Automates à états finis.....	8
A. Automates probabilistes.....	10
B. Automates temporisés.....	11
1.2.2. Réseaux de Petri.....	12
1.3. Sémantiques temporelles et contraintes temporelles.....	18
1.4. Réseaux de Petri temporisés et temporels.....	19
1.5. Réseaux de Petri stochastiques.....	21
1.5.1. Modèles RdPS markoviens.....	24
1.5.2. Modèles RdPS non markoviens.....	25
1.6. Business Process Modeling.....	28
1.7. Transformation des BPM en RdP équivalent.....	29
1.7.1. Les contraintes structurelles.....	29
1.7.2. Les contraintes temporelles.....	31
1.8. Exemple.....	32
1.9. Conclusion.....	36
<b>Chapitre 2. Diagnostic par calcul des probabilités des trajectoires compatibles</b> .....	39
2.1. Introduction.....	41
2.2. Modélisation des défauts.....	41
2.2.1. Réseaux de Petri partiellement mesurés (POPN) et (POTPN).....	42
2.2.2. Défauts structurels et temporels.....	44
A. Défauts structurels.....	44
B. Défauts temporels.....	46
2.3. Etat de l'art.....	48
2.3.1. Diagnostic des RdP non temporisés.....	48
A. Méthodes structurelles.....	49
B. Méthodes basées sur la dynamique du RdP.....	50



2.3.2. Diagnostic des RdP temporels .....	52
2.4. Calcul des trajectoires compatibles .....	55
2.4.1. Trajectoires compatibles non temporisées .....	55
A. Approche par LMI .....	57
B. Approche incrémentale .....	59
2.4.2. Trajectoires compatibles partiellement temporisées.....	64
2.5. Probabilités des trajectoires .....	66
2.6. Probabilités des défauts .....	77
2.6.1. Probabilités des défauts structurels .....	78
2.6.2. Probabilités des défauts temporels .....	79
2.7. Cas d'étude .....	83
2.8. Conclusion.....	87
<b>Chapitre 3. Diagnostic par carte de contrôle.....</b>	<b>89</b>
3.1. Introduction .....	91
3.2. Diagnostic et défauts temporels.....	91
3.2.1. Définitions.....	91
3.2.2. Modélisation des défauts temporels.....	92
A. Hypothèses de travail .....	92
B. Variation des supports des lois des distributions des durées.....	93
3.3. Carte de contrôle .....	95
3.3.1. Cartes de contrôle classiques de Shewhart .....	97
A. La carte de la moyenne / étendue ( $\bar{X}, R$ ) .....	98
B. La carte de la moyenne / écart-type ( $\bar{X}, S$ ) .....	98
C. La carte de la médiane / étendue ( $\tilde{X}, R$ ) .....	98
3.3.2. Carte de contrôle à somme cumulée (CuSum).....	100
3.3.3. Carte de contrôle à Moyenne Mobile pondérée exponentiellement (EWMA).....	101
3.3.4. Carte de contrôle à Moyenne Mobile (MM) .....	102
3.4. Table de signatures et diagnostic .....	106
3.4.1. Conception de la table de signatures .....	106
3.4.2. Détection et localisation des défauts temporels.....	107
3.5. Exemple .....	108
3.6. Conclusion.....	112
<b>Conclusion Générale et Perspectives.....</b>	<b>113</b>

# Liste des Figures

<b>Figure 1.1.</b> Exemple d'un automate à états finis déterministe $A$ .....	9
<b>Figure 1.2.</b> Exemple d'un automate à états finis probabiliste $A_p$ .....	10
<b>Figure 1.3.</b> Exemple d'un automate à états finis temporisé $A_T$ .....	11
<b>Figure 1.4.</b> Exemple simple d'un RdP.....	16
<b>Figure 1.5.</b> Graphe d'atteignabilité du RdP de la figure 1.4.(c).....	17
<b>Figure 1.6.</b> Franchissement d'une transition temporisée.....	19
<b>Figure 1.7.</b> Exemple d'un RdP $T$ -temporel.....	21
<b>Figure 1.8.</b> Distribution exponentielle des durées de tir de transition.....	24
<b>Figure 1.9.</b> Distribution uniforme bornée des durées de tir de transition.....	26
<b>Figure 1.10.</b> Distribution triangulaire des durées de tir de transition.....	26
<b>Figure 1.11.</b> Distribution triangulaire symétrique des durées de tir de transition.....	27
<b>Figure 1.12.</b> Exemple d'un BPM.....	29
<b>Figure 1.13.</b> La transformation des contraintes structurelles.....	30
<b>Figure 1.14.</b> La transformation des contraintes temporelles.....	32
<b>Figure 1.15 a).</b> La contrainte inter-tâche Début / Fin dans le modèle BPM.....	33
<b>Figure 1.15 b).</b> La contrainte inter-tâche Début / Début dans le modèle BPM.....	34
<b>Figure 1.15 c).</b> La dernière partie BPM de ce processus clinique.....	34
<b>Figure 1.16 a).</b> La contrainte inter-tâche Début / Fin dans le modèle RdP.....	35
<b>Figure 1.16 b).</b> Le XOR-Split connecteur dans le modèle RdP.....	35
<b>Figure 1.16 c).</b> La contrainte inter-tâche Début / Début dans le modèle RdP.....	36
<b>Figure 1.16 d).</b> La dernière partie RdP de ce processus clinique.....	36
<b>Figure 2.1.</b> Exemple simple d'un RdP temporisé partiellement mesuré (POTPN).....	46
<b>Figure 2.2.</b> Exemples de densités de probabilité des durées de tir des transitions : A) une densité de probabilité uniforme avec support fini ; B) une densité de probabilité triangulaire symétrique ; C) une densité de probabilité exponentielle tronquée ; D) une densité de probabilité arbitraire.....	47
<b>Figure 2.3.</b> Calcul des trajectoires compatibles avec une trajectoire de mesures donnée $TRo$ .....	63
<b>Figure 2.4.</b> Arbre de détermination de $\Gamma_E^{-1}(TRo)$ .....	70
<b>Figure 2.5.</b> POTPN d'un processus d'impression 3D.....	83
<b>Figure 2.6.</b> Probabilité des trajectoires compatibles et probabilité du défaut structurel $R$ pour $M_I = (1\ 0\ 0\ 0\ 0)^T$ .....	86
<b>Figure 2.7.</b> Probabilité des trajectoires compatibles et probabilité du défaut structurel $R$ pour $M_I = (2\ 0\ 0\ 0\ 0)^T$ .....	86
<b>Figure 2.8.</b> Probabilité que la trajectoire compatible satisfait $TC$ pour $M_I = (1\ 0\ 0\ 0\ 0)^T$ .....	86
<b>Figure 2.9.</b> Probabilité du défaut temporel pour $M_I = (1\ 0\ 0\ 0\ 0)^T$ .....	86
<b>Figure 2.10.</b> Probabilité que la trajectoire compatible satisfait $TC$ pour $M_I = (2\ 0\ 0\ 0\ 0)^T$ .....	87
<b>Figure 2.11.</b> Probabilité du défaut temporel pour $M_I = (2\ 0\ 0\ 0\ 0)^T$ .....	87
<b>Figure 3.1.</b> Variation de support de la loi : A) triangulaire symétrique ; B) uniforme bornée.....	93
<b>Figure 3.2.</b> Image d'une carte de contrôle.....	96
<b>Figure 3.3.</b> Carte de contrôle EWMA.....	102
<b>Figure 3.4.</b> Utilisation d'une carte de contrôle MM.....	103
<b>Figure 3.5.</b> Exemple de RdPSTPO.....	104
<b>Figure 3.6.</b> La variation de la moyenne des durées de franchissement de $T_2$ avec une densité de probabilité uniforme bornée.....	105

---

<b>Figure 3.7.</b> La variation de l'écart-type des durées de franchissement de $T_2$ avec une densité de probabilité uniforme bornée .....	105
<b>Figure 3.8.</b> Exemple de RdPSTPO .....	109
<b>Figure 3.9 a).</b> La carte de contrôle MM appliquée sur : $d(T_3, T_1)$ .....	110
<b>Figure 3.9 b).</b> La carte de contrôle MM appliquée sur : $d(T_3, T_4)$ .....	110
<b>Figure 3.9 c).</b> La carte de contrôle MM appliquée sur : $d(T_1, T_3)$ .....	110
<b>Figure 3.9 d).</b> La carte de contrôle MM appliquée sur : $d(T_1, T_4)$ .....	110
<b>Figure 3.9 e).</b> La carte de contrôle MM appliquée sur : $d(T_1, T_1)$ .....	111
<b>Figure 3.9 f).</b> La carte de contrôle MM appliquée sur : $d(T_4, T_3)$ .....	111
<b>Figure 3.9 g).</b> La carte de contrôle MM appliquée sur : $d(T_4, T_1)$ .....	111
<b>Figure 3.9 h).</b> La carte de contrôle MM appliquée sur : $d(T_4, T_4)$ .....	111

---

# Liste des Tableaux

<b>Tableau 2.1.</b> Les trajectoires de l'ensemble étendu $\Gamma_E^{-1}(TRo)$ .....	71
<b>Tableau 2.2.</b> Probabilité des trajectoires compatibles.....	79
<b>Tableau 2.3.</b> La signification des opérations .....	85
<b>Tableau 3.1.</b> Tableau MSP de $m \times n$ mesures $X_{i,j}$ .....	96
<b>Tableau 3.2.</b> Tableau des Coefficients intervenant dans le calcul des limites de contrôle des cartes de Shewhart.....	99
<b>Tableau 3.3.</b> La table de signatures de l'exemple de la figure 3.8.....	112



---

# Abréviations

ASCG	Atomic States Class Graph - Graphe des classes d'états atomiques
BPM	Business Process Modeling - Modélisation de processus métier
BRD	Basis Reachability Diagnoser - Diagnostiqueur d'accessibilité de base
BRG	Basis Reachability Graph - Graphe d'accessibilité de base
CuSum	Cumulative Sum - Somme cumulée
EFS	Elementary Firing Sequence - Séquence de tirs élémentaire
EOS	Elementary Observation Sequence - Séquence de mesure élémentaire
EWMA	Exponentially Weighted Moving Average - Moyenne mobile pondérée exponentiellement
FDG	Fault Diagnosis Graph - Graphe de diagnostic des fautes
GREAH	Groupe de Recherche en Electrotechnique et Automatique du Havre
ILP	Integer Linear Programming - Programmation linéaire en nombres entiers
LMI	Linear Matrix Inequality – Inégalité matricielle linéaire
MBRG	Modified Basis Reachability Graph - Graphe d'accessibilité de base modifié
MM	Moving Mean - Moyenne Mobile
MSCG	Modified State Class Graph - Graphe des classes modifié
MSP	Maîtrise Statistique des Procédés
PDF	Probability Density Function - Fonction de densité de probabilité
POTPN	Partially Observed Timed Petri Net - Réseau de Petri temporisé partiellement observé
RdP	Réseau de Petri
RdPS	Réseau de Petri Stochastique
RdPSG	Réseau de Petri Stochastique Généralisé
RdPSTPO	Réseau de Petri Stochastiques Temporisés Partiellement Observé
RG	Reachability Graph - Graphe d'accessibilité
SCG	State Class Graph - Graphe des classes d'états
SED	Systèmes à événements discrets
SSD	Semi Symbolic Diagnoser - Diagnostiqueur semi symbolique
SPC	Statistical Process Control - Contrôle statistique des processus

---

TC            Temporel Constraints - Contraintes temporelles  
VA            Variable Aléatoire

---

# Introduction Générale

---

L'environnement industriel connaît depuis plusieurs années une évolution très rapide due à un progrès technologique accéléré contribuant à un volume élevé des flux de données à transmettre, analyser et stocker. Cette évolution exige de nouvelles compétences pour la gestion et le contrôle des systèmes automatisés. Afin de s'adapter à cet environnement et survivre au sein d'une concurrence très vive dans la plupart des secteurs et dans des domaines variés, il est impératif de fixer certaines mesures pour assurer une amélioration continue des systèmes automatisés, notamment :

- une adaptation aux perturbations non souhaitées,
- une meilleure estimation de l'état courant du système,
- une surveillance accrue des écarts avec les plans prévisionnels,
- une surveillance permanente de l'état de santé du système afin de garantir sa disponibilité, sa sécurité ainsi que le maintien de ses propriétés dans le temps.

Ces démarches relèvent du domaine général de la sûreté de fonctionnement [Villemeur 1988], permettant de faire face à la complexité croissante des systèmes automatisés et de limiter les comportements indésirables qui peuvent remettre en cause l'intégralité des systèmes.

L'accroissement du nombre d'informations et des données disponibles se traduit le plus souvent par un ralentissement des traitements et quelquefois par leur saturation. Ces deux phénomènes sont inacceptables dans les systèmes temps-réels qui doivent respecter des contraintes temporelles. Ces systèmes sont présents dans de nombreux domaines tels que : les systèmes de production automatisés, de communication, d'information, de pilotage embarqué (automobile, avionique, ...), de gestion ...etc. Pour de nombreux systèmes, le non-respect des contraintes temporelles est critique et s'apparente à un défaut qu'il faut détecter, diagnostiquer puis circonscrire.

Notre étude s'articule autour de la surveillance des systèmes temps-réels, afin de détecter la violation des contraintes temporelles et donc garantir leur bon fonctionnement dans le temps. Nous nous plaçons dans un contexte temporel, et nous nous intéressons à la catégorie des Systèmes à Événements Discrets (SED). Les



SED sont des systèmes dynamiques pour lesquels les variables d'états évoluent de manière discrète au cours du temps. Cette évolution est conditionnée par l'occurrence d'événements asynchrones [Ramadge et Wonham 1989 ; Cassandras et Lafortune 2009]. Les SED couvrent aujourd'hui une classe importante d'applications réelles, notamment les processus manufacturiers, les protocoles de communication, les réseaux de télécommunications, les services informatiques ainsi que les réseaux de transport et les systèmes logistiques.

Assurer la sûreté de fonctionnement des SED dans le cadre de notre travail revient à diagnostiquer ces systèmes afin de détecter des défauts temporels. Le but du diagnostic d'un SED [Zaytoon et Lafortune 2013] est de détecter au plus tôt une déviation du comportement nominal du système et d'en déterminer les causes.

L'occurrence d'un défaut conduit ce dernier à un comportement fautif et donc à une déviation de son comportement nominal. Le système devient alors incapable de réaliser le service pour lequel il a été conçu, on dit que le système est défaillant. Les tâches du diagnostic permettent entre autre de déterminer les causes premières des défaillances en se basant sur une série de mesures issues des capteurs. L'enjeu du diagnostic repose donc sur la capacité à traiter et exploiter les mesures et les données disponibles pour surveiller l'état du système.

Dans cette thèse, nous nous intéressons aux défauts temporels qui correspondent à une violation de contraintes temporelles. Le verrou scientifique auquel nous nous intéressons est de réaliser cette détection avec un volume d'information réduit et incomplet. L'information disponible est mesurée au fil du temps et modélisée par une trame datée. En utilisant les densités de probabilité des événements non mesurés et la spécification des contraintes temporelles, et en analysant les trames mesurées, nous proposons d'estimer la probabilité que le système respecte ou non les contraintes temporelles. A cette fin, il est nécessaire de disposer d'un outil de modélisation formel permettant d'analyser les SED, de les contrôler et de les simuler. Parmi les outils existants, cette thèse utilise en particulier les réseaux de Petri (RdP).

L'outil de modélisation RdP a été introduit par le mathématicien allemand Carl Adam Petri dans sa thèse de doctorat [Petri 1966]. Depuis, cet outil est devenu un formalisme mathématique et graphique puissant et universellement utilisé dans la modélisation et l'analyse du comportement des SED. Le RdP possède la capacité de représenter explicitement certains comportements du système dont : la synchronisation, le partage de ressources, le parallélisme et encore les structures de choix. Afin de décrire explicitement les comportements temporels et aléatoires notre thèse utilise une extension particulière de RdP : les RdP Stochastiques Temporisés Partiellement Observés (RdPSTPO). L'extension stochastique temporisée permet

---

de caractériser l'incertitude des dates d'occurrence des événements qui sont modélisées par des Variables Aléatoires (VA).

La contribution principale de la thèse est l'exploitation au mieux des informations temporelles disponibles pour le diagnostic des défauts dans les SED avec des contraintes temporelles. Dans une première phase, nous nous focalisons sur l'analyse des mesures datées dans les SED partiellement observés par des capteurs hétérogènes. Dans une deuxième phase, nous évaluons les probabilités de violation des contraintes temporelles (défauts temporels) pour les trames mesurées. Finalement, nous proposons une méthode de diagnostic à base de carte de contrôle Moyenne Mobile (MM) pour les dérives lentes des systèmes cycliques. La carte de contrôle est une technique statistique importante utilisée dans le contrôle statistique des processus (Statistical Process Control (SPC)) qui permet d'améliorer la qualité et la fiabilité des produits par une plus grande maîtrise du procédé [Shewhart 1931]. Certains systèmes étudiés sont susceptibles d'être affectés par de faibles variations temporelles. Une analyse moyenne a donc été menée afin de détecter ces variations temporelles non détectables ponctuellement. Enfin, la localisation du défaut temporel parmi plusieurs candidats est réalisée grâce à une table de signatures.

Le mémoire de thèse est organisé en trois chapitres qui regroupent nos contributions et les accompagnent par un état de l'art relatif à la problématique et aux modèles utilisés dans chaque chapitre. Le contenu de chaque chapitre est résumé ci-dessous :

**Le premier chapitre** introduit les deux formalismes les plus connus pour modéliser et analyser des SED, à savoir : les automates à états finis et les RdP. Il présente aussi les définitions et les concepts fondamentaux relatifs à ces outils de modélisation, leurs extensions ainsi que les notations et principe d'évolution sous la sémantique retenue dans nos travaux de recherche. Par la suite, grâce à l'outil de modélisation de processus métier (Business Process Modeling (BPM)), connu pour sa capacité à traiter des systèmes d'information complexes comportant des contraintes de temps entre les tâches, nous avons pu introduire de nouvelles contraintes temporelles complexes dans les RdP. Les règles de base permettant de transformer un processus BPM en un RdP équivalent sont détaillées dans ce chapitre. Un exemple d'étude de cas est donné pour illustrer la transformation des différentes contraintes temporelles.

---

**Le deuxième chapitre** établit un état de l'art sur le diagnostic des RdP non temporisés, les RdP temporels ainsi que les extensions de RdP temporisés. Il présente les défauts (structurels et temporels) auxquels notre travail s'est intéressé. Il présente aussi d'une manière détaillée les deux méthodes utilisées pour déterminer les comportements compatibles avec les mesures : résolution d'un ensemble d'inégalités matricielles linéaires (Linear Matrix Inequality (LMI)) et résolution incrémentale. Le calcul de la probabilité des trajectoires de mesures puis de la probabilité de chaque type de défaut est ensuite expliqué. Finalement, un exemple est présenté pour illustrer les différentes méthodes.

**Le troisième chapitre** est dédié au diagnostic des défauts temporels par l'utilisation de cartes de contrôle MM. Il détaille dans un premier temps les différents types de cartes de contrôle utilisées dans le SPC. Une modélisation des dérives temporelles à partir des variations de la moyenne des durées de franchissement des transitions est proposée. Ce chapitre montre aussi l'intérêt du diagnostic par carte de contrôle MM en terme de complexité : l'implémentation est simple et rapide en comparaison de l'approche présentée dans le deuxième chapitre. Finalement, une table de signatures permettant d'isoler le défaut parmi plusieurs candidats est détaillée avec un exemple de cas d'étude.

Enfin, le mémoire se termine avec une conclusion générale qui résume les résultats ainsi que les perspectives de poursuite de ces travaux de recherche.

---

# Chapitre 1. Modélisation des SED à contraintes temporelles

---

1.1. Introduction .....	7
1.2. Modélisation des SED .....	7
1.2.1. Automates à états finis .....	8
A. Automates probabilistes.....	10
B. Automates temporisés .....	11
1.2.2. Réseaux de Petri .....	12
1.3. Sémantiques temporelles et contraintes temporelles .....	18
1.4. Réseaux de Petri temporisés et temporels.....	19
1.5. Réseaux de Petri stochastiques .....	21
1.5.1. Modèles RdPS markoviens.....	24
1.5.2. Modèles RdPS non markoviens .....	25
1.6. Business Process Modeling.....	28
1.7. Transformation des BPM en RdP équivalent.....	29
1.7.1. Les contraintes structurelles.....	29
1.7.2. Les contraintes temporelles .....	31
1.8. Exemple .....	32
1.9. Conclusion.....	36



## 1.1. Introduction

Au lieu de s'intéresser au déroulement continu des phénomènes, les Systèmes à Événements Discrets (SED) ne se soucient que des débuts et des fins de ces phénomènes (les événements discrets) ainsi que de leurs enchaînements logiques, dynamiques et temporels. Les SED sont des systèmes dynamiques pour lesquels l'espace d'état est discret. Leur évolution se fait conformément aux occurrences des événements caractérisant les changements d'état du système [Ramadge et Wonham 1989 ; Cassandras et Lafortune 2009]. Ces systèmes couvrent aujourd'hui une classe importante d'applications réelles, notamment les processus manufacturiers, les systèmes informatiques, logistiques et de transport ainsi que les réseaux de télécommunications. De nombreux formalismes de modélisation ont été développés pour analyser les SED. La complexité croissante de ces derniers nécessite des outils de modélisation enrichis dans le but de mieux représenter les différents comportements et phénomènes liés à ses systèmes.

Dans ce chapitre, nous introduisons deux formalismes largement utilisés pour l'analyse des SED, à savoir les automates à états finis et les Réseaux de Petri (RdP). Ces formalismes ont été enrichis par la suite afin d'analyser des séquences incomplètes d'information dans un cadre temporel et stochastique. Deux extensions de RdP ont été exploitées dans ce travail : les réseaux de Petri  $T$ -temporels et les réseaux de Petri stochastiques (RdPS) markoviens et non markoviens. Dans le cadre du diagnostic, la modélisation et l'analyse des séquences temporisées d'événements discrets constituent un défi majeur. Lorsque les mesures sont incomplètes, le problème devient critique, en particulier lorsqu'un raisonnement sur des intervalles de temps est requis. Afin de faire face à une telle situation, une modélisation systématique des contraintes temporelles complexes est nécessaire. Grâce à l'outil de modélisation de processus métier (Business Process Modeling (BPM)), connu pour sa capacité de traiter des systèmes d'information complexes comportant des contraintes de temps entre les tâches, nous avons pu introduire de nouvelles contraintes temporelles complexes dans les RdP. Les règles de base permettant de transformer un processus BPM en un RdP équivalent sont détaillées dans ce chapitre. Enfin, les différentes contraintes temporelles, en particulier, les contraintes inter-tâches sont introduites dans le modèle RdP. De ce fait, cette extension de RdP est utile pour prendre en compte les informations temporelles dans le processus de diagnostic et pour résoudre des problèmes de diagnostic dans plusieurs domaines d'étude tels que la logistique, la chaîne d'approvisionnement ou l'automatisation.

## 1.2. Modélisation des SED

Plusieurs formalismes de modélisation ont été développés pour l'analyse des SED selon l'application envisagée. Une grande variété de travaux de recherche est disponible dans la littérature. Ces travaux peuvent

être classés selon l'outil de modélisation utilisé. Dans ce qui suit, nous allons présenter brièvement le formalisme des automates à états finis. Ceci nous permettra d'illustrer le principe de quelques approches de diagnostic basées sur ce modèle. Ensuite, nous accordons une attention particulière à l'outil de modélisation RdP sur lequel seront basées nos approches.

### 1.2.1. Automates à états finis

Les automates à états finis ou automates finis, sont des machines à états qui permettent de décrire le comportement d'un SED. Ils sont représentés par une succession d'états et de transitions associées à des événements. L'évolution des automates finis est assurée par l'occurrence de ces événements. Un ensemble de cercles (qui représentent les états) et un ensemble d'arcs (qui représentent les transitions) forment les différents éléments graphiques de cet automate [Shallit 2008 ; Cassandras et Lafortune 2009]. Avant de représenter formellement les automates à états finis, il est intéressant de mentionner que de nombreux travaux ont été élaborés pour développer ce formalisme en s'appuyant sur la théorie des langages [Kleene 1951 ; Moore 1956 ; Burks et Wang 1957].

Un automate à états finis déterministe  $A$ , peut être défini de la façon suivante [Shallit 2008 ; Cassandras et Lafortune 2009] :

**Définition 1.1.** Un automate à états finis  $A$  est un quadruplet  $A = \langle \Sigma, X, \delta, x_0 \rangle$  où :

- $\Sigma$  un ensemble fini de symboles appelé alphabet décrivant l'ensemble des événements
- $X$  un ensemble fini d'états qui constitue l'espace discret d'états
- $\delta : X \times \Sigma \rightarrow X$  la fonction de transitions entre les états
- $x_0 \in X$  l'état initial désigné par une flèche entrante sur le graphe

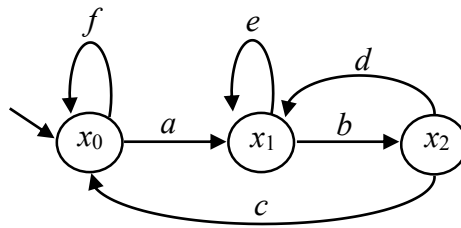
L'automate  $A$  est dit déterministe dans le sens où l'état initial est considéré unique et que depuis tout état, il n'existe pas deux transitions de sortie qui soient associées à un même événement et qui conduisent à deux états différents. Un mot (ou chaîne ou séquence) noté  $s$  défini sur un alphabet  $\Sigma$  est une suite finie et ordonnée d'événements. La longueur d'un mot est notée  $|s|$  et représente le nombre d'événements de  $s$ . Etant donné un alphabet  $\Sigma$ ,  $\Sigma^n$  représente l'ensemble des mots sur  $\Sigma$  de longueur  $n$  avec  $n$  entier naturel. Un mot qui ne comporte aucun événement est appelé mot vide et est noté  $\varepsilon$  tel que  $\Sigma^0 = \{\varepsilon\}$ . Par extension, l'ensemble des

mots de longueur  $n$  quelconque construit sur  $\Sigma$  est noté par  $\Sigma^*$  où « \* » représente l'Etoile de Kleene. Ainsi  $\Sigma^*$  peut être défini par :

$$\Sigma^* = \bigcup_{i \geq 0} \Sigma^i \quad (1.1)$$

avec  $i$  entier naturel.

L'ensemble des mots réalisables par un automate  $A$  correspond au langage  $L(A)$  engendré par cet automate. On note que tout langage  $L$  défini sur un alphabet  $\Sigma$  est un sous-ensemble de  $\Sigma^*$ .



**Figure 1.1.** Exemple d'un automate à états finis déterministe  $A$

**Exemple 1.1.** Dans l'exemple de la figure 1.1 l'ensemble des états de l'automate  $A$  est  $X = \{x_0, x_1, x_2\}$ . L'état initial  $x_0$  est indiqué par une flèche entrante. La fonction de transition d'états  $\delta$  est représentée par des arcs associés à des événements de l'alphabet  $\Sigma$ . Dans notre exemple, l'alphabet correspond à l'ensemble d'événements suivant :  $\Sigma = \{a, b, c, d, e, f\}$ . Si l'on considère la séquence d'événements  $abca$ , le franchissement de cette dernière à partir de l'état initial  $x_0$  fait que le modèle évolue de l'état  $x_0$  vers l'état  $x_1$  on note :  $\delta = (x_0, abca) = x_1$ . L'ensemble de toutes les séquences possibles d'événements dans le système est donné par le langage  $L(A)$ . Ce langage représente des mots auxquels correspondent des chemins dans  $A$  :  $L(A) = \{\varepsilon, f, fa, a, ab, abc, abd, abde, abca, abcab, \dots\}$ .

Dans l'étude et l'analyse des SED, l'une des questions est de savoir si le système est observable ou non. Afin de répondre à cette question [Ramadge et Wonham 1989], il est nécessaire de partitionner, a priori, l'ensemble des événements d'un SED modélisé par un automate à états finis en deux sous-ensembles  $\Sigma_o$  et  $\Sigma_u$  qui sont respectivement l'ensemble des événements mesurables et l'ensemble des événements silencieux (incluant les événements qui représentent des défauts) avec  $\Sigma_o \cap \Sigma_u = \emptyset$ .

Il existe de nombreuses extensions des automates à états finis à savoir les automates probabilistes et temporisés. Dans un premier temps, nous allons brièvement présenter l'extension probabiliste de ces

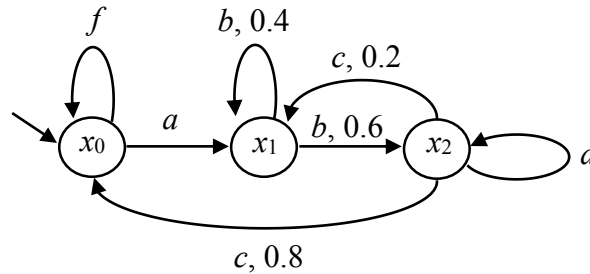


automates. Ensuite, nous définissons les automates temporisés qui sont introduits pour analyser les SED à contraintes temporelles.

### A. Automates probabilistes

Le formalisme des automates probabilistes a été introduit par [Rabin 1963]. Cette extension des automates à états finis représente une généralisation des automates finis non déterministes où une probabilité d'occurrence est associée à chaque événement. Ce type d'automates a été introduit dans le but d'analyser les SED probabilistes dans lesquels les fréquences d'exécution des actions sont connues. Les transitions dans un automate probabiliste sont représentées de manière compacte par une matrice stochastique. Le langage engendré par ce type d'automates est appelé langage stochastique. Comme pour un automate à états finis usuel, un automate à états finis probabiliste  $A_P$  est un quadruplet :  $A_P = \langle \Sigma, X, \delta_P, \pi_0 \rangle$  composé comme suit :

- $\Sigma$  un ensemble d'événements appelé alphabet
- $X$  un ensemble fini d'états qui constitue l'espace discret d'états
- $\delta_P: X \times \Sigma \times X \rightarrow [0,1]$  la fonction des probabilités de transition entre les états telle que  $\delta_P(x_i, \alpha, x_j)$  détermine la probabilité de passer de l'état  $x_i$  vers l'état  $x_j$  suite à l'occurrence de l'événement  $\alpha$
- $\pi_0: X \rightarrow [0,1]$  la distribution initiale ( $\pi_0(x_j)$  est la probabilité que  $x_j$  soit l'état initial)



**Figure 1.2.** Exemple d'un automate à états finis probabiliste  $A_P$

On note que pour chaque  $x_i \in X$  et pour chaque  $\alpha \in \Sigma$  :

$$\sum_{x_j \in X} \delta_P(x_i, \alpha, x_j) = 1 \quad (1.2)$$

Considérons l'automate  $A_P$  de la figure 1.2. La distribution initiale est telle que la probabilité initiale est égale à 1 pour l'état initial  $x_0$ , et nulle pour les états  $x_1$  et  $x_2$  ( $\pi_0(x_0) = 1$  et  $\pi_0(x_j) = 0$  pour  $j = 1,2$ ). A chaque transition est associée une probabilité. Par exemple,  $\delta_P(x_2, c, x_0) = 0.8$  est la probabilité de transiter de l'état  $x_2$  vers l'état  $x_0$  suite à l'occurrence de l'événement  $c$ .

## B. Automates temporisés

Les automates temporisés [Alur et Dill 1994] sont une extension des automates à états finis auxquels sont ajoutées des contraintes temporelles. Ce formalisme a été introduit pour analyser les SED temporisés. Les contraintes temporelles sont représentées par des inéquations mettant en jeu plusieurs horloges, soit  $\mathcal{C}(\mathcal{H})$  cet ensemble des contraintes. Celles-ci évoluent d'une manière continue avec le temps et peuvent être indépendamment réinitialisées suite à l'occurrence d'un événement donné. Une horloge mesure donc le délai séparant sa dernière mise à zéro et l'occurrence d'un événement. Deux types de contraintes sont représentées dans cet automate : celles associées aux états et appelées invariants, et celles associées aux transitions et appelées gardes. Formellement, un automate à états finis temporisé  $A_T$  est défini comme suit :

**Définition 1.2.** Un automate à états finis temporisé  $A_T$  est un 6-uplet  $A_T = \langle \Sigma, X, \mathcal{H}, Inv, \delta_T, x_0 \rangle$  où :

- $\Sigma$  est l'ensemble des événements appelé alphabet
- $X$  un ensemble fini d'états qui constitue l'espace discret d'états
- $\mathcal{H}$  est un ensemble fini d'horloges qui sont incrémentées de façon synchrone
- $\delta_T : X \times \mathcal{C}(\mathcal{H}) \times \Sigma \times 2^X \times X$  est un ensemble fini de transitions, une transition entre l'état  $x$  et  $x'$  est de la forme  $(x, ct, \alpha, \beta, x')$  avec  $ct \in \mathcal{C}(\mathcal{H})$ ,  $\alpha$  est l'événement produit entre  $x$  et  $x'$  et  $\beta$  est l'ensemble des horloges remises à zéro
- $Inv : X \rightarrow \mathcal{C}(\mathcal{H})$  associe un invariant à chaque état
- $x_0 \in X$  est l'état initial

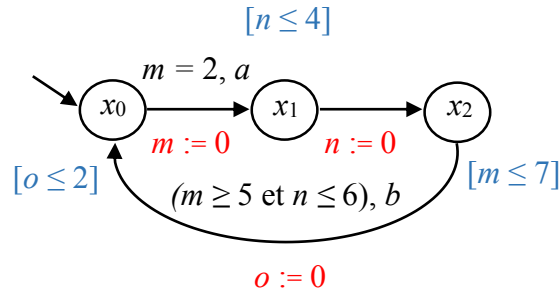


Figure 1.3. Exemple d'un automate à états finis temporisé  $A_T$

**Exemple 1.2.** L'automate temporisé  $A_T$  de la figure 1.3 est caractérisé par trois horloges  $m$ ,  $n$  et  $o$ . A l'état  $x_0$ , les trois horloges sont à zéro ( $m := 0$ ,  $n := 0$  et  $o := 0$ ). La garde  $m = 2$  signifie que la transition vers  $x_1$  peut avoir lieu seulement lorsque l'horloge  $m$  est égale à 2 Unités de Temps (UT) et que l'événement  $a$  se produit, l'horloge  $m$  est alors remise à zéro et l'automate évolue de l'état  $x_0$  vers  $x_1$ . La valeur de l'horloge  $n$  vaut 2 car  $n$  n'a pas été réinitialisée. L'invariant  $n \leq 4$  signifie que l'horloge  $n$  peut avoir une valeur inférieure ou égale à 4 lorsque le système est dans l'état associé à cet invariant, ainsi le modèle doit obligatoirement quitter cet état lorsque  $n > 4$ . La transition de l'état  $x_2$  vers l'état initial  $x_0$  nécessite l'occurrence de l'événement  $b$ , ce dernier doit se produire lorsque la valeur de l'horloge  $m$  dépasse 5 et celle de  $n$  ne dépasse pas 6. Dans le cas où une contrainte n'est pas respectée, l'automate se trouve bloqué.

Les automates à états finis représentent un puissant formalisme pour la modélisation des SED, néanmoins leurs avantages sont atténués par l'explosion combinatoire du nombre d'états. La modélisation par des RdP, quant à elle, permet de s'affranchir de cet inconvénient en offrant des structures de modélisation beaucoup plus riches et des extensions plus adaptées pour modéliser des systèmes complexes.

### 1.2.2. Réseaux de Petri

Les (RdP) sont des formalismes puissants pour la modélisation des SED. La complexité croissante de ces derniers nécessite des méthodes de représentation permettant de reproduire le plus fidèlement possible les différents comportements et phénomènes liés au système. Plusieurs raisons motivent l'utilisation des RdP dans un cadre théorique simple et puissant pour étudier les problèmes liés aux SED, à savoir :

- Les RdP possèdent un formalisme mathématique et graphique permettant de modéliser les SED. Cet aspect graphique clarifie et synthétise la représentation du système alors que l'aspect mathématique permet de poser les équations d'état du système et d'analyser ce dernier soit par l'algèbre linéaire soit par la théorie des graphes.
- Les RdP donnent une représentation compacte de l'espace d'état. En fait, ils ne nécessitent pas de représenter explicitement tous les états accessibles possibles, mais uniquement les règles d'évolution.
- Les RdP peuvent représenter le concept de simultanéité et modéliser facilement les phénomènes de synchronisation.

- Les RdP peuvent représenter un SED avec un nombre infini d'états tout en utilisant un graphe avec un nombre fini de sommets.

Les RdP ont été introduits par Carl Adam Petri en 1962 [Petri 1962] dans sa thèse de doctorat. Depuis, ils sont largement utilisés pour l'analyse et la modélisation d'une classe importante de systèmes [Girault et Valk 2013]. Grâce à la facilité de modélisation des phénomènes de concurrence, synchronisation ou encore de partage de ressources, l'outil RdP est ainsi appliqué pour modéliser une grande variété de systèmes tels que : les systèmes de communication [Billington et diaz 1999], manufacturiers [DiCesare et al. 1993], informatiques [Reisig 2013] ou encore biologiques et chimiques [Barilla et al. 1963].

De manière générale, un RdP est un graphe biparti orienté composé de deux types de sommets : des places et des transitions. Les places, associées à des états du système, sont représentées par des cercles contenant des marques. Les marques communément appelées jetons constituent des objets discrets dont la répartition sur le réseau représente le marquage de ce dernier. Les transitions sont représentées par des barres et permettent de décrire les changements d'état. Les places et les transitions sont connectées entre elles par des arcs. Un arc ne peut pas relier deux sommets de même type. Les places, transitions et arcs permettent de représenter graphiquement la partie statique du réseau. Afin de simuler le comportement du système, une structure dynamique est superposée à cette structure statique. Cette structure correspond au marquage du réseau, évoluant dynamiquement en fonction des tirs des transitions. Pour plus de détails on pourra se reporter à [Murata 1989 ; Bobbio 1990 ; Giua et Seatzu 2007 ; Cassandras et lafortune 2009].

Les RdP sont formellement définis comme suit :

**Définition 1.3.** Un RdP, aussi appelé réseau de places / transitions, est une structure  $G = \langle \mathcal{P}, \mathcal{T}, W_{PR}, W_{PO} \rangle$  où :

- $\mathcal{P} = \{P_1, \dots, P_n\}$  est un ensemble fini et non vide de  $n$  places
- $\mathcal{T} = \{T_1, \dots, T_q\}$  est un ensemble fini et non vide de  $q$  transitions
- $W_{PR} : \mathcal{P} \times \mathcal{T} \rightarrow \mathbb{N}^{n \times q}$  est la matrice d'incidence avant comportant les poids des arcs allant des places vers les transitions, avec  $\mathbb{N}$  l'ensemble des entiers naturels
- $W_{PO} : \mathcal{P} \times \mathcal{T} \rightarrow \mathbb{N}^{n \times q}$  est la matrice d'incidence arrière comportant les poids des arcs allant des transitions vers les places.

La matrice d'incidence  $W$  est définie par :

$$W = W_{PO} - W_{PR} \quad (W \in \mathbb{N}^{n \times q}) \quad (1.3)$$

L'état d'un RdP est décrit par son vecteur de marquage  $M = (m_k) \in (\mathbb{N}^n)$  tel que  $m_k$  représente le nombre de jetons dans la place  $P_k$ . Ce dernier est modifié selon les règles de tir suivantes :

- Une transition est dite validée (ou sensibilisée) par un marquage  $M$  si et seulement si  $M \geq W_{PR}(:, j)$ , où  $W_{PR}(:, j)$  représente la  $j^{\text{ème}}$  colonne de la matrice  $W_{PR}$ . Lorsqu'une transition est sensibilisée à partir d'un marquage  $M$  elle peut être franchie avec un nombre fini de tirs simultanés donné par son degré de validation  $n_j(M)$ . Ce dernier est calculé comme suit :

$$n_j(M) = \min_{P_k \in {}^{\circ}T_j} \left( \left\lfloor \frac{M(P_k)}{W_{PR}(k, j)} \right\rfloor \right) \quad (1.4)$$

où le preset de  $T_j$  ( ${}^{\circ}T_j$ ) désigne les places en amont de  $T_j$ , et les crochets  $\lfloor \ ]$  représentent la partie entière du nombre à l'intérieur des crochets.

- A partir du marquage  $M$ , une des transitions validées  $T_j$  (on note :  $M[T_j >$ ) peut être tirée (ou franchie), ceci entraîne un nouveau marquage  $M'$ , on note :  $M[T_j > M'$ .
- Le tir d'une transition validée  $T_j$  supprime un ou plusieurs jetons (selon le poids des arcs entrant dans la transition) de chaque place d'entrée de  $T_j$ , et rajoute un ou plusieurs jetons (selon le poids des arcs sortant de la transition) à chaque place de sortie de  $T_j$ . On note que, lorsqu'un élément de  $W_{PR}$  (respectivement de  $W_{PO}$ ) est supérieur strictement à 1, un poids est indiqué sur l'arc correspondant sur le graphe. Lorsqu'un poids est égal à 1, aucune indication n'est reportée sur l'arc. On appelle « réseau ordinaire » un réseau dont tous les poids des arcs en amont et en aval de chaque transition sont égaux à 1. Dans le cas où les spécifications des arcs ne sont pas égales à 1, le réseau est appelé « réseau généralisé ».

Soit  $T(j) \in \mathcal{T}$  pour  $j = 1, \dots, h$  une suite de transitions pouvant être tirées successivement à partir d'un marquage  $M$  tel que  $h$  est la longueur de cette séquence de transitions. On définit :

$$\sigma_U = T(1) T(2) \dots T(h) \quad (1.5)$$

la séquence de franchissement de ces transitions à partir du marquage  $M$  et on note alors  $M[\sigma_U >$ . On parle d'une trajectoire  $(\sigma_U, M)$  lorsque le marquage initial  $M$  et les marquages intermédiaires sont spécifiés :

$$(\sigma_U, M) = M(0) [T(1) > M(1) \dots [T(h) > M(h) \quad \text{avec } M(0) = M \quad (1.6)$$

Suite à l'occurrence d'une trajectoire  $(\sigma_U, M)$  un nouveau marquage  $M'$  est obtenu. Ce dernier peut être calculé mathématiquement par l'équation suivante :

$$M' = M + W \cdot X(\sigma_U) \quad (1.7)$$

avec  $W$  la matrice d'incidence du réseau et  $X(\sigma_U) \in \mathbb{N}^q$  le vecteur de franchissements de la séquence  $\sigma_U$ . Ce vecteur de franchissements est un vecteur colonne de dimension  $q$  (nombre de transitions). Chaque composante  $x_j$  de ce vecteur représente le nombre de fois que la transition  $T_j$  est tirée dans  $\sigma_U$ . La représentation de la séquence  $\sigma_U$  par son vecteur de franchissement  $X(\sigma_U)$  est compacte mais ne donne aucune information sur l'ordre de franchissement des transitions de cette séquence.

Un RdP marqué est défini par le couple  $\langle G, M_I \rangle$  où  $G = \langle \mathcal{P}, \mathcal{T}, W_{PR}, W_{PO} \rangle$  représente un RdP avec un marquage initial  $M_I$ . Un marquage  $M'$  est atteignable à partir de  $M$  s'il existe une séquence de franchissements  $\sigma_U$  telle que  $M [\sigma_U > M'$ . L'ensemble  $ReachSet(G, M_I)$  constitue l'ensemble des marquages atteignables à partir de  $M_I$ . Quel que soit le marquage atteint à partir de  $M_I$ , si le nombre de jetons dans chaque place est borné par un entier naturel  $k$ , le réseau marqué est dit  $k$ -borné et par conséquent, l'espace d'état du RdP est fini. Le réseau possède une infinité d'états différents dans le cas où le nombre de jetons dans l'une ou plusieurs de ses places n'est pas borné, un tel réseau est dit « réseau non borné ».

Dans le cas d'un réseau borné, tous les marquages atteignables  $ReachSet(G, M_I)$  à partir d'un marquage initial ainsi que les transitions dont le franchissement entraîne le passage d'un marquage à un autre sont regroupés dans un graphe appelé « graphe d'atteignabilité » (ou « graphe de marquages ») [Campos et al. 1989 ; René et Alla 2010]. Chaque sommet du graphe correspond à un marquage atteignable et chaque arc correspond au franchissement d'une transition validée. La figure 1.5 représente le graphe d'atteignabilité du RdP de la figure 1.4.(c). Ce réseau possède trois marquages atteignables à partir du marquage initial  $M_I = (3 \ 1 \ 1 \ 0)^T$  qui sont  $M_1 = (1 \ 0 \ 1 \ 1)^T$  suite au franchissement de  $T_1$ ,  $M_2 = (3 \ 1 \ 0 \ 1)^T$  après le franchissement de  $T_2$  et  $M_3 = (1 \ 0 \ 0 \ 2)^T$  après le franchissement de  $T_1T_2$  ou de  $T_2T_1$ .

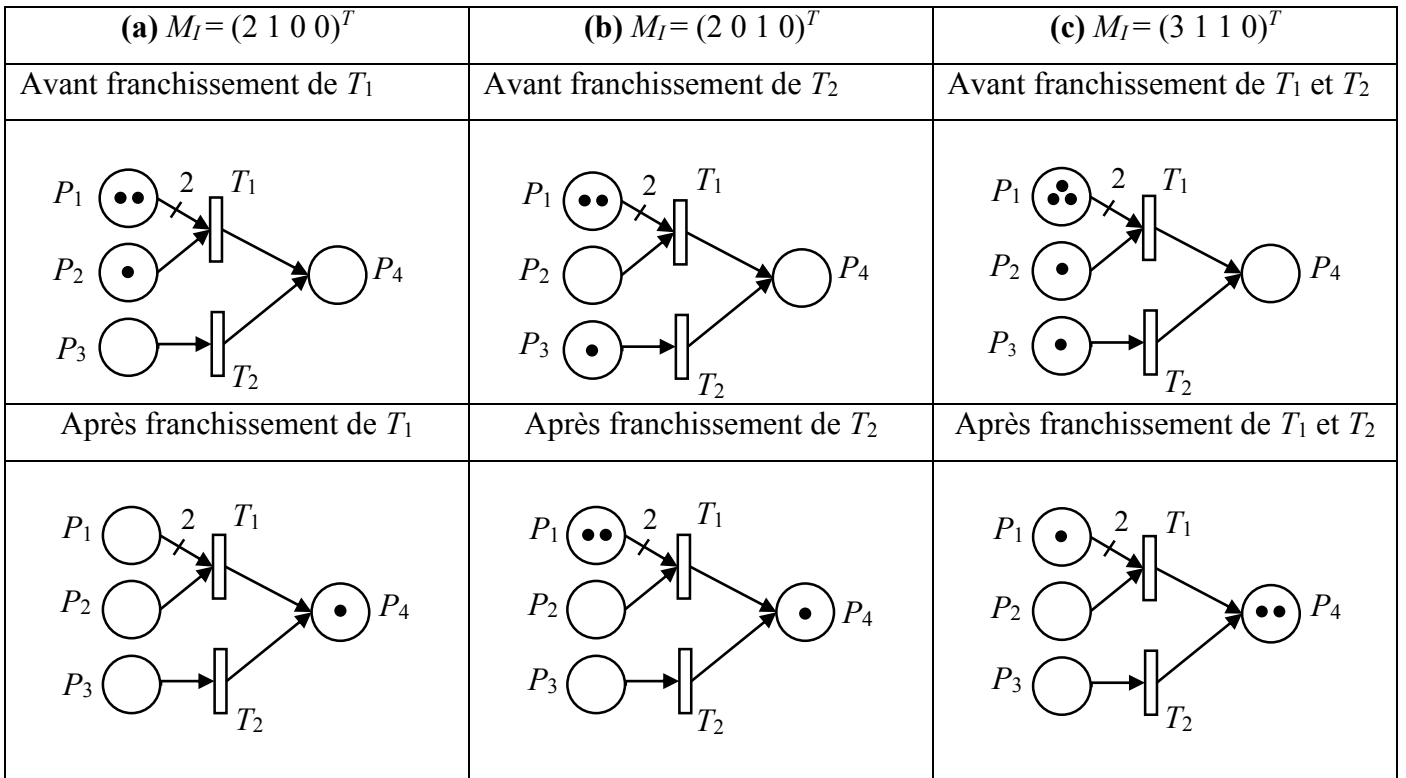


Figure 1.4. Exemple simple d'un RdP

**Exemple 1.3.** L'exemple de la figure 1.4 représente les règles d'évolution pour trois marquages différents (a), (b) et (c). Ce RdP est composé de quatre places  $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$ , deux transitions  $\mathcal{T} = \{T_1, T_2\}$  et des arcs de poids égaux à 1 sauf pour l'arc reliant la place  $P_1$  à la transition  $T_1$  dont le poids vaut 2. La matrice d'incidence de ce réseau, calculée à partir de (1.3) vaut :

$$W = W_{PO} - W_{PR} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} - \begin{pmatrix} 2 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix}$$

La figure 2.1.(a) représente l'état du réseau avant et après franchissement de la transition  $T_1$ . Le marquage initial  $M_I = (2 \ 1 \ 0 \ 0)^T$  (représenté aussi sous la forme  $2P_1, 1P_2$ , une place non citée implique que son marquage est nul) est considéré. Seule la transition  $T_1$  est validée à partir de  $M_I$ , ainsi son degré de validation est :  $n_I(M_I) = \min \left( \left\lfloor \left( \frac{2}{2}, 1 \right) \right\rfloor \right) = 1$ . Le tir de la séquence  $\sigma_U = T_1$ , représentée par le vecteur de franchissement  $X = (1 \ 0)^T$ ,

mène vers un nouveau marquage  $M' = (0 \ 0 \ 0 \ 1)^T$  ce qui est noté  $M_I [T_j > M']$ . Ce dernier est calculé grâce à l'équation (1.7) avec  $M = M_I$  :

$$M' = M + W \cdot X(\sigma_U) = \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -2 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

La figure 2.1.(b) reprend ce RdP et lui associe un nouveau marquage initial  $M_I = (2 \ 0 \ 1 \ 0)^T$ . Seule la transition  $T_2$  est validée à partir de  $M_I$ , ainsi son degré de validation est :  $n_2(M_I) = 1$ . Le tir de cette transition représentée par  $X = (0 \ 1)^T$  mène au marquage suivant :

$$M' = M + W \cdot X(\sigma_U) = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -2 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

De la même façon pour la figure 2.1.(c), on considère le marquage initial  $M_I = (3 \ 1 \ 1 \ 0)^T$ . Les deux transitions  $T_1$  et  $T_2$  sont validées à partir de  $M_I$ , ainsi leurs degrés de validation sont respectivement  $n_1(M_I) = \min\left(\left\lceil \left(\frac{3}{2}, 1\right) \right\rceil\right) = 1$  pour la transition  $T_1$  et  $n_2(M_I) = 1$  pour la transition  $T_2$ . Le tir de la séquence  $\sigma_U = T_1 T_2$ , représentée par le vecteur de franchissement  $X = (1 \ 1)^T$ , mène vers le marquage  $M' = (1 \ 0 \ 0 \ 2)^T$  calculé comme suit :

$$M' = M + W \cdot X(\sigma_U) = \begin{pmatrix} 3 \\ 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -2 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

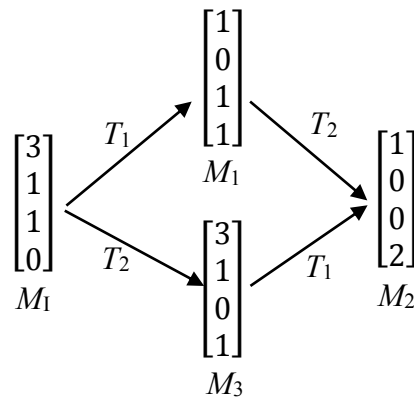


Figure 1.5. Graphe d'atteignabilité du RdP de la figure 1.4.(c)

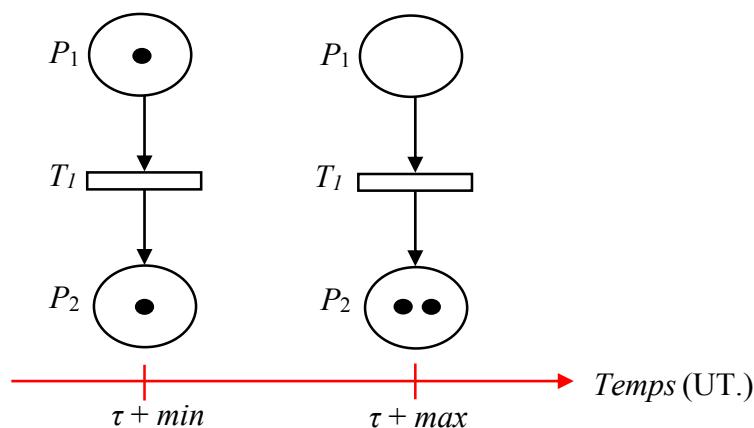


### 1.3. Sémantiques temporelles et contraintes temporelles

Différentes classes de modèles temporisés sont considérées pour décrire le temps nécessaire à une opération. Une sémantique temporelle est nécessaire pour définir les différentes spécifications temporelles associées à chaque modèle. Le temps est introduit dans les RdP pour modéliser l'interaction entre plusieurs activités en tenant compte de leurs instants de démarrage et d'achèvement [Ramchandani 1973 ; Merlin 1975]. Quatre grandes classes de modèles temporisés sont définies en fonction de l'association des temporisations :

- Modèle à places temporisées (le temps est associé aux places RdP *P*-temporisés [Sifakis 1977a ; Sifakis 1977b]) : les jetons générés dans les places de sorties deviennent disponibles pour franchir une transition seulement après l'écoulement d'un délai.
- Modèle à jetons temporisés (le temps est associé aux jetons) : les jetons portent un horodatage indiquant le moment à partir duquel ils sont disponibles pour franchir une transition.
- Modèle à arcs temporisés (le temps est associé aux arcs) [Frutos Escrig et al. 2000] : un délai de déplacement est associé à chaque arc, et les jetons sont disponibles pour franchir une transition seulement quand ils l'atteignent.
- Modèle à transitions temporisées (le temps est associé aux transitions [Ramchandani 1973 ; Merlin 1975]) : le début et la fin d'une activité correspondent respectivement à la validation et au tir d'une transition.

Dans ce travail, on s'intéresse à l'association de la temporisation aux transitions. Seuls ces modèles RdP sont utilisés dans ces travaux de recherche et seront détaillés dans le paragraphe suivant. En général, la prise en compte de l'information temporelle permet de caractériser la validité et la performance d'une activité (comme par exemple, les transferts dans les systèmes de transport automatisés, les chargements de serveurs dans des systèmes de communications, ...). Le franchissement d'une transition appartenant à ces modèles s'effectue après une durée incluse dans un intervalle de temps  $[a : b]$  dont la borne inférieure  $a$  et supérieure  $b$  correspondent respectivement aux délais minimal et maximal de franchissement. Ces délais sont des durées de franchissement, les deux termes seront indifféremment utilisés par la suite. Cet intervalle est défini à partir de l'instant de sensibilisation de la transition. On note que la durée de franchissement d'une transition  $T$  est réinitialisée lorsque celle-ci n'est plus validée. Dans le cas où une transition est validée mais non franchie dans l'intervalle qui lui est associé, elle est désensibilisée, il faut donc attendre qu'elle soit à nouveau sensibilisée pour pouvoir la franchir dans son intervalle de temps.



**Figure 1.6.** Franchissement d'une transition temporisée

**Exemple 1.4.** Dans la figure 1.6 on considère un RdP composé d'une seule transition  $T_1$  et de deux places  $\{P_1, P_2\}$ . Le marquage initial est donné par  $M_I = (1 \ 1)^T$ . On suppose que la transition  $T_1$  est validée à la date  $\tau$ . En associant deux dates  $min$  et  $max$  à la transition,  $T_1$  ne peut pas être tirée avant la date  $(\tau + min)$  et doit être tirée au plus tard à la date  $(\tau + max)$  (sauf si le tir d'une autre transition avec laquelle  $T_1$  était en conflit rend  $T_1$  non validée avant cette date). On associe à  $T_1$  une contrainte temporelle définie par l'intervalle  $[min : max]$ .

#### 1.4. Réseaux de Petri temporisés et temporels

La prise en compte du facteur temps est essentielle pour améliorer le diagnostic des SED. L'information temporelle permet de discerner un comportement fautif d'un comportement nominal d'un système. Plusieurs extensions ont été élaborées pour modéliser, analyser ou encore diagnostiquer les SED temporisés. Dans ce qui suit, nous nous intéresserons aux RdP dont la temporisation est associée aux transitions. Nous présenterons deux extensions de ces réseaux : les RdP  $T$ -temporels (Time PNs) et les RdP  $T$ -temporisés (Timed PNs).

Comme nous l'avons expliqué dans la section précédente, les réseaux de Petri présentent un modèle formel pour décrire et étudier le traitement de l'information dans des systèmes caractérisés comme étant concurrents, asynchrones, parallèles, déterministes ou stochastiques. Depuis sa création, le formalisme a été étendu par des praticiens et des théoriciens pour traiter des systèmes complexes avec des aspects temporels attachés à de nombreux champs d'application. Ce travail de recherche est basé sur les RdP  $T$ -temporels [Merlin 1975]. Il s'agit d'un RdP où chaque transition est caractérisée par un intervalle de temps. Une transition sensibilisée ne

peut être franchie que dans cet intervalle de temps. Cet intervalle est constitué d'une durée minimale et une autre maximale. Formellement, un RdP  $T$ -temporel est défini comme suit :

**Définition 1.4.** Un RdP  $T$ -temporel,  $G_{TL}$  est un doublet défini par  $G_{TL} = \langle G, I \rangle$  où :

- $G = \langle \mathcal{P}, \mathcal{T}, W_{PR}, W_{PO} \rangle$  est un RdP
- $I : \mathcal{T} \rightarrow \mathbb{R}^+ \times (\mathbb{R}^+ \cup \{\infty\})$  est la fonction qui associe à chaque transition  $T$  un intervalle de franchissement ( $\mathbb{R}^+$  est l'ensemble des nombres réels positifs)

Un RdP  $T$ -temporel marqué est noté par  $\langle G_{TL}, M_I \rangle$  avec  $M_I$  le marquage initial du réseau.

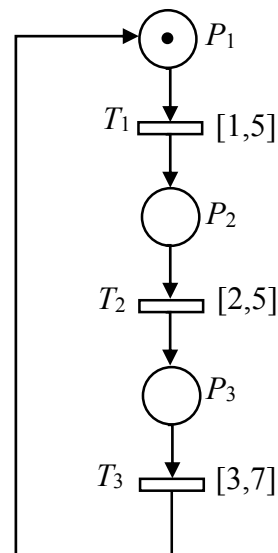
En ce qui concerne les RdP  $T$ -temporisés [Ramchandani 1973], une durée de sensibilisation est associée à chaque transition (les RdP  $T$ -temporisés peuvent être considérés comme un cas particulier des RdP  $T$ -temporels). Formellement, un RdP  $T$ -temporisé est défini comme suit :

**Définition 1.5.** Un RdP  $T$ -temporisé,  $G_{TS}$  est un doublet défini par  $G_{TS} = \langle G, D \rangle$  où :

- $G = \langle \mathcal{P}, \mathcal{T}, W_{PR}, W_{PO} \rangle$  est un RdP
- $D : \mathcal{T} \rightarrow \mathbb{R}^+$  est la fonction durée qui associe à chaque transition  $T$  un nombre réel positif décrivant la durée de sensibilisation de  $T$  avant son franchissement

Un RdP  $T$ -temporisé marqué est noté par  $\langle G_{TS}, M_I \rangle$  avec  $M_I$  le marquage initial du réseau.

La sémantique est que les transitions doivent rester validées au moins la durée  $d$  associée à ces transitions. Pendant cette durée  $d$  la transition n'est pas franchie, son franchissement n'est assuré qu'après l'écoulement de cette durée.



**Figure 1.7.** Exemple d'un RdP  $T$ -temporel

**Exemple 1.5.** On considère l'exemple de la figure 1.7 d'un RdP  $T$ -temporel composé de trois transitions  $\{T_1, T_2, T_3\}$  et de trois places  $\{P_1, P_2, P_3\}$ . Le marquage initial est donné par  $M_I = (1 \ 0 \ 0)^T$ . Un intervalle de temps est associé à chaque transition. La transition  $T_3$  par exemple ne peut être franchie avant l'écoulement de trois unités de temps depuis sa sensibilisation. Par contre cette dernière devrait être franchie avant sept unités de temps si elle reste toujours sensibilisée. La contrainte temporelle suivante  $3 \leq d_3 \leq 7$  est obtenue pour la transition  $T_3$ , où  $d_3$  représente le temps nécessaire pour franchir cette transition.

### 1.5. Réseaux de Petri stochastiques

Les Réseaux de Petri Stochastiques (RdPS) représentent une extension des RdP  $T$ -temporisés prenant en compte à la fois le temps et les phénomènes stochastiques [Molloy 1982 ; Valette 1992]. Les durées de franchissement des transitions sont représentées par des variables aléatoires (VA) définies par une fonction de densité de probabilité et permettent de définir les dates d'occurrence des événements. L'évaluation des performances ramène l'étude à deux extensions particulières désignées par modèles RdPS markoviens et modèles RdPS non markoviens [Bobbio et al. 1998 ; Trivedi et al. 1995 ; Haddad et al. 2006 ; Horváth et al. 2012 ; German 2000, Distefano et al. 2017]. Trois types de lois de probabilité particulières permettant de se rapprocher des distributions habituelles des processus stochastiques sont considérées dans notre travail [Lefebvre 2017a] :

- loi uniforme sur support fini
- loi triangulaire symétrique
- loi exponentielle tronquée

Grâce aux RdPS, il est possible d'évaluer les performances des systèmes fondés sur des modèles probabilistes. Le fonctionnement d'un RdPS doit obéir, en plus à des règles de validation et de franchissement définies précédemment, à un ensemble de politiques. Ces politiques servent à gérer la sélection de la prochaine transition à franchir, pour un marquage donné, lorsque plusieurs transitions sont validées à partir de ce même marquage. Les RdPS sont caractérisés par leurs politiques de choix, de service et de mémoire [Marsan 1988 ; Bause et Kritzinger 2002 ; Diaz 2013] définies comme suit :

## Les politiques des RdPS : Politique de (choix / mémoire / service)

### Politique de choix :

En cas de conflit, on doit spécifier la manière dont est choisie la prochaine transition à franchir. Deux types de politiques de choix sont considérées : la première dite par pré-sélection (preselection policy), pour laquelle la prochaine transition à franchir au sein d'un conflit est déterminée par un calcul de probabilité selon une distribution uniforme couplée à des poids préalablement attribués aux transitions. La seconde politique est appelée par compétition (race policy), dans ce cas, pour un ensemble de transitions en conflit à partir d'un marquage donné, un tirage aléatoire des durées est effectué et par la suite la transition correspondant à la durée la plus petite sera franchie.

### Politique de mémoire :

Face aux transitions sensibilisées mais non franchies, il faut déterminer si leurs durées de tir restantes doivent être mémorisées ou pas. Dans le cas où ces durées sont réinitialisées après chaque tir de transition, la politique est appelée politique de réinitialisation (resampling memory). On parle alors d'une absence de mémoire. Un autre choix consiste à prendre en compte uniquement les durées relatives à la dernière sensibilisation du système (enabling memory). Une remise à zéro de la mémoire est effectuée lors de la désensibilisation de la transition. Il est aussi possible de considérer les délais cumulés de toutes les sensibilisations de la transition depuis son dernier tir (age memory).

### Politique de service :

On distingue deux politiques de service : finie et infinie. Dans le cas de la politique serveur fini, un nombre fixe de jetons peut être tiré simultanément. On parle alors d'un serveur à degré de réentrance borné (multiple-server policy). Lorsque ce nombre est égal à 1, on parle de serveur non-réentrant (single-server policy), contrairement au cas serveur infini (ou réentrant) où le nombre de franchissements est illimité et dépend uniquement du degré de validation des transitions.

### 1.5.1. Modèles RdPS markoviens

On s'intéresse dans cette partie à la description et à l'exploitation d'une sous-classe de processus stochastiques à savoir les processus markoviens, aussi appelés processus sans mémoire, c'est-à-dire que l'état futur dépend uniquement de l'état présent, et non du passé. Toutes les transitions d'un RdPS markovien sont temporisées avec un temps aléatoire distribué selon une loi exponentielle. La loi exponentielle modélise la durée de vie d'un phénomène sans mémoire (ou sans vieillissement, ou sans usure). La probabilité  $P$  que le phénomène dure au moins  $s + t$  sachant qu'il a déjà duré  $t$  est identique à la probabilité de durer  $s$  à partir de son initialisation. L'absence de mémoire représente une propriété importante de la distribution exponentielle (figure 1.8) que l'on peut traduire mathématiquement par l'équation :

$$\forall s, t \geq 0 \quad P(t' > s + t | t' > t) = P(t' > s) \quad (1.8)$$

où  $t'$  modélise la durée de vie d'un système et  $P(t' > s + t | t' > t)$  est la probabilité conditionnelle de  $(t' > s + t)$  sachant que  $(t' > t)$ .

La distribution d'une variable aléatoire exponentielle de paramètre  $\lambda > 0$  est donnée par :

$$f(t) = \begin{cases} \lambda e^{-\lambda t} & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases} \quad (1.9)$$

$f(t)$  est la fonction de densité de probabilité caractérisant la durée de franchissement des transitions, ainsi sa fonction de répartition  $F(t)$  est calculée comme suit :

$$F(t) = \int_{-\infty}^t f(x) dx = \int_0^t \lambda e^{-\lambda x} dx = 1 - e^{-\lambda t} \quad (1.10)$$

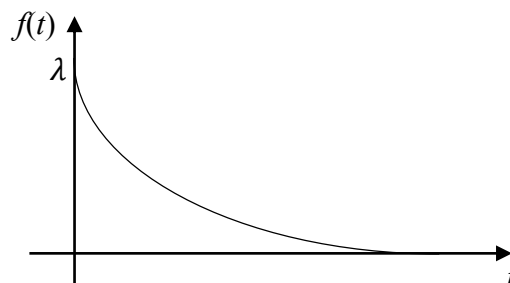


Figure 1.8. Distribution exponentielle des durées de tir de transition

Un RdPS markovien est caractérisé par une politique de choix par compétition, une absence de mémoire et une politique de service infini. Pour plus de détails sur la loi exponentielle ainsi que ses propriétés, voir [Ross 2014].

Formellement, un RdPS est défini comme suit :

**Définition 1.6.** Un RdP Stochastique temporisé  $G_S$  est défini par les politiques de choix, mémoire et service, il est donné par  $G_S = \langle G, \mu \rangle$  où :

- $G = \langle \mathcal{P}, \mathcal{T}, W_{PR}, W_{PO} \rangle$  est un RdP
- $\mu = (\mu_j) \in (\mathbb{R}_+^*)^q$  représente le vecteur des taux de franchissement des transitions ( $q$  est le nombre de transitions)

Un RdPS temporisé marqué sera noté  $\langle G_S, M_I \rangle$  avec  $M_I$  le marquage initial.

### 1.5.2. Modèles RdPS non markoviens

Dans ce travail de thèse, on s'intéresse aussi aux modèles RdPS non markoviens. Les RdPS non markoviens permettent d'imiter le comportement d'un système en traitant des modèles généralement plus proches de la réalité. Contrairement aux modèles markoviens, une prise en compte de la mémoire peut être considérée. Pour un RdPS non markovien toutes les transitions sont temporisées avec une durée aléatoire distribuée selon une loi quelconque. Dans la suite, toute loi à support borné et dont la densité de probabilité est connue peut être utilisée. En particulier, les lois uniformes bornées, triangulaires symétriques et exponentielles tronquées sont considérées. La définition analytique de leur densité de probabilité est supposée être connue.

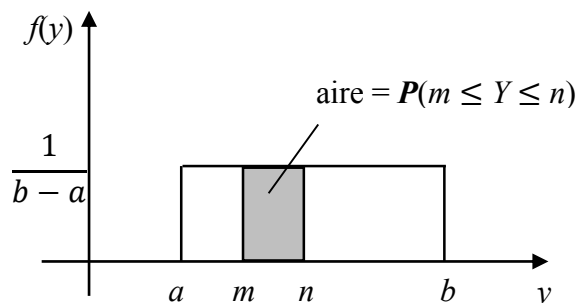
Soit une variable aléatoire  $Y$  susceptible de prendre toutes les valeurs d'un intervalle fini  $[a, b]$ , sans privilégier aucune région de  $[a, b]$  (on parle d'événements équiprobables). La probabilité que  $Y$  prenne une valeur appartenant à l'intervalle  $[m, n] \subset [a, b]$  est proportionnelle à la longueur de  $[m, n]$ , d'où :

$$P(m \leq Y \leq n) = \frac{m-n}{b-a}, \quad \text{où } P(m \leq Y \leq n) = \int_m^n f(y)dy \quad (1.11)$$

$$\text{où } f(y) = \begin{cases} \frac{1}{b-a} & \text{si } a \leq y \leq b \\ 0 & \text{sinon} \end{cases} \quad (1.12)$$



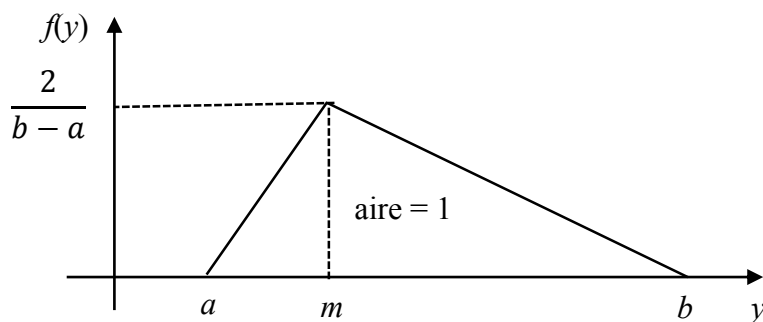
La fonction  $f(y)$  représente la densité de probabilité qui définit le comportement aléatoire (stochastique) de la variable aléatoire  $Y$  et permet ainsi de caractériser sa loi de probabilité (distribution). Dans ce travail, cette variable aléatoire représente les durées de franchissement des transitions d'une loi uniforme bornée (figure 1.9).



**Figure 1.9.** Distribution uniforme bornée des durées de tir de transition

Une loi triangulaire (figure 1.10) est définie de la même manière avec l'équation suivante :

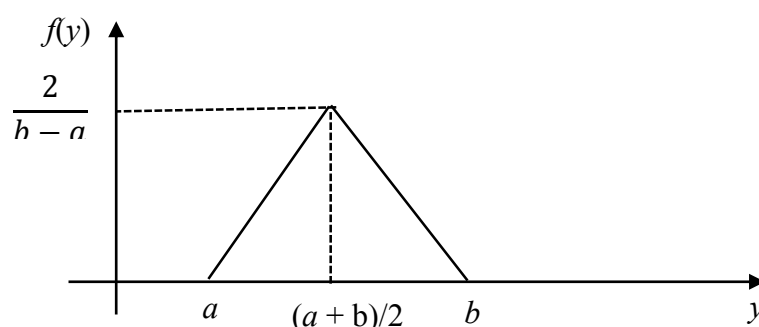
$$f(y) = \begin{cases} \frac{2(y-a)}{(m-a)(b-a)} & \text{si } a \leq y \leq m \\ \frac{2(b-y)}{(b-m)(b-a)} & \text{si } m \leq y \leq b \\ 0 & \text{sinon} \end{cases} \quad (1.13)$$



**Figure 1.10.** Distribution triangulaire des durées de tir de transition

Dans ce travail, on s'intéresse au cas particulier d'une loi triangulaire pour sa facilité d'implémentation. Dans le cas  $m = (a + b)/2$ , on obtient la figure 1.11 et la distribution est appelée distribution triangulaire symétrique. L'équation (1.13) devient :

$$f(y) = \begin{cases} \frac{4}{(b-a)^2 \cdot (y-a)} & \text{si } a \leq y \leq (a+b)/2 \\ \frac{4}{(b-a)^2 \cdot (b-y)} & \text{si } (a+b)/2 \leq y \leq b \\ 0 & \text{sinon} \end{cases} \quad (1.14)$$



**Figure 1.11.** Distribution triangulaire symétrique des durées de tir de transition

D'une part, la motivation pour considérer des densités de probabilité uniformes repose sur le fait que des processus aléatoires uniformes (positifs) ainsi que les processus déterministes peuvent être obtenus sous la forme de deux cas limites de (1.12) :  $a = 0$  et  $b = \infty$  conduit à un processus uniforme aléatoire positif et  $a = b = d$  conduit à un processus déterministe de durée constante  $d$ . D'autre part, la motivation pour considérer des densités de probabilité triangulaires symétriques (1.14) est de décrire la dispersion de la durée  $d$  autour d'une valeur moyenne. Enfin, les densités de probabilité exponentielles sont utilisées avec les modèles markoviens pour modéliser les phénomènes de vieillissement et d'échec. Les densités de probabilité exponentielles tronquées sont des approximations des densités de probabilité exponentielles dont le comportement limite pour  $b = \infty$  est équivalent aux densités de probabilité exponentielles habituelles.

## 1.6. Business Process Modeling

Le "Business Process Modeling" (BPM), traduit littéralement en français par la modélisation de processus métier, est défini par [Van der Aalst et al. 2003] comme l'utilisation de méthodes, techniques et systèmes logiciels pour concevoir, exécuter, contrôler et analyser des processus opérationnels faisant intervenir des hommes, des applications, des documents et d'autres sources d'information. Le but est, en effet, de favoriser l'efficacité opérationnelle tout en abordant la question de l'amélioration continue des processus priorisant le point de vue « métier » au point de vue « technique ». Cela rend donc les processus plus efficaces et capables de s'adapter à d'éventuels changements dans l'entreprise.

Le BPM est un mécanisme de modélisation d'un ensemble d'activités au sein d'une organisation. Il se compose d'un diagramme appelé le diagramme de processus métiers (Business Process Diagram (BPD)). Ce dernier a été conçu pour être facile à utiliser et à comprendre, mais offre également la possibilité de modéliser des processus métiers complexes. Les tâches sont ordonnées et des dépendances temporelles peuvent exister entre elles. La durée autorisée de chaque tâche est comprise entre un minimum et un maximum. Ces tâches sont représentées graphiquement par des rectangles et les contraintes inter-tâches sont représentées entre deux tâches par une ligne en pointillés. Les séquences d'activités sont connectées aux deux connecteurs ET et OU exclusif (AND / XOR), qui représentent les conditions de routage entre les flux. Les connecteurs AND et XOR sont respectivement identifiés par les symboles + et  $\times$ . Chaque connecteur peut fusionner des flux d'entrée et diviser des flux de sortie. Les états initiaux et terminaux du modèle sont représentés par des cercles. Les activités d'un processus métier sont définies grâce aux portes logiques. De ce fait, l'utilisation de connecteurs permet de modéliser des flux de tâches complexes. Ces flux combinent des exécutions parallèles et séquentielles ainsi que des divisions et connexions (splits and joins).

**Exemple 1.6.** L'exemple de la figure 1.12, représente un BPM avec huit tâches structurées par un XOR connecteur de division X1, un AND connecteur de division A1, un AND connecteur de connexion A2 et un XOR connecteur de connexion X2. La durée autorisée de chaque tâche est comprise entre un minimum et un maximum. En effet, un intervalle de temps est associé à chaque tâche. Des contraintes peuvent également exister entre une paire de tâches représentant une contrainte temporelle inter-tâche. Par exemple, dans l'exemple BPM de la figure 1.12, la tâche TA1 a une durée comprise entre 2 et 4 UT. La ligne en pointillés TAI1 limite la durée entre le début de TA1 et celui de TA2 à 8 UT.

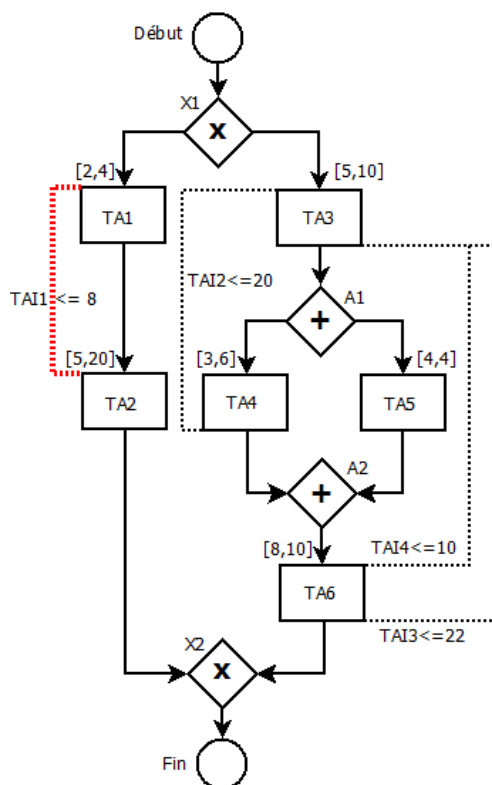


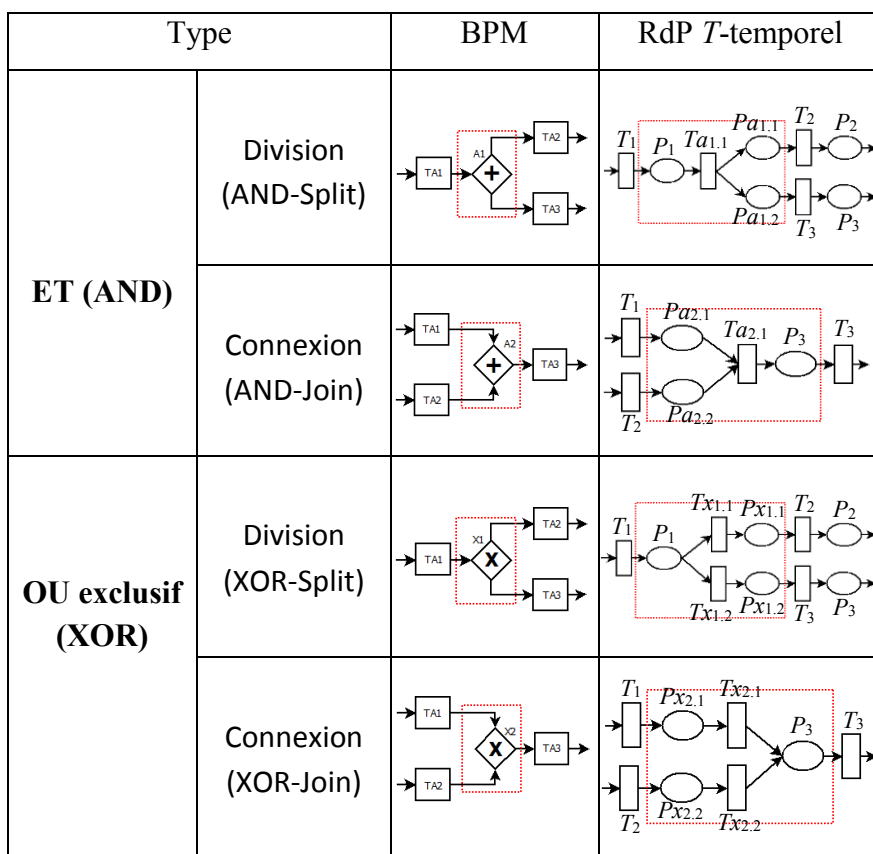
Figure 1.12. Exemple d'un BPM

## 1.7. Transformation des BPM en RdP équivalent

### 1.7.1. Les contraintes structurelles

Beaucoup de similitudes existent entre les deux modélisations, par RdP et par BPM, en termes de notations et de représentations ; mais le BPM fournit un certain nombre d'avantages concernant les spécifications temporelles complexes. Dans cette section nous proposons une contribution pour transformer un modèle BPM incluant des contraintes temporelles en un modèle RdP  $T$ -temporel.

Le comportement d'un processus est interprété en termes de contraintes structurelles sur l'activité. La présence des portes logiques permet de contrôler ce comportement. La transformation des contraintes structurelles du BPM vers le formalisme RdP  $T$ -temporel 1-borné (c'est-à-dire que toutes les places du réseau sont 1-bornées) peut être assurée de manière systématique conformément aux règles de transformation détaillées dans la figure 1.13.



**Figure 1.13.** La transformation des contraintes structurelles

Chaque tâche du modèle de processus métier est représentée dans le RdP  $T$ -temporel par la succession d'une transition  $T$  et de son postset  $T^\circ$  (c'est-à-dire l'ensemble des places de sortie de  $T$ ) :

- Le connecteur AND-Split A1 dans le BPM est modélisé par une structure de parallélisme  $Ta_{1,1}$ , de telle sorte que TA2 et TA3 s'exécutent en parallèle.
- Le connecteur AND-Join A2 est transformé en considérant la réalisation des deux tâches TA1 et TA2 en parallèle afin que TA3 puisse être exécutée. Avec le formalisme RdP  $T$ -temporel, la transition  $Ta_{2,1}$  n'est activée que lorsque TA1 et TA2 sont exécutées. Ensuite, le tir de  $Ta_{2,1}$  active  $T_3$ .
- Le connecteur XOR-Split X1 divise le flux entrant en deux flux sortants selon un choix effectué dans  $P_1$ . Ainsi, le jeton à la place  $P_1$  permettra l'exécution que d'une seule transition entre les deux transitions en concurrence.

- Le connecteur XOR-Join X2 est modélisé par les transitions  $T_{x2.1}$  et  $T_{x2.2}$ .

### 1.7.2. Les contraintes temporelles

Des contraintes temporelles sont appliquées afin de limiter la durée de chaque activité dans un intervalle de temps autorisé. Quatre types de contraintes inter-tâches sont définis. Par exemple, la figure 1.12 représente un BPM avec 4 contraintes inter-tâches modélisées avec des lignes en pointillés reliant chaque paire de tâches. La première contrainte inter-tâche TAI1 relie les limites supérieures de TA1 et TA2, ce qui signifie que la contrainte est appliquée à leur date de début (TA1-Début / TA2-Début). La deuxième contrainte TAI2 relie la date de début de TA3 à la date de fin de TA4 (TA3-Début / TA4-Fin). TAI3 est appliquée à la date de fin de TA3 et de TA6 (TA3-Fin / TA6-Fin). TAI4 contraint la date de fin de TA3 et la date de début de TA6 (TA3-Fin / TA6-Début).

Un ensemble de règles de transformation est assurée de manière systématique pour convertir les contraintes inter-tâches du modèle BPM en modèle RdP. Cet ensemble de règles est illustré dans la figure 1.14. La contrainte inter-tâche Début / Fin liant TA1 et TA2 exige que la tâche TA2 doit se terminer au plus tard 30 UT à partir du début de la tâche TA1. Cette contrainte est modélisée dans le modèle RdP grâce aux trois transitions  $T_{c1}$ ,  $T_{c2}$  et  $T_{glob}$ . En plus de la contrainte inter-tâche Début / Fin liant TA1 et TA2, on a une succession de contraintes temporelles sur les différentes tâches séquentielles du chemin liant TA1 et TA2.

On note que les transitions immédiates  $T_{c1}$  et  $T_{c2}$  sont ajoutées pour respecter la structure habituelle d'un modèle RdP. La transition  $T_{c2}$  ne sera pas validée avant l'expiration du délai requis pour le franchissement de la transition  $T_{glob}$ . Pour cette raison, l'intervalle de temps TAI = [0 : 30] est associé à  $T_{glob}$ .

En ce qui concerne la contrainte inter-tâche Début / Début, le modèle RdP correspondant est représenté avec une structure de parallélisme à partir de la transition  $T_{c1}$  (une transition immédiate utilisée pour structurer le modèle sous contrainte). Ensuite, la tâche TA1 doit être accomplie et le délai doit s'écouler afin que la transition  $T_2$  soit validée. De la même manière, les deux autres contraintes inter-tâches Fin / Fin et Fin / Début sont traitées.

Finalement, les diverses contraintes temporelles inter-tâches peuvent être transformées depuis le modèle BPM dans un modèle RdP équivalent afin de modéliser un processus de flux de travail complet.

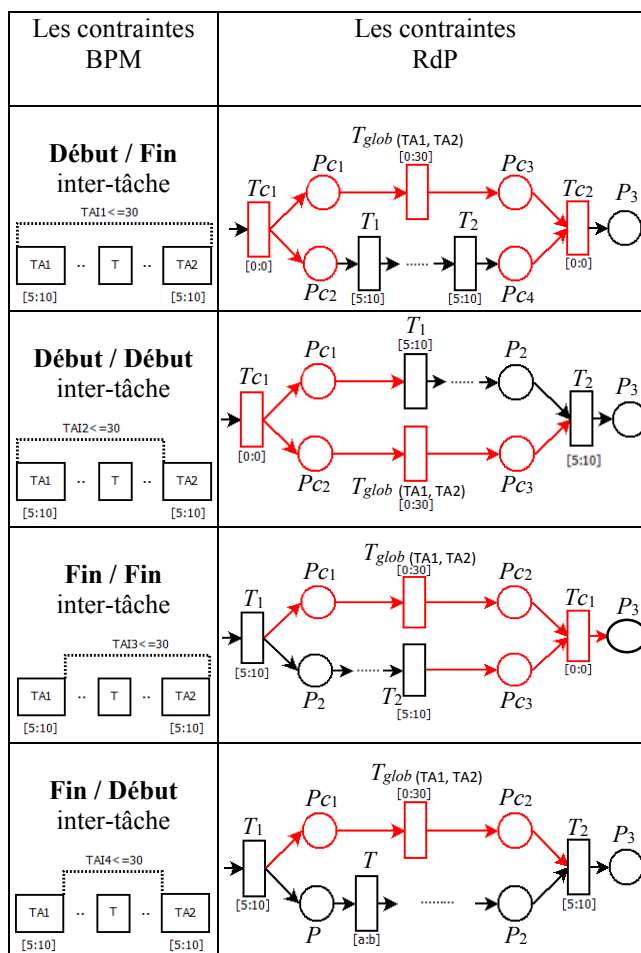


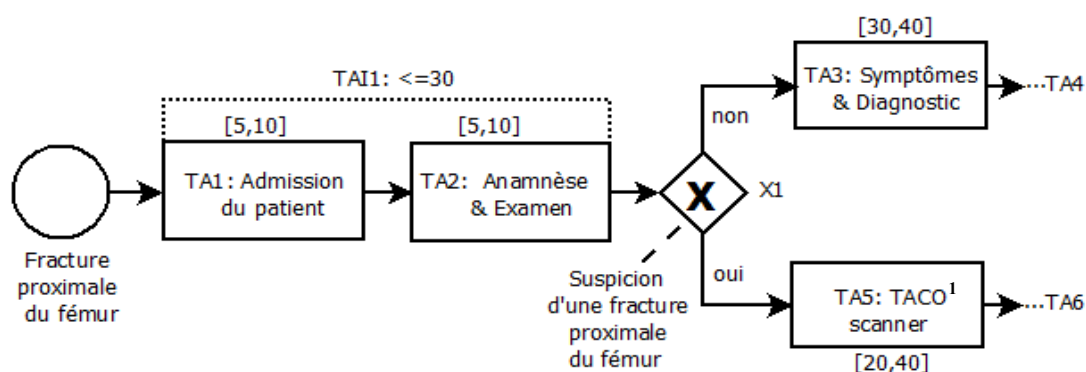
Figure 1.14. La transformation des contraintes temporelles

### 1.8. Exemple

Une étude de cas modélisée par un BPM est présentée dans cette section. Nous illustrons les transformations des spécifications temporelles et structurales du BPM en RdP. La figure 1.15 (qui est divisée en 3 parties : figure 1.15 a), figure 1.15 b) et la figure 1.15 c)) représente le parcours simplifié d'un patient au service des urgences d'un hôpital [Kumar et Barton 2016]. Le processus modélisé dans la figure 1.15 consiste en 14 tâches coordonnées par des flux séquentiels avec des structures de parallélisme et de choix.

Cet exemple est divisé en 3 parties dans lesquelles chaque contrainte inter-tâche est décrite de manière contextuelle. Ensuite, la transformation de ces parties en RdP est représentée dans la figure 1.16.

Une fois admis au service (TA1), le patient subit une anamnèse (entretien qui relate les antécédents médicaux) et un examen (TA2). Une contrainte inter-tâche Début / Fin existe entre TA1 et TA2, comme indiqué sur la figure 1.15 a). Outre la durée d'exécution des deux tâches TA1: [5:10] et TA2: [5:10], la contrainte inter-tâche TAI1 prend en compte un éventuel temps d'attente supplémentaire. En effet, TAI1 prend en compte le temps d'attente possible après l'admission du patient et avant le début de son examen. Selon le résultat de l'examen, le patient doit passer un scanner (TA5). Sinon, il est diagnostiqué et ensuite préparé pour une thérapie (TA3).



**Figure 1.15 a).** La contrainte inter-tâche Début / Fin dans le modèle BPM

Si le patient est préparé pour une thérapie, le diagnostic est suivi par la thérapie personnalisée A (TA4). Alternativement, en fonction des résultats de son diagnostic d'imagerie (TA6), il est soit traité avec une thérapie B (TA7), soit par une intervention chirurgicale (TA11) qui doit être programmée (TA8). On note que le temps entre la chirurgie et sa planification doit être inférieur à 140 UT (représenté dans la figure 1.15 b) sous la contrainte inter-tâche TAI2 de type Début / Début). Cela signifie que la durée maximale autorisée pour réaliser la planification est de 20 UT. La planification chirurgicale (TA9) et l'administration du médicament contre la douleur (TA10), sont deux tâches parallèles préalables qui suivent (TA8) et qui doivent être effectuées avec une durée maximale de 40 UT et 80 UT respectivement.

<sup>1</sup> Tomographie Axiale Calculée par Ordinateur (TACO)



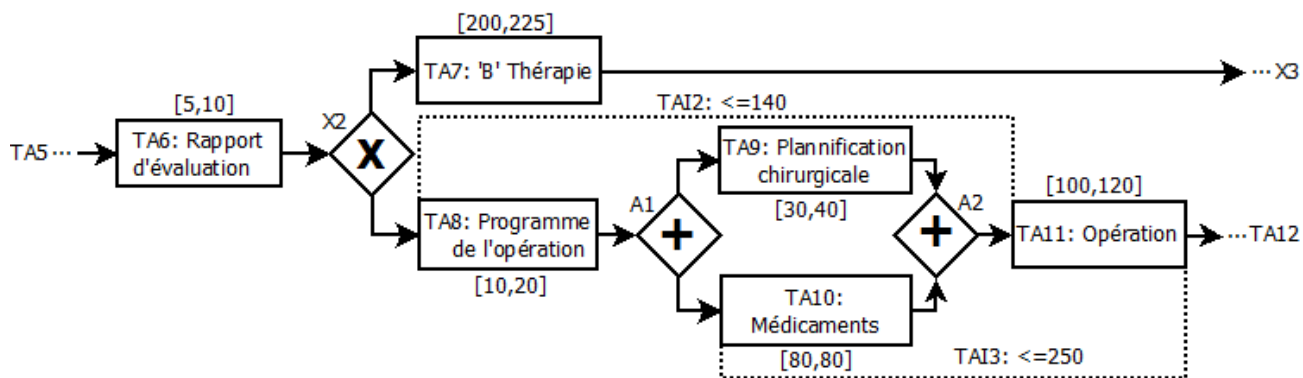


Figure 1.15 b). La contrainte inter-tâche Début / Début dans le modèle BPM

Finalement, un connecteur XOR-Join (X4 sur la figure 1.15 c)) assure l'exécution d'une seule séquence de flux, soit celle contenant le traitement thérapeutique soit celle nécessitant une intervention chirurgicale. Cela mène ensuite aux deux tâches finales de ce processus clinique, à savoir la documentation du cas (TA13) et la libération du patient (TA14).

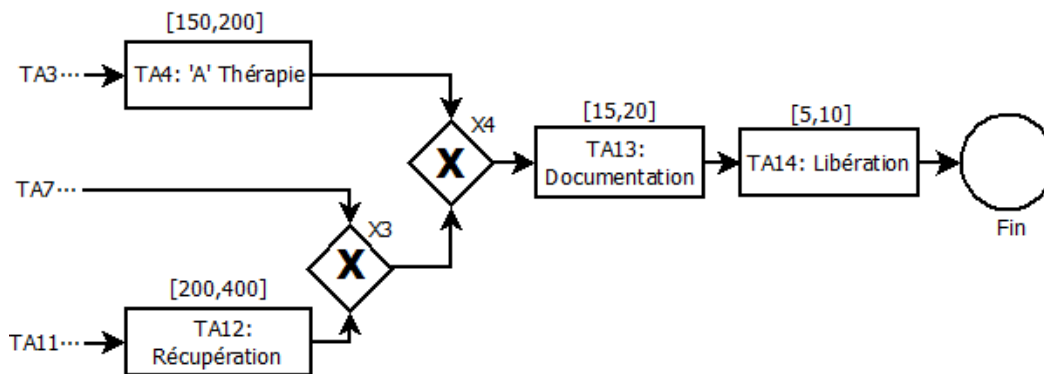


Figure 1.15 c). La dernière partie BPM de ce processus clinique

La transformation en RdP de ce processus clinique est obtenue en appliquant les règles de transformation présentées dans la section 1.7. La figure 1.16 a) montre la transformation de la première partie du processus clinique. La contrainte inter-tâche Début / Fin TAI1 entre l'admission du patient (TA1) et son examen (TA2)

est représentée par une transition qui s'active après le début du processus. Des places supplémentaires  $P_{c_i}$ ,  $i = 1, \dots, 4$  sont employées pour cette raison.

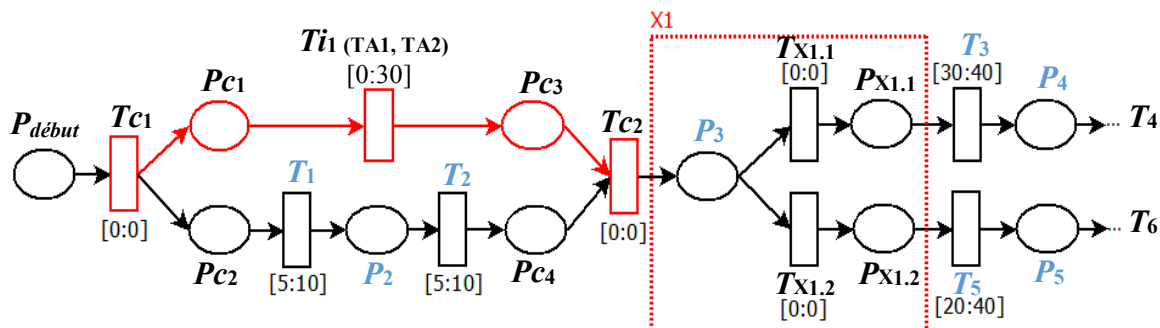


Figure 1.16 a). La contrainte inter-tâche Début / Fin dans le modèle RdP

La transformation du BPM en RdP se poursuit de la même manière, en respectant le type et l'emplacement de chaque connecteur. Par exemple, sur la figure 1.16 b) le connecteur XOR-Split X2 n'autorise qu'une seule séquence de flux basée sur les résultats du diagnostic du patient.

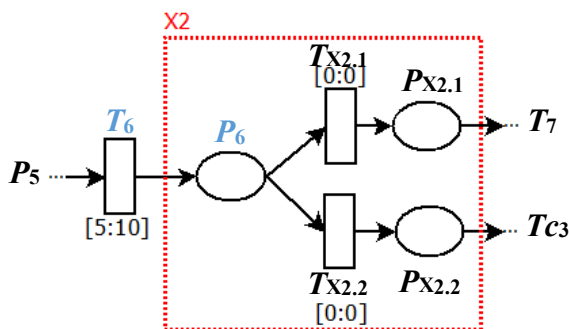


Figure 1.16 b). Le XOR-Split connecteur dans le modèle RdP

La figure 1.16 c) représente deux contraintes critiques. La première est une contrainte inter-tâche TAI2 de type Début / Début, qui contraint la chirurgie (TA11) et sa planification (TA8) avec un temps d'attente qui ne doit pas dépasser 140 UT. La deuxième est une contrainte Début / Fin ((TAI3) ≤ 250 UT), entre la chirurgie et la prise du médicament antidouleur (TA10). On note que, les deux contraintes inter-tâches TAI2 et TAI3 doivent être respectées car le non-respect de l'une influence l'autre.

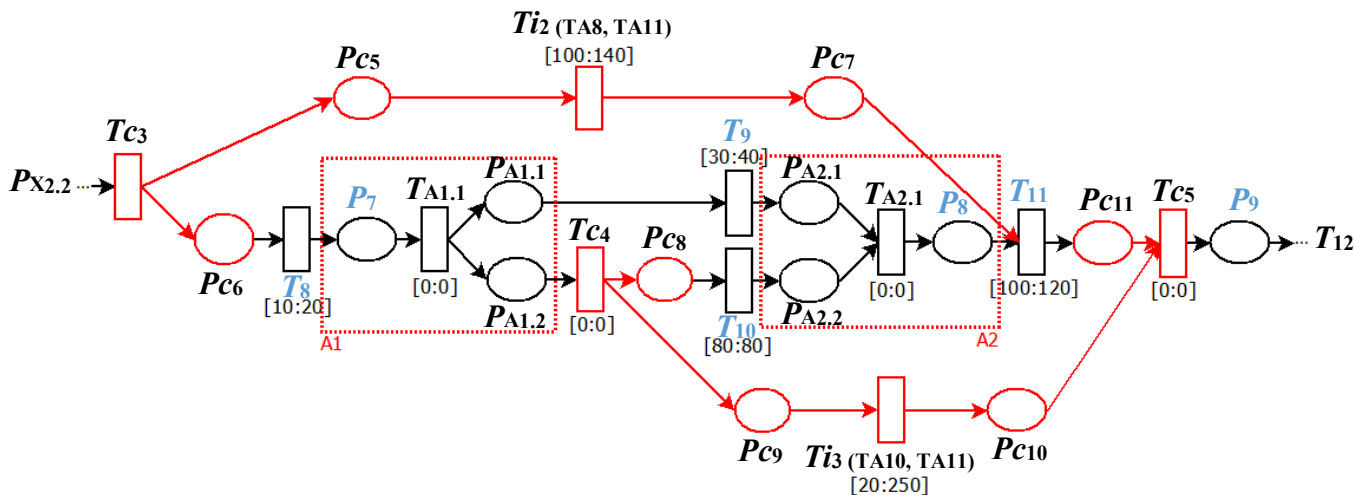


Figure 1.16 c). La contrainte inter-tâche Début / Début dans le modèle RdP

Enfin, pour atteindre la fin du processus (c'est-à-dire la libération du patient (TA14)), deux connecteurs de type XOR-Join sont employés sur la figure 1.16 d), ainsi la place  $P_{end}$  correspond à la fin du processus.

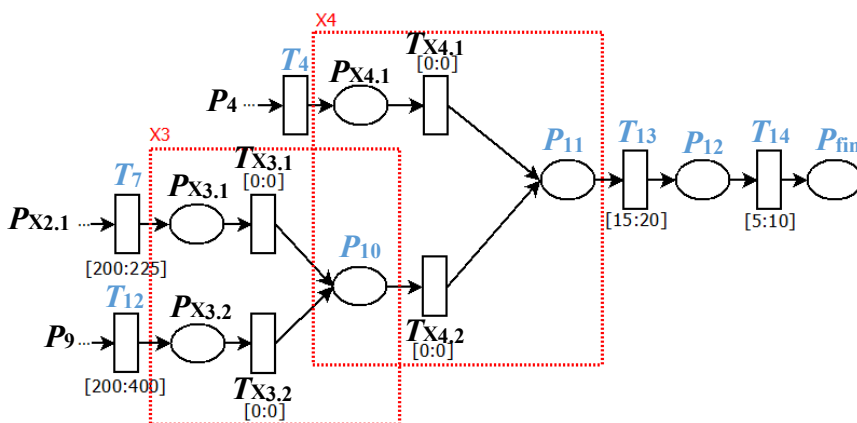


Figure 1.16 d). La dernière partie RdP de ce processus clinique

### 1.9. Conclusion

Dans ce chapitre, nous avons introduit le formalisme des automates à états finis ainsi que le formalisme des RdP. Ces deux formalismes sont les plus utilisés dans la modélisation des SED. Nous avons par la suite exposé les extensions de RdP correspondant aux RdP  $T$ -temporels, RdP  $T$ -temporisés et RdPS en s'appuyant sur

---

l'importance de la notion du temps dans notre travail. Nous nous sommes intéressés à deux types de processus stochastiques markoviens et non markoviens. Un intérêt particulier est porté sur les processus non markoviens dont les temporisations sont distribuées selon des lois de probabilité autres que la loi exponentielle habituellement utilisée.

Nous avons ensuite introduit de nouveaux types de contraintes temporelles dans les RdP  $T$ -temporels. Dans ce but, nous avons fait appel aux BPM comme outil de modélisation qui traite des systèmes d'information complexes. Contrairement à d'autres modélisations qui ne considèrent que des spécifications temporelles simples (c'est-à-dire que chaque contrainte est généralement attachée à une tâche unique), cette limitation a été levée grâce aux BPM. Une approche systématique pour ajouter des contraintes temporelles complexes aux modèles des RdP  $T$ -temporels a été élaborée. En particulier, les contraintes inter-tâches ont été prises en compte. Elles sont tout d'abord exprimées avec les modèles BPM, puis des règles de transformation (BPM / RdP  $T$ -temporel) sont appliquées pour inclure ces contraintes dans les modèles RdP  $T$ -temporels. Finalement, un exemple d'étude de cas a été élaboré afin de mieux illustrer cette transformation BPM/ RdP  $T$ -temporel.



# Chapitre 2. Diagnostic par calcul des probabilités des trajectoires compatibles

2.2. Modélisation des défauts .....	41
2.2.1. Réseaux de Petri partiellement mesurés (POP <sub>N</sub> ) et (POTP <sub>N</sub> ) .....	42
2.2.2. Défauts structurels et temporels .....	44
A. Défauts structurels.....	44
B. Défauts temporels .....	46
2.3. Etat de l'art .....	48
2.3.1. Diagnostic des RdP non temporisés .....	48
A. Méthodes structurelles.....	49
B. Méthodes basées sur la dynamique du RdP.....	50
2.3.2. Diagnostic des RdP temporels .....	52
2.4. Calcul des trajectoires compatibles .....	55
2.4.1. Trajectoires compatibles non temporisées .....	55
A. Approche par LMI .....	57
B. Approche incrémentale .....	59
2.4.2. Trajectoires compatibles partiellement temporisées.....	64
2.5. Probabilités des trajectoires .....	66
2.6. Probabilités des défauts .....	77
2.6.1. Probabilités des défauts structurels .....	78
2.6.2. Probabilités des défauts temporels.....	79
2.7. Cas d'étude .....	83
2.8. Conclusion.....	87



## 2.1. Introduction

N'importe quel système peut dévier de son comportement nominal et tomber en panne ou fonctionner en mode dégradé. Généralement le comportement fautif est représenté par des événements non souhaités, inattendus et non observables appelés défauts. La présence de ces défauts dans un système engendre une baisse de qualité de service, et peut causer des dégâts matériels voire mettre en danger les opérateurs. Il est donc nécessaire de détecter l'occurrence des défauts en surveillant le système pour pouvoir intervenir à temps. Le diagnostic est le remède à ce problème. Il consiste à analyser les mesures issues des capteurs pour détecter l'occurrence de comportements fautifs. Dans ce chapitre nous nous intéressons au diagnostic des SED modélisés par des RdP temporels partiellement mesurés. Le diagnostic est basé sur le calcul de l'ensemble de trajectoires compatibles avec une trajectoire mesurée.

Un état de l'art sur le diagnostic des RdP temporisés et non temporisés est d'abord proposé. Par la suite, nous présentons les différents algorithmes exploités pour le calcul des probabilités de trajectoires compatibles avec les mesures datées. Ces probabilités sont déterminées en fonction des événements mesurés ainsi que leurs dates d'occurrences. L'exploitation de l'information temporelle a permis de discriminer les différents comportements du système et de déduire la probabilité d'occurrence des défauts structurels et temporels.

## 2.2. Modélisation des défauts

Dans ce qui suit, nous allons présenter les outils de modélisation des SED stochastiques affectés par des défauts. Un intérêt particulier est porté sur les RdP étiquetés ou labélisés (Labeled Petri Nets ou (LPN)), offrant à la fois une modélisation du fonctionnement du système ainsi que de ses capteurs, et permettant, dans le cas de leur extension temporelle d'évaluer la probabilité des défauts. Dans la grande majorité des travaux, les défauts sont représentés explicitement dans le modèle, ce qui implique une énumération a priori de tous les défauts candidats afin de les incorporer dans le modèle RdP. Dans notre cas, les défauts ne sont pas modélisés par des transitions spécifiques mais plutôt avec un ensemble de contraintes structurelles et temporelles qui doivent être satisfaites.

Nous nous intéressons par la suite aux RdP stochastiques temporisés dans lesquels la durée de tir des transitions est définie par une fonction de densité de probabilité arbitraire de support fini. Les capteurs qui sont utilisés pour collecter les mesures sont représentés par une fonction de mesure qui résulte du formalisme des RdP partiellement mesurés définis dans la sous-section qui suit. La modélisation des processus stochastiques arbitraires avec des RdP temporisés partiellement mesurés (Partially Observed Timed Petri Nets



ou (POTPN)) va nous permettre par la suite d'évaluer la probabilité des trajectoires qui sont compatibles avec une trajectoire de mesures temporisée, afin d'obtenir la probabilité de deux types de défauts ; défauts structurels et défauts temporels, auxquels on va s'intéresser dans la suite de ce travail.

### 2.2.1. Réseaux de Petri partiellement mesurés (POPn) et (POTPN)

En associant chaque transition du RdP à un événement, cette classe de RdP est dite étiquetée ou labélisée (LPN), de sorte que le franchissement d'une transition est soit mesuré soit silencieux. D'autres extensions considèrent une mesure du marquage de certaines places. Dans notre travail, on s'intéresse aux RdP partiellement mesurés tenant compte à la fois, d'une mesure (ou observation) partielle des marquages et des transitions. On utilise indifféremment les termes mesures et observations dans la suite.

Afin de représenter la configuration des capteurs du système, les RdP partiellement mesurés sont considérés avec les notions et notations nécessaires. Pour cela, on introduit la fonction de labélisation  $\mathcal{L} : \mathcal{T} \rightarrow E \cup \{\varepsilon\}$  qui associe un événement à chaque transition avec  $E = \{e_1, e_2, \dots, e_{q_0}\}$  l'ensemble de  $q_0$  étiquettes associées aux transitions mesurables du RdP, et  $\varepsilon$  représente l'absence de mesure associée aux transitions non observables.  $e_k$  est l'étiquette associée au tir de la transition mesurée  $T_j$  telle que  $\mathcal{L}(T_j) = e_k$  et dans le cas où la transition est silencieuse :  $\mathcal{L}(T_j) = \varepsilon$ . La concaténation des étiquettes satisfait :  $\varepsilon.\varepsilon = \varepsilon$  et  $\varepsilon.e_k = e_k.\varepsilon = e_k$ . Chaque étiquette  $e_k$  est représentée par un vecteur  $e_k = (e_{kj})$  avec  $e_{kj} = 0$  pour  $j \neq k$  et  $e_{kj} = 1$  pour  $j = k$ . L'étiquette nulle est représentée par le vecteur nul  $\varepsilon = 0_q$  de dimension  $q$ . La fonction de mesure  $\mathcal{L}$  peut-être ainsi codée par la matrice de mesure des transitions  $L = (l_{kj}) \in \{0,1\}^{q_0 \times q}$  avec  $l_{kj} = 1$  si la transition  $T_j$  est associée à l'étiquette  $e_k$  et  $l_{kj} = 0$  sinon. Un capteur de marquage est aussi introduit. La matrice de mesure des marquages est notée  $H = (h_{kj}) \in \{0,1\}^{n_0 \times n}$  où  $n_0$  représente le nombre de mesures partielles de marquage (tel que  $n_0 \leq n$ ) informant sur le nombre de jetons dans un sous ensemble de places.  $H = (h_{kj})$  tel que  $h_{kj} = 1$  si et seulement si le marquage de la place  $P_j$  est dans la  $k^{\text{ème}}$  mesure et  $h_{kj} = 0$  sinon. La partie mesurée de marquage est notée  $M^{obs}$  tel que  $M^{obs} = H \cdot M$ . Finalement, un RdP partiellement mesuré est défini comme suit :

**Définition 2.1.** La structure d'un RdP partiellement mesuré est définie par :  $G_o = \langle G, L, H \rangle$  où  $G$  est la structure d'un RdP ordinaire :  $G = \langle P, T, W_{PR}, W_{PO} \rangle$ .  $L$  et  $H$  sont les matrices de mesures des transitions et de marquages qui déterminent la configuration des capteurs du système.

Un RdP partiellement mesuré marqué, sera noté par  $\langle G_o, M_I \rangle$  avec  $M_I$  le marquage initial du réseau.

La fonction de mesure  $\Gamma$  capture les  $K$  mesures successives de marquages et d'événements d'une trajectoire de mesures non temporisée  $(\sigma_u, M)$ . Les mesures capturées grâce aux deux matrices  $L$  et  $H$  sont organisées suivant leur ordre d'occurrence, formant une suite de marquages et d'événements mesurés appelée trajectoire de mesures non temporisée  $TRo$  et donnée par (2.1) :

$$TRo = \Gamma(\sigma_u, M) = (M_0^{obs}) e_1^{obs} (M_1^{obs}) \dots e_{k-1}^{obs} (M_{k-1}^{obs}) e_k^{obs} (M_k^{obs}) \quad (2.1)$$

où  $e_j^{obs} \in E \cup \{\varepsilon\}$  est la  $j^{\text{ème}}$  étiquette mesurée,  $M_j^{obs}$  est le  $j^{\text{ème}}$  marquage mesuré et  $M_0^{obs} = H \cdot M$ .

$K$  est la taille de  $TRo$ .

Il est important de noter qu'une trajectoire  $(\sigma_u, M)$  qui satisfait  $\Gamma(\sigma_u, M) = TRo$  est dite compatible avec  $TRo$  et que l'ensemble des trajectoires compatibles avec  $TRo$  est noté  $\Gamma^{-1}(TRo)$ .

Finalement, les RdP temporisés partiellement mesurés sont définis comme suit :

**Définition 2.2.** Un RdP temporisé partiellement mesuré  $G_{oT}$  est défini par  $G_{oT} = \langle G, L, H, PDF \rangle$  où :  $G$  est la structure d'un RdP ordinaire :  $G = \langle P, T, W_{PR}, W_{PO} \rangle$ ,  $L$  et  $H$  sont les matrices de mesures des transitions et de marquages qui déterminent la configuration des capteurs du système et  $PDF$  est l'ensemble des fonctions de densité de probabilité qui définissent les durées de tir des transitions.

Lorsque le RdP temporisé partiellement mesuré est marqué, il sera noté  $\langle G_{oT}, M_I \rangle$  avec  $M_I$  le marquage initial du réseau.

La fonction de mesure  $\Gamma$  définie précédemment est appliquée à la trajectoire temporisée  $(\sigma, M)$  de taille  $K$  dans un intervalle de temps  $[\tau_0, \tau_{end}]$ , et les mesures sont organisées dans une trajectoire de mesures temporisées de la forme (2.2) :

$$TRo = \Gamma((\sigma, M), \tau_0, \tau_{end}) = M^{obs}(\tau_0) e^{obs}(\tau_1) M^{obs}(\tau_1) \dots e^{obs}(\tau_{K-1}) M^{obs}(\tau_{K-1}) e^{obs}(\tau_K) M^{obs}(\tau_{end}) \quad (2.2)$$

où  $M^{obs}(\tau_0) = H.M$ ,  $e^{obs}(\tau_j) \in E \cup \{\varepsilon\}$  est la  $j^{\text{ème}}$  étiquette mesurée à la date  $\tau_j$ ,  $M^{obs}(\tau_j)$  est le  $j^{\text{ème}}$  marquage mesuré à la date  $\tau_j$  et  $\tau_j, j = 0, \dots, K$  correspondent aux dates de mesures collectées dans l'intervalle de temps  $[\tau_0, \tau_{end}]$ .

Une trajectoire temporisée  $(\sigma, M)$  est dite compatible avec une trajectoire de mesures donnée  $TRo$  dans un intervalle de temps  $[\tau_0, \tau_{end}]$  si et seulement si elle satisfait :  $\Gamma((\sigma, M), \tau_0, \tau_{end}) = TRo$ . Plusieurs trajectoires de tailles différentes peuvent être compatibles avec  $TRo$ , c'est-à-dire que non seulement les événements et les marquages mesurés de  $(\sigma, M)$  correspondent à  $TRo$  mais aussi que les dates de mesures correspondent aux dates de tir prévues.  $\Gamma^{-1}(TRo)$  représente l'ensemble des trajectoires compatibles avec  $TRo$ .

### 2.2.2. Défauts structurels et temporels

Modéliser le dysfonctionnement du système dans notre cas, revient à considérer l'ensemble des types de défauts dont on va évaluer la probabilité.

Nous nous intéressons dans ce travail à des défauts structurels et temporels, et nous calculons leur probabilité en nous appuyons sur les mesures obtenues lors du fonctionnement du système.

#### A. Défauts structurels

Les défauts structurels, comme leur nom l'indique, correspondent à des séquences spécifiques d'événements qui doivent satisfaire des conditions de priorité définies par des motifs. Ces défauts structurels sont représentés par des *motifs de défauts*, c'est-à-dire des séquences ordonnées d'événements. Les *motifs de défauts* permettent de modéliser des comportements complexes, tel qu'un ensemble d'événements particuliers, la répétition d'événements et bien d'autres comportements qui ne pourraient pas être représentés avec un simple tir de transition.

Les motifs ont été introduits pour la première fois pour modéliser certaines spécifications du système. Ensuite, ils ont été utilisés dans le cadre de la supervision et du diagnostic des défauts [Jiang et Kumar 2006 ; Gougam et al. 2013]. Les *motifs de défauts* sont une extension des défauts structurels habituellement utilisés dans le diagnostic des SED, et qui sont modélisés par le tir d'une simple transition dans un RdP. Cette extension a une grande importance lorsqu'un système peut tomber en panne en raison de l'ordre spécifique d'un ensemble d'événements. Les événements du motif peuvent se suivre directement ou s'intercaler avec d'autres événements.

Avant de donner la définition formelle d'un *motif de défaut*, introduisons d'abords quelques opérateurs et notations nécessaires.

Soit " $\subset$ " l'opérateur d'inclusion,  $\rho$  et  $\sigma$  deux séquences de transitions. On dit que  $\rho$  est incluse dans  $\sigma$  (on note :  $\rho \subset \sigma$ ) si et seulement si on retrouve la sous-séquence  $\rho$  dans  $\sigma$ . On reformule cet opérateur d'inclusion sous l'expression mathématique suivante : ( $\rho \subset \sigma$ ) si  $\sigma = \sigma_1 \rho \sigma_2$  où  $\sigma_1, \sigma_2 \in T^*$  avec " $*$ " l'étoile de Kleene et  $T^*$  l'ensemble des séquences qui peuvent être générées avec l'ensemble des transitions  $T$ . On note que  $\sigma_1$  et  $\sigma_2$  peuvent éventuellement être des séquences vides.

Soit " $\ll$ " l'opérateur de concordance (matching) et  $\rho$  une séquence non nulle, on dit qu'une séquence  $\sigma$  est concordante (match) avec  $\rho$  et on note  $\rho \ll \sigma$  avec  $\rho \neq \varepsilon$ , si et seulement si  $\sigma$  contient toutes les transitions de  $\rho$ , de plus ces transitions sont tirées en respectant le même ordre. On peut reformuler mathématiquement l'expression d'un opérateur de concordance comme suit : ( $\rho \ll \sigma$  avec  $\rho \neq \varepsilon$ ) si  $\rho = \rho' s$  avec  $s \in T^*$ ,  $\rho' \in T^*$  et  $\exists \sigma_1 \in T^*$  et  $\sigma_2 \in T^*$  tel que  $\sigma = \sigma_1 s \sigma_2$  et  $\rho' \ll \sigma_1$ .

Un *motif de défaut* est défini comme suit :

**Définition 2.3.** Un *motif de défaut* est défini par l'ensemble  $\mathbf{R} = \{(\rho_1, o_1), \dots, (\rho_s, o_s)\}$  avec :

- $\rho_i, i = 1, \dots, s$  sont des sous-séquences de transitions
- $o_i, i = 1, \dots, s$  sont les opérateurs utilisés tels que  $o_i \in \{\subset, \ll\}$

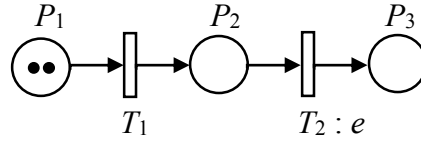
Pour modéliser le dysfonctionnement du système, on considère la fonction binaire  $F_R$ . Une séquence  $\sigma$  contient un défaut structurel modélisé par le motif  $\mathbf{R}$  si la fonction binaire  $F_R(\sigma)$  définie par l'équation (2.3) satisfait  $F_R(\sigma) = 1$ .

$$F_R(\sigma) = (\rho_1 o_1 \sigma) \wedge \dots \wedge (\rho_s o_s \sigma) \quad (2.3)$$

Dans les sections suivantes, notre objectif consiste à évaluer la probabilité d'un défaut structurel  $\text{Prob}(F_R / TRo)$  quand  $TRo$  est mesurée dans un intervalle de temps  $[\tau_0, \tau_{end}]$ .

**Exemple 2.1.** L'exemple de la figure 2.1 montre un RdP temporisé partiellement mesuré, composé de trois places et de deux transitions. Le marquage initial est donné par  $M_I = (2 \ 0 \ 0)^T$ . La transition  $T_1$  correspond à une transition silencieuse (qui ne délivre aucune mesure), alors que la transition  $T_2$  délivre l'étiquette  $e$ . Aucune place n'est mesurée.

Soit  $R = \{(T_1 T_1, <<)\}$  le motif de défaut considéré. Le tir répétitif de la transition  $T_1$  dans la séquence  $\sigma$  définit le défaut structurel auquel on s'intéresse :  $F_R(\sigma) = (T_1 T_1, <<)$  et donc  $F_R(\sigma) = 1$  si dans  $\sigma$  on retrouve deux tirs de la transition  $T_1$ . L'ensemble exhaustif des séquences non temporisées de tirs réalisables pour ce système est :  $\{T_1 ; T_1 T_1 ; T_1 T_2 ; T_1 T_1 T_2 ; T_1 T_2 T_1 ; T_1 T_1 T_2 T_2 ; T_1 T_2 T_1 T_2\}$  et donc seulement cinq séquences :  $\{T_1 T_1 ; T_1 T_1 T_2 ; T_1 T_2 T_1 ; T_1 T_1 T_2 T_2 ; T_1 T_2 T_1 T_2\}$  sont concordantes avec  $R$ .

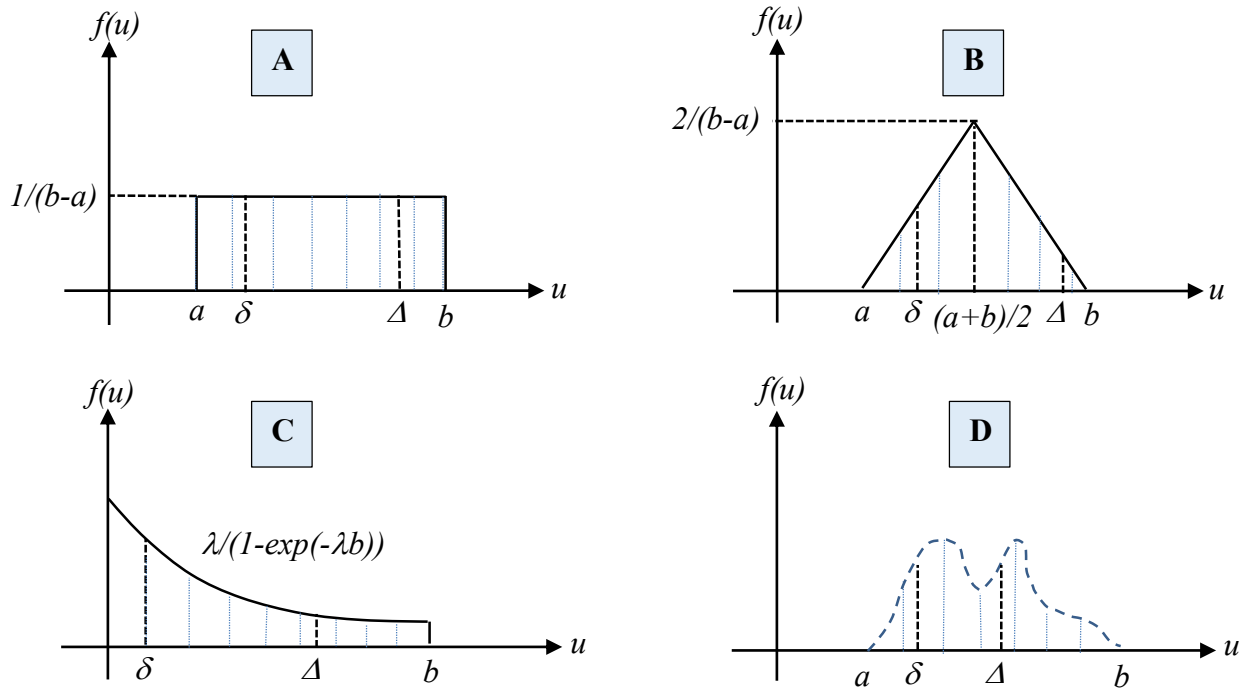


**Figure 2.1.** Exemple simple d'un RdP temporisé partiellement mesuré (POTPN)

## B. Défauts temporels

Dans cette sous-section, nous nous intéressons aux défauts temporels dans un RdP temporisé et partiellement mesuré. Nous travaillons sur un modèle stochastique markovien ou non markovien et nous définissons le modèle de défaut considéré comme étant la violation d'une contrainte temporelle. En effet, à chaque transition  $T$  est associé un intervalle de tolérance  $TC(T) = [\delta(T) : \Delta(T)]$  (figure 2.2) alors que les durées de franchissement sont définies par leur densité de probabilité (uniforme, triangulaire, exponentielle tronquée, ...) [Lefebvre 2017a ; Lefebvre 2017b].

Un défaut temporel survient lorsqu'une transition est franchie en dehors de son intervalle de tolérance  $[\delta(T) : \Delta(T)]$ , et on dira donc que le système satisfait les contraintes temporelles associées à la transition  $T$  tant que la durée de franchissement de  $T$  est comprise entre  $[\delta(T) : \Delta(T)]$ . Dans ce cas, l'opération qui est représentée par la transition  $T$  est effectuée pendant sa durée normale d'exécution. Autrement, un défaut sera détecté à cause de l'accomplissement trop tôt ou trop tard de cette opération.



**Figure 2.2.** Exemples de densités de probabilité des durées de tir des transitions : A) une densité de probabilité uniforme avec support fini ; B) une densité de probabilité triangulaire symétrique ; C) une densité de probabilité exponentielle tronquée ; D) une densité de probabilité arbitraire

Le but d'un système de diagnostic est de générer une alarme lorsque les contraintes temporelles ne sont plus respectées. On note  $TC$  l'ensemble des contraintes temporelles.

Une activité de durée  $d$  connue exactement est représentée par une transition  $T$  avec l'intervalle de tolérance  $[d : d]$ . Alors qu'une activité sans contrainte temporelle est représentée par une transition  $T$  avec un intervalle de tolérance  $[\delta : \Delta]$  égale à  $[a : b]$  où  $a$  et  $b$  définissent le support de la densité de probabilité de cette transition. Une séquence  $\sigma$  contient un défaut temporel  $F_{TC}$  s'il existe au moins une transition dans la séquence  $\sigma$  dont la durée de tir ne satisfait pas sa contrainte temporelle figurant dans  $TC$ . Par conséquent, la fonction binaire  $F_{TC}$  est utilisée pour modéliser ce nouveau type de dysfonctionnement du système pour chaque séquence temporisée  $\sigma$  de la forme  $\sigma = T(t_1) T(t_2) \dots T(t_H)$ . Cette fonction est définie par l'équation (2.4) :

$$F_{TC}(\sigma) = (d_1 \notin TC(T(t_1))) \vee \dots \vee (d_H \notin TC(T(t_H))) \quad (2.4)$$

où  $d_H$  est la durée de franchissement de la transition  $T(t_H)$ .

La fonction  $F_{TC}(\sigma)$  satisfait (2.5) :

$$\begin{aligned}
 F_{TC}(\sigma) &= 1, \text{ si au moins un défaut temporel se produit dans la séquence } \sigma \\
 F_{TC}(\sigma) &= 0, \text{ sinon}
 \end{aligned}
 \tag{2.5}$$

Dans ce qui suit, on s'intéresse à évaluer la probabilité  $\text{Prob}(F_{TC} / TRo)$  qu'un défaut temporel  $F_{TC}$  se soit produit dans une trajectoire de mesures  $TRo$  durant l'intervalle de temps  $[\tau_0, \tau_{end}]$ .

**Exemple 2.2.** On considère l'exemple de la figure 2.1 avec un marquage initial  $M_I = (2 \ 0 \ 0)^T$ . L'étiquette  $e$  est collectée lors du franchissement de la transition  $T_2$ . Supposons que la durée de franchissement est définie par une densité de probabilité uniforme de support  $a_2 = 1$  et  $b_2 = 4$ . La transition  $T_1$ , quant à elle, ne délivre aucune étiquette (silencieuse) avec une densité de probabilité de même type que  $T_2$  et un support défini par  $a_1 = 0$  et  $b_1 = 10$ . On considère que les places ne sont pas mesurées et que l'ensemble des défauts temporels est :  $TC(T_1) = [2 : 8]$  et  $TC(T_2) = [2 : 3]$ . La fonction binaire  $F_{TC}(\sigma) = 1$  si  $(d_1 \notin [2 : 8]) \vee (d_2 \notin [2 : 3])$ .

## 2.3. Etat de l'art

### 2.3.1. Diagnostic des RdP non temporisés

Il est apparu durant nos recherches sur le diagnostic des SED, que dans la littérature ce problème de diagnostic a été largement étudié par le formalisme des systèmes de transitions. Pour les automates à états finis, deux catégories principales de méthodes peuvent se présenter à savoir : celles à base d'événements [Lin 1994 ; Sampath et al. 1995 ; Sampath et al. 1996] et celles à base d'états [Zad et al. 2003]. Pour les RdP [Zaytoon et Lafortune 2013], différentes approches ont aussi été développées. Celles-ci dépendent entre autres de l'extension des RdP considérés pour modéliser les systèmes (non temporisés, temporels, stochastiques ...) ainsi que la classification du type de mesures traitées. Cette classification se fait du point de vue des mesures complètes ou partielles du tir des transitions et du marquage.

La modélisation par les RdP reste très avantageuse par rapport à d'autres outils, grâce à sa représentation graphique et mathématique [Murata 1989]. L'outil RdP rend plus facile la modélisation des systèmes complexes dans différents domaines et grâce à la manière dont il résout les problèmes de parallélisme et synchronisations.

Globalement les approches de diagnostic par l'outil RdP peuvent être classées dans deux catégories principales :

## A. Méthodes structurelles

Ces méthodes exploitent les propriétés structurelles du RdP. Une méthode limitée à l'identification des erreurs de capteur ou de processus a été proposée suite à une variation du nombre de jetons dans un P-invariant [Prock 1991]. Les auteurs de [Wen et al. 2005] ont proposé une approche pour analyser la diagnosticabilité sous une condition suffisante qui suppose la mesure partielle du marquage et des événements. Ils établissent une condition suffisante pour la diagnosticabilité calculée à travers un algorithme polynomial basé sur la programmation linéaire. Par conséquent, une analyse de la diagnosticabilité a été effectuée en se basant sur les propriétés structurelles. Les auteurs de [Ramírez-Treviño et al. 2007] proposent, quant à eux, un algorithme polynomial pour vérifier la diagnosticabilité dans des RdP interprétés. Ils considèrent des mesures partielles d'événements et du marquage. Le diagnostic des défauts ainsi que leur localisation sont assurés grâce à un algorithme réalisant une comparaison entre la sortie du diagnostiqueur et celle du système afin de détecter une déviation du comportement. Cette approche repose également sur la structure du modèle, permettant ainsi le diagnostic du défaut. L'idée générale des approches se basant sur les représentations structurelles des RdP, est de développer un diagnostiqueur interprété. Ce dernier permet de déduire le comportement du système en se basant sur les événements observables et la programmation linéaire en nombres entiers (Integer Linear Programming (ILP)) [Basile et al. 2009, Dotoli et al. 2009]. Il effectue des diagnostics en utilisant des observations en ligne du comportement du système. Ce type de diagnostiqueur est également utilisé pour énoncer et vérifier les conditions hors ligne nécessaires et suffisantes pour la diagnosticabilité. Dans une série de travaux [Basile et al. 2008, Basile et al. 2009, Basile et al. 2012], les auteurs introduisent la notion de g-marking qui représente les marquages du réseau pouvant comporter des composantes négatives. Ces composantes négatives sont dues à la présence de transitions silencieuses. L'obtention de ce g-marking est assurée par la mise à jour du marquage avec un événement mesuré en ayant recours à l'équation d'état. Deux conditions suffisantes ont été établies concernant la diagnosticabilité. Dans le travail [Basile et al. 2008], l'une d'entre elles est jugée nécessaire afin de détecter l'occurrence d'une faute. L'utilisation de la programmation linéaire en nombres entiers en ligne afin de décider si le comportement du système est normal ou s'il présente quelques fautes, a été détaillée dans le travail de [Dotoli et al. 2009]. Cette méthode de détection de défauts en ligne dans une architecture centralisée est étendue dans [Cong et al. 2017] aux architectures décentralisées. En se basant sur les RdP, et étant donné un modèle de supervision, est-il toujours possible pour ce superviseur de déterminer avec certitude que le défaut s'est produit après avoir observé une séquence finie d'événements ? Plusieurs travaux ont été proposés pour répondre à cette question [Jiang et Kumar 2004, Jéron et al. 2006]. Les auteurs dans [Gougam et al. 2013, Gougam et al. 2014] s'attaquent à ce problème en exploitant les



bénéfices de la méthode de dépliage (unfolding) afin de détecter les motifs de défauts qui sont représentés par l'occurrence d'une suite d'événements traduisant un comportement fautif du système.

## **B. Méthodes basées sur la dynamique du RdP**

Afin d'étendre les approches de diagnostic des défauts en utilisant les RdP, la représentation des comportements peut être très avantageuse et permet ainsi grâce à la construction d'un graphe à partir du modèle du système de représenter de manière compacte l'espace d'état du système en se basant sur les mesures. Les approches basées sur les diagnostiqueurs [Sampath et al. 1995, Zad et al. 2003] permettent ainsi de s'attaquer au problème de diagnosticabilité et du diagnostic en ligne. Elles ont été étendues afin de pouvoir être utilisées avec les modèles RdP. Dans [Ushio et al. 1998], les auteurs proposent deux types de diagnostiqueurs : w-diagnostiqueurs et w-diagnostiqueurs raffinés (w- and w-refined diagnosers). Ils considèrent que le marquage des places est mesuré contrairement aux transitions qui sont non observables avec la possibilité d'occurrence de défauts associés à certaines d'entre elles. Des conditions suffisantes ont été proposées par les auteurs de cette approche pour pallier ce problème de la diagnosticabilité en se basant sur le graphe de marquages. Dans [Yoo et Lafortune 2002 ; Jiang et al. 2001], les auteurs présentent leurs approches basées sur la construction d'un vérificateur afin d'analyser la diagnosticabilité. Dans [Jiroveanu et Boel 2010], les auteurs utilisent un automate-ROF pour tester la diagnosabilité dans les RdP bornés. Cet automate est construit en se basant sur le calcul des explications minimales de défauts. Celles-ci correspondent aux séquences d'événements non observables ayant une longueur minimale qui permettent ainsi d'expliquer le franchissement d'événements mesurés, et qui servent à générer un espace d'état réduit par rapport au graphe de marquages dit graphe d'accessibilité (Reachability Graph ou RG). Une technique se basant sur ce type de graphe a été d'abord introduite dans [Giua et Seatzu 2005 ; Cabasino et al. 2010] pour la détection de défauts dans un RdP avec des transitions non mesurées. Les auteurs utilisent cette technique afin de construire une partie réduite du graphe de marquages. Le calcul de l'ensemble des marquages de base est ainsi déterminé par un algorithme itératif qui permet par la suite de définir un automate déterministe dit graphe d'accessibilité de base (Basis Reachability Graph ou BRG). Ce dernier peut être considéré comme un diagnostiqueur. En se basant sur leur propre travail [Cabasino et al. 2009a ; Cabasino et al. 2010], les auteurs présentent dans [Cabasino et al. 2009b] une approche pour le diagnostic des défauts dans les RdP bornés où l'analyse de la diagnosticabilité se fait à travers deux graphes : le graphe d'accessibilité de base modifié (Modified Basis Reachability Graph ou (MBRG)) et le diagnostiqueur d'accessibilité de base (Basis Reachability Diagnoser

ou (BRD)). Bien que la taille de ces deux graphes soit inférieure à celle du RG, la procédure proposée pour construire MBRG peut nécessiter un nombre d'étapes égal à celui nécessaire pour construire le RG. Les conditions nécessaires et suffisantes pour la diagnosabilité des RdP bornés sont présentées dans ce travail. Dans [Boussif 2016], l'auteur propose une variante de diagnostiqueur avec une nouvelle structure afin d'analyser la diagnosticabilité des fautes permanentes. Il étend ce diagnostiqueur pour traiter les fautes intermittentes. Ensuite il propose de nouvelles techniques pour analyser la diagnosticabilité de ces deux types de défauts dans le cadre de vérification par model-checking. Une comparaison des caractéristiques de cette approche dans [Boussif et al. 2017] avec trois autres approches de diagnostiqueur à savoir l'approche MBRG/BRD [Cabasino et al. 2014], l'approche FMG/FMSG [Liu et al. 2017] et l'approche d'un diagnostiqueur semi-symbolique (semi-symbolic diagnoser ou (SSD)) [Boussif et al. 2015] a été menée.

Toujours dans le cadre d'un diagnostic en ligne, dans [Lefebvre 2014a], l'auteur propose une méthode basée sur les séquences d'observations sous une mesure partielle des marquages et des transitions. L'idée derrière ce travail est de déterminer tous les comportements compatibles avec la trajectoire de mesures en s'appuyant sur la programmation linéaire en nombre entiers (Integer Linear Programming ou (ILP)). Avec la même hypothèse de travail, une mesure partielle des transitions et des marquages, [Lefebvre et Leclercq 2015] proposent une approche pour la diagnosticabilité des RdP-bornés et non bornés. Leur méthode est basée sur la transformation du graphe de couverture en un graphe de marquages qui code toutes les séquences d'observations de marquage et d'événements mesurés en respectant la configuration des capteurs du système. L'objectif essentiel derrière l'utilisation de ces graphes, est de pouvoir représenter toutes les séquences d'observations collectées avec une fonction de mesure donnée afin de déterminer toutes les séquences de tir d'itérations compatibles avec les séquences d'observations. Des conditions nécessaires et suffisantes découlant de la conception et de l'analyse de ces graphes ont été fournies. Le même formalisme est exploité dans [Ru et Hadjicostis 2009], où les auteurs transforment des RdP partiellement observables (mesure partielle des marquages et des transitions) en des RdP labélisés. Ils calculent par la suite un degré de croyance (belief) entre le nombre de trajectoires compatibles avec une trajectoire de mesures, et celui des trajectoires compatibles contenant une trace de défauts. Plus récemment, et en exploitant la représentation mathématique des RdP labélisés avec des transitions indiscernables, [Basile et al. 2012] proposent l'utilisation d'ILP pour l'analyse hors ligne de la K-diagnosticabilité. Ce concept de K-diagnosticabilité repose sur la diagnosabilité dans un délai fini. C'est-à-dire K événements après l'occurrence de la faute. Dans [Lefebvre et Delherm 2007], les auteurs caractérisent et proposent les conditions qui doivent être satisfaites par l'ensemble des places qu'il faut observer pour détecter les différentes fautes associées à certaines transitions. L'avantage de leur approche est

la conception des diagnostiqueurs minimaux qui déterminent et localisent les transitions défauts sans devoir élargir le graphe de marquages du modèle. Récemment, une approche dite à la volée (on-the-fly) a été proposée par les auteurs de [Liu et al. 2014 ; Liu 2014] pour l'analyse de la diagnosticabilité et la  $K/K_{\min}$  diagnosticabilité (c.à.d. qu'une faute peut être déduite après un délai donné et fini de  $K$  événements après son occurrence avec une valeur minimale de  $K$ , pour assurer la diagnosabilité). L'approche consiste à construire deux types de graphes : un graphe de marquage de défauts (Fault Marking graph or FM-graph) qui permet d'enregistrer certains marquages spécifiques avec leurs informations respectives d'occurrence de défaut. Ce type de graphe peut être utilisé pour réaliser le diagnostic en ligne dans le cas où les modèles RdP sont diagnosticables. Et plus récemment, et en se basant sur les approches dites à la volée, les auteurs dans [Liu et al. 2017] ont proposé une méthode pour analyser la diagnosabilité au cours de la phase de la construction du diagnostiqueur. Cette méthode est applicable pour le diagnostic en ligne des RdP labélisés. Le point clé de ces approches « on-the-fly » est que l'analyse de seulement une partie du graphe de marquages est souvent suffisante pour donner un résultat de diagnostic. Des améliorations de cette approche ont été présentées dans le travail de [Li et al. 2015a] en utilisant les T-invariants ou encore en utilisant les explications minimales [Li et al. 2015b].

Pour un survol de la littérature relative aux problèmes de diagnostic des SED, il est conseillé de voir les travaux suivants : [Basile et al. 2014 ; Zaytoon et al. 2013].

### 2.3.2. Diagnostic des RdP temporels

Le problème de diagnostic et de la diagnosticabilité des SED défini d'une manière qualitative à base d'automates à états finis dans [Lin 1994 ; Sampath et al. 1995 ; Sampath et al. 1996] s'est focalisé dans un premier temps sur la capacité à vérifier si toutes les fautes sont diagnosticables dans un délai fini et pour un système donné. Dans sa version standard, et en supposant un système dynamique avec deux types de comportements : l'un normal et l'autre fautif, le diagnostic avec un degré d'observabilité assez limité, ne permet pas toujours de différencier certaines séquences d'événements caractérisant des défauts par rapport à d'autres traduisant des comportements normaux du système. Cette ambiguïté est justifiée par un manque d'informations sur les dates d'occurrences des événements, d'où des extensions temporelles des travaux initiaux pour étudier le diagnostic en exploitant le formalisme des automates temporisés [Chen et Provan 1997 ; Tripakis 2002 ; Bouyer et al. 2005 ; Zad et al. 2005 ; Jiang et Kumar 2006]. D'autres travaux s'attaquent au même problème en exploitant le formalisme des RdP. Celui-ci se distingue par sa capacité à

---

traiter des informations simultanées, asynchrones, parallèles, ou encore distribuées dans un contexte déterministe ou stochastique.

Depuis 1970, plusieurs extensions ont été proposées pour étudier les contraintes temporelles dans les modèles des RdP. Diverses applications ont joué le rôle de catalyseur pour prendre en compte l'aspect temporel. L'objectif principal est de décrire les durées nécessaires à l'accomplissement des opérations. Dans un premier temps, les modèles de RdP temporisés ont été introduits par Ramchandani (Timed PNs) [Ramchandani 1973] et par Merlin (Time PNs) [Merlin 1975]. Bien que les appellations se ressemblent, ces modèles sont différents par deux aspects : (a) les types de contraintes temporelles (b) les éléments du réseau auxquels ces contraintes sont associées.

Les contraintes temporelles dans le travail de [Ramchandani 1973] sont décrites comme étant des durées minimales de sensibilisation des transitions, cela veut dire que ces transitions peuvent être franchies ou non à partir de ces durées minimales. Dans ce travail, ces durées minimales sont associées aux transitions et non aux places (RdP T-temporisés), contrairement au travail de [Sifakis 1980], qui introduit une autre définition des RdP temporisés en associant des durées aux places du réseau (RdP P-temporisés). Une extension de l'approche présentée dans le travail de [Ramchandani 1973] montre le pouvoir des RdP temporels pour évaluer les performances de deux architectures de processeurs simples [Zuberek 1980].

D'autres modèles, nommés RdP temporels (Time PNs) [Merlin 1975], représentent les contraintes temporelles sous forme d'intervalle. Le franchissement d'une transition appartenant à ces modèles s'effectue après une durée incluse dans cet intervalle dont la borne inférieure et supérieure correspondent respectivement à la durée minimale et la durée maximale de franchissement. Lorsqu'on spécifie l'association de ces contraintes temporelles aux transitions du réseau, on parle de RdP T-temporels, et de RdP P-temporels si elles sont associées aux places. Le réseau peut être soit déterministe ; c'est-à-dire que la durée de sensibilisation de la transition avant son possible franchissement est connue ; soit stochastique, dans ce cas cette durée est une variable aléatoire avec une densité de probabilité connue. Il faut rappeler que cette durée, quel que soit son type, peut aussi être associée aux places du réseau plutôt qu'aux transitions.

Une prise en compte des contraintes temporelles dans le diagnostic des RdP T-temporels [Ghazel 2005] ne peut être obtenue par l'analyse du graphe d'états. L'auteur dans [Ghazel 2005] utilise plutôt des classes d'états. L'ensemble des classes formées de marquages ainsi que des contraintes temporelles liées aux franchissements

des transitions sensibilisées d'une classe vers une autre, constitue le graphe des classes d'états (State Class Graph (SCG)). L'analyse d'un SCG permet de conclure sur la diagnosabilité du système et de fournir les conditions nécessaires et suffisantes pour que ce dernier soit diagnosticable [Berthomieu et Menasche 1983 ; Berthomieu et Diaz 1991]. On note que cette méthode a ensuite été améliorée avec le graphe des classes d'états atomiques (Atomic States Class Graph (ASCG)) dans [Berthomieu et Vernadat 2003]. Dans [Jiroveanu et al. 2006 ; Boel et Jiroveanu 2013], les auteurs utilisent le même formalisme et proposent une approche basée sur une classe d'états énumérative. Cette technique consiste à calculer un SCG, ensuite elle calcule les chemins dans le graphe afin de déterminer les états de diagnostic. Le graphe de diagnostic des défauts (Fault Diagnosis Graph (FDG)), inspiré du graphe des classes, constitue la contribution de l'approche de [Wang et al. 2013 ; Wang et al. 2015] employée afin de détecter le tir des transitions fautives en observant le tir de celles qui sont observables. On note que dans ces travaux, certaines transitions non observables décrivent un comportement normal tandis que d'autres représentent un comportement fautif et que les classes d'états et les séquences de tirs des transitions ne sont pas calculées si elles ne sont pas nécessaires au diagnostic de défaut. On en déduit que cette limitation rend le nombre de classes d'états dans le FDG inférieur à celui du SCG, ce qui représente un gain en termes de complexité. Ainsi la mise à jour de chaque état courant à chaque nouvelle mesure permet de détecter le défaut. Diagnostiquer un système sans devoir nécessairement explorer tout l'espace d'état est l'un des points forts d'une approche basée sur les algorithmes à la volée. L'analyse à la volée de la diagnosticabilité est très avantageuse face aux approches classiques d'énumération systématique de tous les états. Cette technique est utilisée dans [Liu et al. 2013], afin d'analyser la diagnosticabilité des SED modélisés par des RdP T-temporels et labellisés. Les auteurs proposent ainsi la construction d'un observateur temporel appelé Graphe des Classes d'Etats Enrichies (GCEE). Le GCEE permet de tirer les informations nécessaires pour la diagnosticabilité d'un système ainsi que calculer le délai minimum ( $\Delta_{\min}$ ) qui garantit cette diagnosticabilité. Le SCG introduit dans [Berthomieu et Diaz 1991] ne cesse d'inspirer les auteurs ayant comme objectif de diagnostiquer les défauts dans un RdP temporel. Dans [Basile et al. 2015] qui étend les résultats de [Basile et al. 2013], cette inspiration a donné lieu à un graphe des classes modifié (Modified State Class Graph (MSCG)) permettant une représentation exhaustive de l'évolution d'un système temporisé en fonction des observations datées. L'exploration d'un MSCG a permis d'obtenir une estimation d'état à un temps donné et de réaliser le diagnostic en ligne des défauts dans un système temporisé en se basant sur les mesures datées.

Dans la littérature, le diagnostic des défauts dans un contexte temporel, stochastique et sous une observation partielle des événements a été récemment introduit dans [Lefebvre 2014b], tout d'abord, dans un contexte

markovien des RdPS, (c'est-à-dire avec uniquement des densités de probabilité exponentielles), ensuite en appliquant le même formalisme avec des densités de probabilité non exponentielles. Ces derniers sont désignés par les RdPS non-markoviens [Lefebvre et al. 2018a ; Lefebvre et al. 2018b]. D'une part, l'idée derrière ce travail est de pouvoir discerner les trajectoires des événements mesurés représentant le comportement normal du système et celles fautives. Souvent, différencier ces deux types de trajectoires n'est pas possible si les informations temporelles ne sont pas prises en compte. Grâce aux dates de mesures et au calcul de la probabilité des trajectoires datées [Ammour et al. 2015 ; Ammour et al. 2017a ; Amour et al. 2017b ; Lefebvre 2017b] il a été montré qu'il était possible de distinguer les deux comportements (normal et fautif). Par conséquent la probabilité des fautes est établie. Ce constat a été étendu aux réseaux non markoviens dans [Lefebvre 2017a ; Lefebvre 2017b] et constitue le contexte de notre travail de thèse.

## 2.4. Calcul des trajectoires compatibles

L'objectif de cette section est de déterminer tous les comportements du système cohérents avec des mesures collectées. Le but est donc de déterminer les trajectoires compatibles avec une trajectoire de mesures obtenue lors du fonctionnement du système. L'analyse de ces trajectoires devra fournir des informations sur l'occurrence d'éventuels défauts, permettant ainsi un diagnostic du système en fonction des mesures collectées. Dans un premier temps il faut déterminer l'ensemble des trajectoires compatibles dans un cadre non temporisé, puis dans le but d'évaluer la probabilité de chaque trajectoire compatible, il faudra considérer l'information temporelle, cas des RdP temporisés.

### 2.4.1. Trajectoires compatibles non temporisées

La définition d'un RdP partiellement mesuré marqué  $\langle G, L, H, M_I \rangle$  est donnée par la définition 2.1, la trajectoire de mesures ( $TRo$ ) donnée par l'expression (2.1) est rappelée ci-dessous :

$$TRo = (M_0^{obs}) e_1^{obs} (M_1^{obs}) \dots e_{k-1}^{obs} (M_{k-1}^{obs}) e_k^{obs} (M_k^{obs}) \quad (2.6)$$

où  $e_j^{obs} \in E \cup \{\varepsilon\}$  correspond au  $j^{\text{ème}}$  événement mesuré,  $M_j^{obs}$  est le  $j^{\text{ème}}$  marquage mesuré et  $M_0^{obs} = H.M$ .  $M$  correspond au marquage à partir duquel les mesures ont été enregistrées, et est supposé inconnu.

Plusieurs (et parfois une infinité) trajectoires sont compatibles avec la trajectoire de mesures. Une séquence de mesure est de type élémentaire (Elementary Observation Sequence ou (EOS)) si elle est de taille  $K = 1$  :

$(M_{k-1}^{obs}) e_k^{obs} (M_k^{obs})$ . L'hypothèse suivante est introduite afin d'éviter les séquences de tir ayant un nombre infini d'événements.

**Hypothèse H2.1.** La partie silencieuse (non mesurée) du graphe de marquages est acyclique.

L'hypothèse H2.1 est suffisante pour assurer que la taille des séquences non observables du système qui ne modifient pas la partie observable du marquage est bornée. On note que cette hypothèse peut être assurée grâce à la configuration des capteurs. Par conséquent, le nombre maximal d'événements silencieux qui peuvent être tirés sans modifier le marquage mesuré est limité par le nombre  $h_{max} - 1$  [Lefebvre 2014b]. Ainsi,  $h_{max}$  est définie comme étant la taille maximale d'une séquence de la forme :

$$\sigma_{Um} = T(1) T(2) \dots T(h_{max} - 1) T(h_{max}) \quad (2.7)$$

où  $T(j)$ ,  $j = 1, \dots, h_{max} - 1$  correspond à la  $j^{\text{ème}}$  transition silencieuse et  $T(h_{max})$  correspond à l'événement qui produit une étiquette et/ou un changement dans le marquage mesuré. Le paramètre  $h_{max}$  est défini comme la limite supérieure de  $h$ , il est calculé par l'expression suivante :

$$h_{max} = \min \{h \geq 0 \text{ tel que } (A_\varepsilon)^h = 0\} \quad (2.8)$$

$A_\varepsilon$  représente la matrice relative à la partie non mesurée du graphe de marquages. Ce graphe est défini comme étant un graphe dans lequel toutes les transitions observables dont le tir peut engendrer un changement du marquage mesuré sont éliminées. Cette matrice est définie par  $A_\varepsilon = (a_{\varepsilon ik}) \in \{0, 1\}^{N \times N}$  où  $N$  est le nombre d'états du graphe de marquages. Cette matrice est définie comme suit :

$$\begin{cases} a_{\varepsilon ik} = 1 & \text{si } H(M_i) = H(M_k) \equiv T \text{ est silencieuse entre } M_i \text{ et } M_k \\ a_{\varepsilon ik} = 0 & \text{sinon} \end{cases} \quad (2.9)$$

On note que toute séquence de tirs compatible avec une EOS et qui se termine par un événement observable et/ou un changement dans le marquage mesuré est dite (Elementary Firing Sequence (EFS)).

On note aussi que la complexité de calcul pour trouver  $h_{max}$  dépend d'abord de la cardinalité de l'ensemble des états atteignables dans des RdP bornés. De ce fait, l'hypothèse suivante est retenue :

**Hypothèse H2.2.** L'ensemble des états atteignables depuis le marquage initial  $M_I$  :  $ReachSet(G, M_I)$  est supposé connu et de cardinalité finie.

Deux approches sont présentées par la suite dans le but de caractériser l'ensemble des comportements possibles du système dont l'observation coïncide avec une trajectoire de mesures et respectant les deux hypothèses H2.1 et H2.2.

Une première méthode développée dans [Lefebvre 2014a] permet d'obtenir l'ensemble des trajectoires compatibles à partir de la résolution d'un ensemble d'inégalité matricielle linéaire (Linear Matrix Inequality (LMI)). La deuxième approche, dite incrémentale [Ammour et al. 2017b], présente une complexité réduite par rapport à la précédente.

### A. Approche par LMI

L'ensemble  $\Gamma^{-1}(TRo)$  des trajectoires compatibles avec la trajectoire de mesures  $TRo$  donnée par (2.1) est obtenu en résolvant un ensemble de LMI. Pour se faire, on considère le couple  $(\sigma_u, M)$  tel que  $(\sigma_u, M) \in \Gamma^{-1}(TRo)$ . Une séquence de tirs  $\sigma_u$  est définie en tenant compte de la borne  $h_{max}$ , comme suit :

$$\begin{aligned} \sigma_u = & T(1,1) \dots T(1,h_1-1)\mathbf{T}(\mathbf{1},\mathbf{h}_1)T(2,1) \dots T(2,h_2-1)\mathbf{T}(\mathbf{2},\mathbf{h}_2) \\ & \dots T(K,1) \dots T(K,h_K-1)\mathbf{T}(\mathbf{K},\mathbf{h}_K)T(K+1,1) \dots T(K+1,h_{K+1}-1) \end{aligned} \quad (2.10)$$

Où  $K$  est la taille de  $TRo$ ,  $h_j-1$  est le nombre d'événements silencieux entre la  $(j-1)^{\text{ème}}$  et la  $j^{\text{ème}}$  mesure et  $T(j, h_j)$  correspond soit à l'événement observable et/ou à celui qui produit un changement de marquage mesuré avec  $j = 1, \dots, k$  et  $h_j \leq h_{max}$ . Une clôture silencieuse peut avoir lieu dans une trajectoire après le franchissement du dernier événement mesuré. Elle est de type  $T(K+1,1) \dots T(K+1,h_{K+1}-1)$ .

Après avoir défini la séquence  $\sigma_u$ , la proposition suivante nous permet de calculer l'ensemble  $\Gamma^{-1}(TRo)$  grâce à la résolution par LMI. On note que ce dernier est de cardinalité finie.

**Proposition 2.1.** On considère un SED modélisé par un RdP partiellement mesuré  $\langle G, M_I \rangle$  qui satisfait les deux hypothèses H2.1 et H2.2 et instrumenté selon la configuration de capteurs  $(L, H)$ . Etant donné une trajectoire de mesures  $TRo$  de la forme (2.1) et une trajectoire  $(\sigma_u, M)$  avec  $\sigma_u$  de la forme (2.10), la trajectoire  $(\sigma_u, M)$  est dite compatible avec  $TRo$  si et seulement si :



1)  $H.M = M_0^{obs}$

2)  $\sigma_u$  et  $M$  satisfont l'inégalité matricielle (2.11), l'égalité (2.12) pour  $k = 1, \dots, K$  et (2.13)

$$\begin{pmatrix} -I_q & 0 & \dots & 0 & 0 \\ 0 & -I_q & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & -I_q & 0 \\ (1_q)^T & 0 & \dots & 0 & 0 \\ 0 & (1_q)^T & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & (1_q)^T & 0 \\ W_{PR} & 0 & \dots & 0 & -I_n \\ -W & W_{PR} & \ddots & \vdots & -I_n \\ \vdots & \ddots & \ddots & 0 & \vdots \\ -W & \dots & -W & W_{PR} & -I_n \end{pmatrix} \cdot \begin{pmatrix} X(T(1,1)) \\ \vdots \\ X(T(1, h_1)) \\ X(T(2,1)) \\ \vdots \\ X(T(2, h_2)) \\ \vdots \\ X(T(K, h_K)) \\ M \end{pmatrix} \leq \begin{pmatrix} 0_q \\ 0_q \\ \vdots \\ 0_q \\ 1 \\ 1 \\ \vdots \\ 1 \\ 0_n \\ 0_n \\ \vdots \\ 0_n \end{pmatrix} \quad (2.11)$$

$$\begin{pmatrix} L & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \ddots & L & 0 \\ 0 & \dots & 0 & L \\ H.W & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & H.W & 0 \\ H.W & \dots & H.W & H.W \end{pmatrix} \cdot \begin{pmatrix} X(T(k, 1)) \\ \vdots \\ X(T(k, h_k - 1)) \\ X(T(k, h_k)) \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon \\ e_k^{obs} \\ 0_{no} \\ \vdots \\ 0_{no} \\ M_k^{obs} - M_{k-1}^{obs} \end{pmatrix} \quad (2.12)$$

$$\begin{pmatrix} L & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L \\ H.W & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & H.W \end{pmatrix} \cdot \begin{pmatrix} X(T(K+1, 1)) \\ \vdots \\ X(T(K+1, h_{K+1} - 1)) \end{pmatrix} = \begin{pmatrix} \varepsilon \\ \vdots \\ \varepsilon \\ 0_{no} \\ \vdots \\ 0_{no} \end{pmatrix} \quad (2.13)$$

**Démonstration.** La démonstration de la proposition 2.1 est expliquée en détail dans [Lefebvre 2012 ; Lefebvre 2014a ; Lefebvre 2014b].

Pour des séquences de tir de la forme :  $\sigma_u = T(1,1) \dots T(1,h_1)T(2,1) \dots T(2,h_2)T(K,1) \dots T(K,h_K)$ , des transitions supplémentaires  $T(K+1,1) \dots T(K+1,h_{K+1}-1)$  ont été considérées, formant ainsi la séquence de la forme (2.10). L'égalité (2.12) concerne la clôture silencieuse. Elle est satisfaite pour le vecteur de tirs  $X(T(K+1,1)) = X(T(K+1,h_{K+1}-1)) = 0$  et garantit qu'aucune étiquette mesurée n'est générée, et qu'aucune transition silencieuse  $T(K+1,1) \dots T(K+1,h_{K+1}-1)$  ne provoque de changement dans le marquage mesuré.

Un vecteur de franchissements  $X(T(k,m))$  est associé à chaque transition de  $\sigma_u$  avec  $k = 1, \dots, (K+1)$  et  $m = 1, \dots, h_{K+1}-1$ . Considérons les séquences de transitions  $\sigma_u(k) = T(k,1) \dots T(k,h_k-1)T(k,h_k)$  de  $\sigma_u$  avec  $k = 1, \dots, K$  qui vérifient (2.11) et (2.6). Selon les lignes 1 à 8 de l'inégalité (2.11), chaque vecteur  $X(T(k',m'))$  avec  $k' = 1, \dots, K$  et  $m' = 1, \dots, h_K$  est à composantes positives et ne contient au plus qu'une seule valeur non nulle, égale à 1 ; autrement dit ces vecteurs peuvent représenter soit le franchissement d'une transition unique, si le vecteur contient une composante égale à 1, soit le fait qu'aucune transition n'est franchie. Enfin, les lignes 9 à 12 signifient l'existence d'un marquage  $M \in ReachSet(G, M_I)$  à partir duquel la séquence  $\sigma_u(k)$  est réalisable. De plus, le marquage  $M$  satisfait  $H.M = M_0^{obs}$ . Les lignes 1 à 3 de (2.6) montrent effectivement que les  $h_k - 1$  transitions  $T(k,1) \dots T(k,h_k-1)$  sont des transitions silencieuses, la projection de leur vecteur de franchissement sur l'ensemble des étiquettes avec la matrice  $L$  est nulle. Seule la transition  $T(k,h_k)$  de  $\sigma_u(k)$  indiquée par la ligne 4 de l'égalité (2.6) est observable et délivre une étiquette  $e_k^{obs}$  lors de son franchissement. De même, les lignes 5 à 7 garantissent que  $T(k,1) \dots T(k,h_k-1)$  ne modifient pas le marquage mesuré, contrairement au franchissement de la dernière transition  $T(k,h_k)$  responsable de l'évolution du marquage mesuré de  $M_{k-1}^{obs}$  à  $M_k^{obs}$ .

## B. Approche incrémentale

Le problème de calcul des trajectoires compatibles en se basant sur une approche incrémentale est présenté dans cette section. L'approche non-incrémentale présentée dans [Lefebvre 2014a ; Ammour et al. 2016] permet de calculer l'ensemble de ces trajectoires compatibles grâce à la résolution d'un ensemble LMI mais présente une complexité croissante exponentiellement en fonction de ces trois paramètres :

- La taille  $K$  de la trajectoire de mesures  $TRo$
- Le nombre d'états  $N$  du graphe de marquages
- La taille maximale  $h_{max}$  d'une trajectoire silencieuse

La complexité du calcul des trajectoires compatibles  $\Gamma^{-1}(TRo)$  est donnée par  $(\Gamma^{-1}(TRo)) = \mathcal{O}((q + 1)^{N \cdot (K+1) \cdot h_{max}})$  avec  $q$  le nombre de transitions. Dans le but de réduire la complexité, un algorithme incrémental a été développé [Ammour et al. 2017b].

Cet algorithme s'appuie sur deux procédures *Sil\_events* et *Consist\_event* qui sont présentées ci-dessous :

La procédure *Sil\_events* permet de compléter de façon récursive une séquence  $(\sigma_u, M)$  avec toutes les combinaisons possibles d'événements silencieux. On obtient ainsi un ensemble de trajectoires compatibles avec  $(\sigma_u, M)$ .

### Procédure 1. *Sil\_events*

**Entrées :**  $G, \mathcal{L}, H, (\sigma_u, M), \text{SilTrajSet}$

**Sortie :**  $\text{SilTrajSet}$

1.  $\text{SilTrajSet} \leftarrow \text{SilTrajSet} \cup (\sigma_u, M)$
2. Calculer  $M'$  tel que  $M[\sigma_u > M']$
3. **pour** toute transition  $T$  validée à partir de  $M'$
4.   **si**  $\mathcal{L}(T) = \varepsilon$
5.     calculer  $M''$  tel que  $M'[T > M'']$
6.     **si**  $H.M'' = H.M'$
7.        $\text{SilTrajSet} \leftarrow \text{Sil\_events}(G, \mathcal{L}, H, (\sigma_u T, M), \text{SilTrajSet})$
8.     **fin si**
9. **fin pour**
10. **fin pour**

La procédure *Consist\_event* permet de déterminer l'ensemble des trajectoires qui complètent  $(\sigma_u, M)$  avec une transition  $T$  de label  $e_j^{obs}$  et qui produit le marquage  $(M_j^{obs})$ .

## Procédure 2. *Consist\_event*

**Entrées :**  $G, \mathcal{L}, H, (\sigma_u, M), e_j^{obs}(M_j^{obs})$

**Sortie :** *ConsistTraj*

**Initialisation :** *ConsistTraj*  $\leftarrow \emptyset$

1. Calculer  $M'$  tel que  $M[\sigma_u > M']$
2. **pour** toute transition  $T$  validée à partir de  $M'$
3.   **si**  $\mathcal{L}(T) = e_j^{obs}$
4.     Calculer  $M''$  tel que  $M'[T > M'']$
5.     **si**  $H.M'' = (M_j^{obs})$
6.       *ConsistTraj*  $\leftarrow$  *ConsistTraj*  $\cup (\sigma_u T, M)$
7.     **fin si**
8. **fin si**
9. **fin pour**

## Algorithme principal. Calcul des trajectoires compatibles

**Entrées :**  $G, \mathcal{L}, H, ReachSet(G, M_I), TRo$

**Sortie :** *ConsistTrajSet*

**Initialisation :** *ConsistInitMark*  $\leftarrow \emptyset$ , *ConsistTrajSet*  $\leftarrow \emptyset$

**1<sup>ère</sup> Etape :** Ensemble des marquages compatibles avec  $(M_0^{obs})$

1. **pour** tout  $M$  dans  $ReachSet(G, M_I)$
2.   **si**  $H.M = (M_0^{obs})$
3.     *ConsistInitMark*  $\leftarrow$  *ConsistInitMark*  $\cup \{M\}$
4.   **fin si**
5. **fin pour**

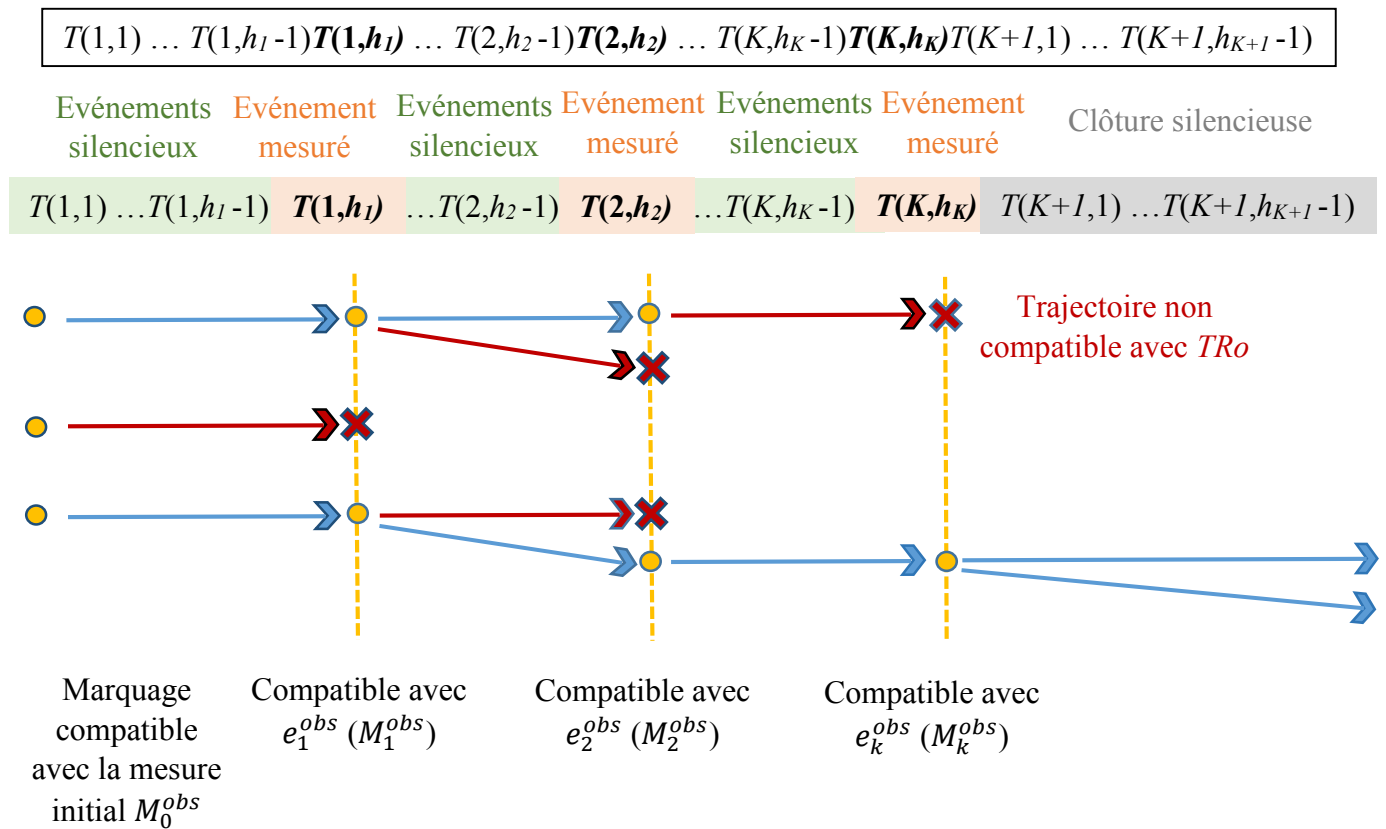
## 2<sup>ème</sup> Etape : Traitement incrémental de la trajectoire de mesures

6. **pour tout**  $M$  dans  $ConsistInitMark$
7.  $ConsistTraj \leftarrow (\varepsilon, M)$
8. Calculer la taille  $K$  de  $TRo$
9. **pour**  $j \leftarrow 1 : (K + 1)$  (pour chaque couple de mesure  $e_j^{obs}(M_j^{obs})$ )
10.  $SilTrajSet \leftarrow \emptyset$
11. **pour**  $(\sigma_u, M)$  dans  $ConsistTraj$
12.  $SilTrajSet \leftarrow Sil\_events(G, \mathcal{L}, H, (\sigma_u, M), SilTrajSet)$
13. **fin pour**
14. **si**  $j \neq (K + 1)$
15.  $ConsistTraj \leftarrow \emptyset$
16. **pour** toute trajectoire  $(\sigma_u, M)$  dans  $SilTrajSet$
17.  $ConsistTraj \leftarrow ConsistTraj \cup Consist\_event(G, \mathcal{L}, H, (\sigma_u, M), e_j^{obs}(M_j^{obs}))$
18. **fin pour**
19. **fin si**
20. **fin pour**
21.  $ConsistTrajSet \leftarrow ConsistTrajSet \cup SilTrajSet$
22. **fin pour**

On rappelle que l'intérêt essentiel d'une telle approche est de calculer l'ensemble des trajectoires compatibles  $\Gamma^{-1}(TRo)$  noté dans la suite par  $ConsistTrajSet$ , et constitue la sortie de l'algorithme incrémental. Les entrées de cet algorithme sont : le modèle du système et sa configuration de capteurs, une trajectoire de mesures ainsi que  $ReachSet(G, M_I)$ , l'ensemble des états atteignables depuis le marquage initial  $M_I$ . On rappelle que l'hypothèse H2.2 est maintenue et que l'ensemble des marquages de l'espace d'état est de cardinalité finie et connue ; tandis que le marquage à partir duquel l'enregistrement des mesures est effectué est supposé inconnu (l'état du système au démarrage de l'enregistrement ne coïncide pas forcément avec l'état initial du système). L'algorithme principal permet de générer l'ensemble  $ConsistTrajSet$ , il est composé de deux étapes de traitement (**Algorithme principal**). La première étape de cet algorithme consiste à parcourir l'ensemble des marquages de  $ReachSet(G, M_I)$  pour déterminer les marquages compatibles avec la mesure initiale du marquage  $M_0^{obs}$  regroupés dans la variable  $ConsistInitMark$ . Lors de la deuxième phase, pour chaque

marquage compatible de l'ensemble  $ConsistInitMark$  un traitement incrémental en profondeur est effectué afin de déterminer l'ensemble des trajectoires compatibles. La détermination de la transition d'étiquette  $e_1^{obs}$  compatible avec la mesure  $e_1^{obs} (M_1^{obs})$  est ajoutée à l'ensemble des trajectoires compatibles. Chaque trajectoire est complétée par des événements silencieux à l'aide de la procédure 1  $Sil\_events$ . De façon incrémentale, les autres mesures  $e_{j+1}^{obs} (M_{j+1}^{obs})$  avec  $j \in \{1, \dots, K-1\}$  sont traitées de la même manière pour toutes les trajectoires de l'ensemble  $ConsistTraj$ . Notons que chaque trajectoire peut être complétée par des clôtures silencieuses  $T(K+1,1) \dots T(K+1, h_{K+1}-1)$ . La sortie de l'algorithme principal  $ConsistTrajSet$  est un ensemble non vide qui contient au moins une trajectoire compatible avec les mesures obtenues, correspondant au comportement du système.

Pour plus de détails sur les deux procédures :  $Sil\_events$  et  $Consist\_event$  on pourra se référer à [Ammour 2017].



**Figure 2.3.** Calcul des trajectoires compatibles avec une trajectoire de mesures donnée  $TRo$

La complexité de cet algorithme incrémental est analysée, elle est donnée par l'expression  $(\Gamma^{-1}(TRo)) = \mathcal{O}(N \cdot q^{(K+1) \cdot h_{max}})$ . En comparant celle-ci à celle obtenue par l'application de la méthode LMI ( $\mathcal{O}((q + 1)^{N(K+1)h_{max}})$ ) on remarque que la complexité devient linéaire en fonction du nombre d'états  $N$ , ce qui permet de réduire la complexité du calcul en fonction de la taille du réseau. On peut atteindre une complexité de  $(N \cdot q^{2 \cdot h_{max} - 1})$  dans le cas où les trajectoires compatibles sont mises à jour à chaque mesure de la forme  $e_k^{obs} (M_k^{obs})$ , ce qui revient à considérer  $K = 1$ . La démonstration du calcul de complexité est donnée dans [Ammour 2017].

### 2.4.2. Trajectoires compatibles partiellement temporisées

Dans le bus d'améliorer le diagnostic, l'exploitation de l'information temporelle doit être considérée dans le calcul des trajectoires compatibles. En effet, lors de cette phase de calcul, et dans un contexte non temporel, la distinction entre les comportements du système qui sont à l'origine de ces trajectoires compatibles est impossible, car elles sont toutes cohérentes avec les mesures. Cependant, en tenant compte des dates de certains événements de ces trajectoires, on peut temporiser partiellement l'ensemble  $\Gamma^{-1}(TRo)$ . Les probabilités de réalisation de chaque trajectoire compatible avec une trajectoire de mesures donnée pourront être déterminées.

Dans cette section, un RdPS partiellement mesuré  $\langle G, M_I \rangle$  ainsi qu'une trajectoire de la forme  $(\sigma, M)$  sont considérés. La proposition 2.2 vient caractériser l'ensemble  $\Gamma^{-1}(TRo(\tau_0, \tau_{end}))$  des trajectoires compatibles avec la trajectoire de mesures temporisée  $TRo$  (on note, que l'on garde la même notation  $TRo$  pour une trajectoire non temporisée et pour une trajectoire temporisée). Une trajectoire de mesures temporisée  $TRo$  est obtenue grâce à l'application de la fonction de mesure  $\Gamma$  sur le couple  $(\sigma, M)$  obtenu dans l'intervalle de temps  $[\tau_0, \tau_{end}]$ . Sa forme reste la même que celle donnée par (2.2), et correspond à une succession d'étiquettes et de marquages mesurés et datés dans l'intervalle de temps  $[\tau_0, \tau_{end}]$  :

$$TRo = \Gamma((\sigma, M), \tau_0, \tau_{end}) = M^{obs}(\tau_0) e^{obs}(\tau_1) M^{obs}(\tau_1) \dots e^{obs}(\tau_{K-1}) M^{obs}(\tau_{K-1}) e^{obs}(\tau_K) M^{obs}(\tau_{end}) \quad (2.14)$$

où  $\tau_j, j = 0, \dots, K$  représente les dates des mesures. L'étiquette  $e^{obs}(\tau_j) \in E \cup \{\varepsilon\}$  correspond au  $j^{\text{ème}}$  événement mesuré à la date  $\tau_j$  et  $M^{obs}(\tau_j)$  est le  $j^{\text{ème}}$  marquage mesuré à la date  $\tau_j$ .  $M^{obs}(\tau_0) = H.M$  est le marquage mesuré à la date  $\tau_0$  tel que  $M$  est le marquage à partir duquel les mesures sont enregistrées.

**Proposition 2.2.** On considère un SED modélisé par un RdPS partiellement mesuré  $\langle G, M_I \rangle$  satisfaisant l'hypothèse H2.1 et une trajectoire de mesures temporisée  $TRo(\tau_0, \tau_{end})$  de la forme (2.14). Un couple  $(\sigma, M)$  est compatible avec  $TRo(\tau_0, \tau_{end})$  si et seulement si :

1) La séquence temporisée  $\sigma$  peut s'écrire sous la forme :

$$\begin{aligned} \sigma = & T(t_{1,1}) \dots T(t_{1,h_1-1}) \mathbf{T}(t_{1,h_1}) T(t_{2,1}) \dots T(t_{2,h_2-1}) \mathbf{T}(t_{2,h_2}) \\ & \dots T(t_{K,h_K-1}) \mathbf{T}(t_{K,h_K}) T(t_{K+1,1}) \dots T(t_{K+1,h_{K+1}-1}) \end{aligned} \quad (2.15)$$

Avec  $h_K \leq h_{max}$ ,  $k = 1, \dots, K$  et  $h_{k+1} \leq h_{max} - 1$ .  $T(t_{k,n})$  pour  $k = 1, \dots, K$  et  $n = 1, \dots, (h_K - 1)$  correspond à un événement silencieux,  $\mathbf{T}(t_{k,h_k}) = \mathbf{T}(\tau_k)$  pour  $k = 1, \dots, K$  correspondent aux transitions qui fournissent une étiquette et/ou qui engendrent un changement de marquage mesuré. La clôture silencieuse  $T(t_{K+1,1}) \dots T(t_{K+1,h_{K+1}-1})$  complète la séquence dans l'intervalle  $[\tau_K, \tau_{end}]$ .

- 2) Les dates  $t_{K,1}, t_{K,2}, \dots, t_{K,h_K}$  satisfont  $\tau_{k-1} \leq t_{K,1} \leq t_{K,2} \leq \dots \leq t_{K,h_K}$  pour tout  $k = 1, \dots, h_{k+1}$  et  $t_{K,h_K} = \tau_K$  pour tout  $k = 1, \dots, K$ .
- 3) La trajectoire  $(\sigma, M)$  satisfait l'ensemble des LMI ((2.11), (2.12) et (2.13)) introduit dans la sous-section 2.4.1 en faisant abstraction du temps.

**Démonstration.** La condition (3) de la proposition 2.2 est utilisée pour déterminer l'ensemble  $\Gamma^{-1}(TRo(\tau_0, \tau_{end}))$  des séquences de tir non temporisées  $(\sigma, M)$ . Dans ce but,  $TRo(\tau_0, \tau_{end})$  est remplacée par  $TRo(K)$  en faisant abstraction du temps. De plus, si la condition (2) est vérifiée, alors chaque sous-séquence  $T(t_{K,1}) \dots T(t_{K,h_K})$  est une séquence temporisée compatible avec  $M^{obs}(\tau_{K-1}) e^{obs}(\tau_K) M^{obs}(\tau_K)$ . De ce fait, si la trajectoire  $(\sigma, M)$  est compatible avec  $TRo(\tau_0, \tau_{end})$  et que  $\sigma = T(t_{1,1}) \dots T(t_{1,h_1}) \dots T(t_{K,h_K}) T(t_{K+1,1}) \dots T(t_{K+1,h_{K+1}-1})$  alors :  $\Gamma((\sigma, M), \tau_0, \tau_{end}) = M^{obs}(\tau_0) e^{obs}(\tau_1) M^{obs}(\tau_1) \dots e^{obs}(\tau_{K-1}) M^{obs}(\tau_{K-1}) e^{obs}(\tau_K) M^{obs}(\tau_{end})$ .

D'autre part, et en se basant sur l'algorithme incrémental présenté dans la sous-section 2.4.1, le calcul de l'ensemble  $\Gamma^{-1}(TRo(\tau_0, \tau_{end}))$  se fait en deux étapes. La première étape consiste à déterminer l'ensemble non temporisé des trajectoires compatibles avec  $TRo(K)$  (le temps n'est pas considéré dans cette étape de calcul).



La deuxième étape appelle la procédure *Consist\_event* (procédure 2 de la sous-section 2.4.1) afin d'identifier les transitions ayant engendrées une mesure et/ou un changement de marquage afin de leur associer les dates  $\tau_K$ . L'ensemble des trajectoires  $(\sigma, M) \in \Gamma^{-1}(TRo(\tau_0, \tau_{end}))$  avec  $\sigma$  de la forme (2.15) est ainsi obtenu.

## 2.5. Probabilités des trajectoires

Dans le but de calculer la probabilité de défauts structurels et temporels, il est important d'évaluer la probabilité de chaque trajectoire  $(\sigma, M)$  partiellement mesurée, compatible avec *TRo*. De ce fait, l'ensemble  $\Gamma^{-1}(TRo)$  est tout d'abord calculé par une méthode basée soit sur les LMI soit sur l'approche incrémentale qui est détaillée dans la sous-section (2.4.1). Ensuite, un ensemble étendu  $\Gamma_E^{-1}(TRo)$  des trajectoires compatibles avec *TRo* est obtenu en considérant la concurrence entre les jetons. Il en résulte une duplication des trajectoires appartenant à  $\Gamma^{-1}(TRo)$ . Par conséquent, toute transition appartenant à une séquence temporisée est complétée par sa date de validation. La forme de la séquence temporisée (2.15) devient :

$$\begin{aligned} \sigma = & T(\tau_0, t_{1,1}) \dots T(t'_{1,h_1-1}, t_{1,h_1-1}) T(t'_{1,h_1}, \tau_1) \dots T(t'_{k,h_k}, \tau_k) \\ & \dots T(t'_{K,1}, t_{K,1}) \dots T(t'_{K,h_K-1}, t_{K,h_K-1}) T(t'_{K,h_K}, \tau_K) \\ & T(t'_{K+1,1}, t_{K+1,1}) \dots T(t'_{K+1,h_{K+1}-1}, t_{K+1,h_{K+1}-1}) \end{aligned} \quad (2.16)$$

où  $t'_{k,h}$  avec  $h = 1, \dots, h_k$  et  $k = 1, \dots, K+1$  correspond à la date de validation de la transition  $T(t'_{k,h}, t_{k,h})$  et  $t_{k,h}$  correspond à sa date de franchissement. On note que la date de validation  $t'$  de chaque transition dans la séquence  $\sigma$  correspond à une date de tir mesurée ou non mesurée d'une autre transition (ou éventuellement la date initiale).

**Remarque 2.1.** L'ensemble étendu des trajectoires compatibles  $\Gamma_E^{-1}(TRo)$  est de cardinal plus grand que celui de  $\Gamma^{-1}(TRo)$ . Ces deux ensembles deviennent équivalents dans le cas où le réseau est 1-borné.

Le calcul des trajectoires étendues est obtenu en utilisant les algorithmes 2.1 et 2.2. L'algorithme 2.1 calcule sous forme d'un arbre nommé  $GE(M, TRo)$  et à partir d'un marquage  $M$ , l'ensemble des trajectoires non temporisées compatibles avec *TRo*. Chaque nœud  $S \in GE(M, TRo)$  est étiqueté par :

- un marquage  $S.m$ .
- une séquence  $S.\sigma$  telle que  $M[S.\sigma > S.m$ .

- un drapeau  $S.tro$  qui indique si  $S.\sigma$  est compatible avec  $TRo$  ou non :  $S.tro = 1$  si  $\Gamma((S.\sigma, S.m), \tau_0, \tau_{end}) = TRo$  et  $S.tro = 0$  sinon.

La construction de l'arbre  $GE$  est obtenue en dupliquant toute transition validée en fonction de son degré de validation. Une transition  $T_j$  de degré de validation  $n_j(M)$  au marquage  $M$ , permet d'obtenir  $n_j(M)$  branches à partir du marquage  $M$ . La construction dépend également de la trajectoire de mesures  $TRo$ . L'exploration d'une branche s'arrête dès qu'elle coïncide avec  $TRo$ . Une branche qui ne coïncide pas avec  $TRo$  est supprimée de l'arbre. On note que l'hypothèse 2.1 est retenue et implique que l'arbre  $GE(M, TRo)$  contient un nombre fini de nœuds.

### Algorithme 2.1. Calcul de l'arbre $GE(M, TRo)$

**Entrées :**  $M, TRo$

**Sortie :**  $GE$

**Initialisation :** Une structure  $S_0$  telle que  $S_0.m \leftarrow M_I, S_0.\sigma \leftarrow \varepsilon, S_0.tro \leftarrow 0, GE \leftarrow \{S_0\}$

1. **pour** chaque nœud  $S \in GE$  tel que  $S_0.tro = 0$
2.     **Pour** chaque transition  $T$  validée par le marquage  $S.m$  avec un degré  $n(S.m)$
3.         **Pour**  $j \leftarrow 1$  jusqu'à  $n(S.m)$
4.             Calculer  $S'$  tel que :
5.              $S'.m \leftarrow M'$  tel que  $M[T > M'$
6.              $S'.\sigma \leftarrow [S.\sigma \sigma']$  (concaténation de  $S.\sigma$  et  $\sigma'$ )
7.              $S'.tro \leftarrow \Gamma([S.\sigma \sigma'], M), \tau_0, \tau_{end})$
8.             **si**  $S'.tro = 1$ ,
9.                 enlever la branche qui se termine par le nœud  $S'.m$
10.            **sinon**
11.                  $GE \leftarrow GE \cup \{S'\}$
12.            **fin si**
13.         **fin pour**
14.     **fin pour**
15. **fin pour**

L'algorithme 2.2, quant à lui, transforme une séquence non temporisée de la forme :  $\sigma = T(1)T(2) \dots T(H)$ , franchie au marquage  $M$  à l'instant 0, en une séquence temporisée  $\sigma'$  de la forme (2.16). Les dates de tir et de validation de chaque transition sont intégrées dans  $\sigma'$ . L'algorithme 2.2 utilise l'ordre chronologique de tir de la séquence non temporisée  $\sigma$  ainsi que la politique de tir au plus tôt pour mettre à jour, à chaque marquage  $M(k)$  de  $(\sigma, M)$ , les transitions validées, leurs dates de validation et leurs dates de franchissement au plus tôt. Ensuite, les dates de validation et de franchissement sont enregistrées dans un calendrier noté  $CAL(M(k)) = \{(T_{j1}, t'_{j1}, t_{j1}), (T_{j2}, t'_{j2}, t_{j2}), \dots\}$ , associé à chaque marquage intermédiaire  $M(k) \in (\sigma, M)$ , avec  $k = 0, \dots, h$ . Ce calendrier contient toutes les transitions validées par le marquage  $M(k)$ .

On note qu'une même transition peut apparaître plusieurs fois dans  $CAL(M(k))$  avec des délais identiques ou différents dans le cas où elle est validée plusieurs fois par le marquage  $M(k)$ . Le calendrier est initialisé en supposant que la trajectoire commence à l'instant 0 et qu'aucune transition n'est validée avant 0 (ligne 1 de l'algorithme 2.2). Le calendrier est mis à jour pour chaque marquage de la trajectoire  $(\sigma, M)$ .

**Algorithme 2.2.** Calcul des dates de validation et franchissement des transitions validées

**Entrées :**  $\sigma, M, D_{min}$

**Sortie :**  $\sigma'$

**Initialisation :**  $\tau \leftarrow 0, CAL \leftarrow \{(T_j, 0, d_{min_j}) \text{ tel que } M[T_j > \}, \sigma' \leftarrow (\varepsilon, 0), h \leftarrow |\sigma|$

1. **pour**  $k \leftarrow 1: h$
2. trouver dans  $CAL$ , la date de tir au plus tôt  $t_k$  de la transition  $T(k)$  dans  $\sigma$
3.  $\tau \leftarrow t_k$ , supprimer de  $CAL$  l'entrée  $(T(k), t'_k, t_k)$
4.  $CAL_{new} \leftarrow \emptyset, M' \leftarrow M - W_{PR}. X(T(k))$
5. **Pour** toutes les transitions  $T'$  tel que  $M'[T' >$
6. calculer le degré de validation  $n'(T', M')$  de  $T'$  à partir de  $M'$
7. **pour**  $j \leftarrow 1$  jusqu'à  $n(T', M')$
8. trouver la  $j^{\text{ème}}$  occurrence  $(T', t'_j, t_j)$  de  $T'$  dans  $CAL$
9.  $CAL_{new} \leftarrow CAL_{new} \cup (T', t'_j, \max(t_j, \tau))$
10. **fin pour**
11. **fin pour**
12.  $M'' \leftarrow M' + W_{PO}. X(T(k))$

13. **Pour** toutes les transitions  $T''$  telle que  $M'' [T'' >$
14.       calculer le degré de validation  $n''(T'', M'')$  de  $T''$  à partir de  $M''$
15.       **pour**  $j \leftarrow 1$  jusqu'à  $n''(T'', M'') - n'(T'', M')$
16.              $CAL_{new} \leftarrow CAL_{new} \cup (T'', \tau, \tau + d_{min}(T''))$
17.       **fin pour**
18. **fin pour**
19.      $CAL \leftarrow CAL_{new}, \sigma' \leftarrow \sigma'(T(k), t'_k, t_k), M \leftarrow M''$
20. **fin pour**

**Exemple 2.3.** On reprend l'exemple de la figure 2.1 avec la trajectoire mesurée suivante :  $TRo = (0, 0) (e, 8) (0, 8)$ , collectée pendant l'intervalle de temps  $[0 : 8]$ . On considère que le marquage initial est connu  $M(0) = M_I$ , et qu'aucune clôture silencieuse n'est prise en compte. Le processus de surveillance s'arrête une fois l'étiquette  $e$  mesurée (c'est-à-dire  $\tau_{end} = 8$ ). L'ensemble  $\Gamma^{-1}(TRo)$  contient deux trajectoires compatibles  $\Gamma^{-1}(TRo) = \{(\sigma_a, M_I), (\sigma_b, M_I)\}$  avec  $\sigma_a = T_1(t_{1,1})T_2(8)$  et  $\sigma_b = T_1(t_{1,1})T_1(t_{1,2})T_2(8)$ . L'ensemble étendu  $\Gamma_E^{-1}(TRo)$  contient six trajectoires compatibles avec  $TRo$  :  $\Gamma_E^{-1}(TRo) = \{(\sigma_i, M_I), i = 1, \dots, 6\}$  que l'on peut énumérer par l'exploration de l'arbre, représenté par la figure 2.4 (Tableau 2.1).

La distinction entre les deux jetons de la place  $P_1$  permet d'obtenir les six trajectoires compatibles de l'ensemble  $\Gamma_E^{-1}(TRo)$ . Chaque nœud de l'arbre représente le franchissement d'une transition, le jeton qui en est responsable (jeton 1 ou jeton 2) est aussi indiqué. Le parcours des branches de l'arbre jusqu'à leur extrémité mène aux transitions qui ne sont pas franchies avant  $\tau_{end}$ . On note que la symétrie de l'arbre obtenu pour cet exemple est dû au fait que la transition  $T_1$  est deux fois validée à l'instant initial.

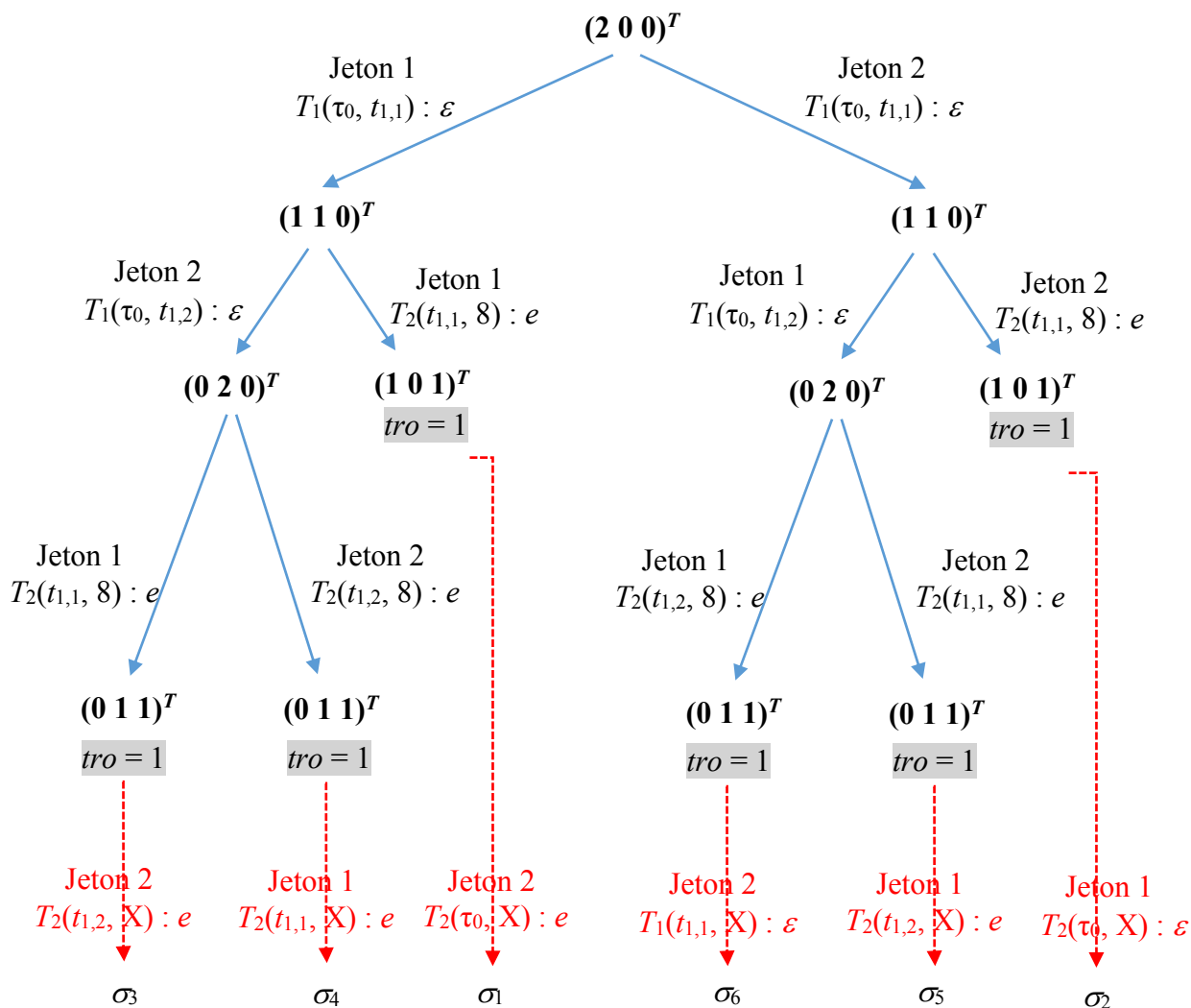


Figure 2.4. Arbre de détermination de  $\Gamma_E^{-1}(TRo)$

Les six trajectoires obtenues sont représentées dans le tableau 2.1. Une transition franchie par le jeton 2 est soulignée, celle non soulignée est franchie par le jeton 1.

**Tableau 2.1.** Les trajectoires de l'ensemble étendu  $\Gamma_E^{-1}(TRo)$ 

$\sigma$		Description
$\sigma_a$	$\sigma_1$	$T_1(0, t_{1,1}) T_2(t_{1,1}, 8)$
	$\sigma_2$	$\underline{T_1(0, t_{1,1})} \underline{T_2(t_{1,1}, 8)}$
$\sigma_b$	$\sigma_3$	$T_1(0, t_{1,1}) \underline{T_1(0, t_{1,2})} T_2(t_{1,1}, 8)$
	$\sigma_4$	$T_1(0, t_{1,1}) \underline{T_1(0, t_{1,2})} \underline{T_2(t_{1,2}, 8)}$
	$\sigma_5$	$\underline{T_1(0, t_{1,1})} T_1(0, t_{1,2}) \underline{T_2(t_{1,1}, 8)}$
	$\sigma_6$	$\underline{T_1(0, t_{1,1})} T_1(0, t_{1,2}) T_2(t_{1,2}, 8)$

Le paragraphe suivant propose une méthode numérique qui permet d'évaluer la probabilité de chaque trajectoire de  $\Gamma_E^{-1}(TRo)$ .

La probabilité  $\text{Prob}((\sigma, M) \mid TRo)$  pour qu'une trajectoire compatible partiellement mesurée  $(\sigma, M) \in \Gamma_E^{-1}(TRo)$  avec  $\sigma$  de la forme (2.16), soit réellement celle qui s'est produite si  $TRo$  est collectée pendant l'intervalle de temps  $[\tau_0, \tau_{end}]$ , est évaluée. En plus des informations temporelles contenues dans cette trajectoire, les trois facteurs suivants sont considérés :

- a)  $\pi_0(M)$  : la probabilité d'être dans le marquage  $M$  à la date  $\tau_0$ . Pour simplifier le problème, dans ce qui suit, on considère que cette probabilité est connue selon l'hypothèse H2.3 :

**Hypothèse H2.3.** On considère que l'ensemble  $M(\tau_0) \subset \text{ReachSet}(G, M_I)$  des marquages possibles  $M$  à la date  $\tau_0$  ainsi que la probabilité  $\pi_0(M)$  avec  $M \in M(\tau_0)$  sont supposés être connus.

- b)  $\pi_1(\sigma/TRo \text{ et } M)$  : la probabilité d'une séquence  $\sigma$  issue du marquage  $M$  et compatible avec  $TRo$ . Cette probabilité satisfait les deux conditions suivantes :

1. les dates de franchissement des transitions respectent l'ordre chronologique de la trajectoire

$$\sigma: \tau_0 \leq t_{1,1} \leq \dots \leq t_{1,h_1-1} \leq \tau_1 \leq \dots \leq \tau_K \leq t_{K+1,1} \leq \dots \leq t_{K+1,h_{K+1}-1} \leq \tau_{end}$$

2. les  $K$  transitions  $T(t_{k,h_k})$  sont franchies dans un petit intervalle de temps après  $\tau_k$  :

$$t_{k,h_k} \in [\tau_k : \tau_k + dt] \text{ pour } k = 1, \dots, K \quad (2.17)$$

où  $dt$  correspond à une petite durée.

De ce fait, cette probabilité implique que le groupe de transitions silencieuses  $T(t'_{k,1}, t_{k,1}) \dots T(t'_{k,h_k-1}, t_{k,h_k-1})$  pour  $k = 1, \dots, K$  est franchi après la date  $\tau_{k-1}$  et avant la date  $\tau_k$  (c'est-à-dire que :  $\tau_{k-1} < t_{k,h_k-1} < \tau_k$ ) et que la clôture silencieuse  $T(t'_{k+1,1}, t_{k+1,1}) \dots T(t'_{k+1,h_{k+1}-1}, t_{k+1,h_{k+1}-1})$  est franchie dans l'intervalle de temps  $]\tau_k : \tau_{end}]$ .

c)  $\pi_2(\sigma/TR_0 \text{ et } M)$  : la probabilité qu'aucune autre transition validée n'est franchie avant la date  $\tau_{end}$

La probabilité  $\text{Prob}((\sigma, M) \in \Gamma_E^{-1}(TR_0) | TR_0)$  qu'une trajectoire compatible partiellement temporisée soit celle dont les mesures sont issues dans l'intervalle  $[\tau_k : \tau_{end}]$  est donnée par :

$$\text{Prob}((\sigma, M) \in \Gamma_E^{-1}(TR_0) | TR_0) = \frac{\pi((\sigma, M) | TR_0)}{\sum_{((\sigma', M') \in \Gamma_E^{-1}(TR_0))} \pi((\sigma', M') | TR_0)} \quad (2.18)$$

où  $\pi((\sigma, M) | TR_0)$  est définie par (2.19) et dépend des trois facteurs présentés précédemment :

$$\pi((\sigma, M) | TR_0) = \pi_0(M) \times \pi_1(\sigma | TR_0 \text{ et } M) \times \pi_2(\sigma | TR_0 \text{ et } M) \quad (2.19)$$

Le calcul numérique des deux probabilités  $\pi_1(\sigma | TR_0 \text{ et } M)$  et  $\pi_2(\sigma | TR_0 \text{ et } M)$  est détaillé par les deux algorithmes 2.3 et 2.4.

Pour chaque transition  $T(t'_{k,h}, t_{k,h})$  de la trajectoire  $\sigma$ , il existe  $k'$  et  $h'$  tels que  $t'_{k,h} = t_{k',h'}$  c'est-à-dire que la date de validation  $t'_{k,h}$  est nécessairement la date de franchissement d'une autre transition (silencieuse ou mesurée) de  $\sigma$  ou éventuellement l'instant initial  $\tau_0$  (on note  $t_{0,0} = \tau_0$ ). Le marquage  $M(t'_{k,h})$  aussi noté  $M(t_{k',h'})$  est responsable de la validation de la transition  $T(t'_{k,h}, t_{k,h})$  et la variable aléatoire (VA)  $u_{k,h}$  :

$$u_{k,h} = t_{k,h} - t'_{k,h} \quad (2.20)$$

représente la durée de franchissement de la transition  $T(t'_{k,h}, t_{k,h})$ .

Intéressons-nous à présent au calcul de  $\pi((\sigma, M) | TR_O)$  défini par (2.19). Dans un contexte stochastique, différentes politiques de choix peuvent être mises en œuvre en cas de conflit. On définit deux ensembles de transitions :

- i.  $C'(k, h)$  l'ensemble des transitions  $T$  en conflit avec  $T(t'_{k,h}, t_{k,h})$  au marquage  $M(t'_{k,h})$ , à l'exclusion de  $T(t'_{k,h}, t_{k,h})$
- ii.  $C(k, h)$  l'ensemble des transitions  $T$  en conflit avec  $T(t'_{k,h}, t_{k,h})$ , y compris  $T(t'_{k,h}, t_{k,h})$ , au marquage  $M(t'_{k,h})$ :  $C(k, h) = C'(k, h) \cup T(t'_{k,h}, t_{k,h})$

Deux cas se présentent en fonction de la politique de choix :

Pour une politique de choix par compétition, on définit la densité de probabilité  $f'_{k,h}(u_{k,h})$  par l'équation (2.21). Cette équation permet d'évaluer la probabilité de franchir une transition spécifique  $T(t'_{k,h}, t_{k,h})$  avant toute autre transition en conflit avec elle.

$$f'_{k,h}(u_{k,h}) = f_{k,h}(u_{k,h}) \left[ \prod_{T_j \in C'(k,h)} (1 - \int_0^{u_{k,h}} f_j(u) du) \right] \quad (2.21)$$

où :  $f_{k,h}$  correspond à la densité de probabilité de  $T(t'_{k,h}, t_{k,h})$ ,  $\left[ \prod_{T_j \in C'(k,h)} (1 - \int_0^{u_{k,h}} f_j(u) du) \right]$  est un terme correctif résultant du fait que les transitions qui sont en conflit avec  $T(t'_{k,h}, t_{k,h})$  ne sont pas franchies dans l'intervalle  $[t'_{k,h}, t_{k,h}]$ .

De manière similaire, on obtient l'équation (2.22) dans le cas où une politique de choix par présélection est utilisée.

$$f'_{k,h}(u_{k,h}) = f_{k,h}(u_{k,h}) \omega'(k, h) \quad (2.22)$$

où  $\omega'(k, h)$  représente le poids normalisé de  $T(t'_{k,h}, t_{k,h})$  :

$$\omega'(k, h) = \frac{\omega(k, h)}{\sum_{T_j \in C(k,h)} \omega(T_j)} \quad (2.23)$$



et  $\omega(k, h)$  représente le poids de  $T(t'_{k,h}, t_{k,h})$ .

Les variables  $u_{k,h}$  ne sont pas toutes indépendantes les unes des autres. Les trois types de transitions suivants sont considérés :

- 1<sup>er</sup> type de transitions, lorsque la date de tir  $t_{k,h}$  et la date de validation  $t'_{k,h}$  sont toutes deux mesurées et valent respectivement  $\tau_k$  et  $\tau_k'$ . Par conséquent,  $u_{k,h}$  est une variable déterministe. Ces transitions ne sont pas prises en compte dans l'évaluation de  $\pi_I(\sigma | TR_O \text{ et } M)$ .
- 2<sup>ème</sup> type de transitions lorsque seule la date de tir  $t_{k,h}$  est mesurée et égale à  $\tau_k$  et  $t_{k',h'}$  est non mesurée. Pour ce type de transitions,  $u_{k,h}$  est une VA qui dépend de  $u_{k',h'}$  :  $u_{k,h} + u_{k',h'} = \tau_k - t_{k',h'}$  où  $t_{k',h'}$  est la date de validation de la transition  $T(t_{k',h'}, t_{k',h'})$ ,  $u_{k,h} = \tau_k - t_{k',h'}$  et  $u_{k',h'} = t_{k',h'} - t_{k',h'}$ . Les transitions de type 2 sont séparées en plusieurs classes de transitions qui sont validées en même temps. (2.24) définit les classes suivantes  $Cs(\mathbf{k}', \mathbf{h}')$ ,  $h' = 1, \dots, h_k$ ,  $k' = 1, \dots, K+1$  :

$$Cs(\mathbf{k}', \mathbf{h}') = \{(k, h) \text{ tel que } T(t_{k',h'}, t_{k,h}) \text{ est du type 2 et } T(t_{k',h'}, t_{k,h}) \text{ est validée au marquage } M(t_{k',h'})\} \quad (2.24)$$

- 3<sup>ème</sup> type de transitions lorsque la date  $t_{k,h}$  est non mesurée quel que soit le statut de mesure de  $t_{k',h'}$  (mesurée ou non mesurée). Pour ce type de transitions,  $u_{k,h}$  est une VA indépendante.

Les transitions du 3<sup>ème</sup> type ainsi que celles appartenant à  $Cs(\mathbf{k}', \mathbf{h}')$  (c'est-à-dire les transitions validées à partir du franchissement de  $T(t_{k',h'}, t_{k,h})$ ) sont employées dans le calcul de  $\pi_I(\sigma | TR_O \text{ et } M)$  grâce aux deux équations (2.25) et (2.26) :

$$g_{k,h}(u_{k,h}) = f'_{k,h}(u_{k,h}) \left( \prod_{(i,j) \in Cs(\mathbf{k}, \mathbf{h})} f'_{i,j}(u_{i,j}) \right) \quad (2.25)$$

On note que, si  $Cs(\mathbf{k}, \mathbf{h}) = \emptyset$ , alors  $g_{k,h}(u_{k,h}) = f'_{k,h}(u_{k,h})$ .

Finalement,  $\pi_I(\sigma | TR_O \text{ et } M)$  est défini par (2.26) :

$$\begin{aligned} \pi_I(\sigma | TR_O \text{ et } M) &= \int_{\tau_0}^{\tau_1} g_{1,1}(u_{1,1}) \left( \int_{t_{1,1}}^{\tau_1} g_{1,2}(u_{1,2}) \dots \right. \\ &\quad \left. \dots \left( \int_{t_{1,h_1-1}}^{\tau_1} g_{1,h_1-1}(u_{1,h_1-1}) \left( \int_{\tau_1}^{\tau_2} g_{2,1}(u_{2,1}) \dots \right. \right. \right. \end{aligned}$$

$$\dots \left( \int_{t_{K+1, h_{K+1}-1}}^{\tau_{end}} g_{K+1, h_{K+1}-1}(u_{K+1, h_{K+1}-1}) \cdot dt_{K+1, h_{K+1}-1} \right) \dots$$

$$\dots dt_{2,1}) dt_{1, h_1-1}) \dots dt_{1,2}) dt_{1,1} \quad (2.26)$$

où la limite inférieure de chaque somme coïncide avec la date de tir de celle qui la précède et la durée  $u_{k,h}$  dépend de  $t_{k,h}$  selon l'équation (2.26).

L'évaluation de  $\pi_1(\sigma | TRo \text{ et } M)$  est obtenue grâce à une méthode d'intégration numérique de l'équation (2.26) présentée dans l'algorithme 2.3. Cet algorithme récursif utilise comme entrée : la trajectoire compatible partiellement mesurée  $(\sigma, M) \in \Gamma_E^{-1}(TRo)$ , l'intervalle de temps  $[\tau_0, \tau_{end}]$  nécessaire pour le calcul de la somme et le paramètre  $dt$ . L'algorithme commence par la première transition du 3<sup>ème</sup> type dans la séquence  $\sigma$  en lui attribuant  $\tau_0$  comme date de validation. Ensuite, des sous-routines (qui ne sont pas détaillées dans l'algorithme) calculent  $Cs(T)$  et  $C(T)$ , trouvent la prochaine transition du 3<sup>ème</sup> type dans une séquence, évaluent la densité de probabilité  $f_T$  d'une transition  $T$  donnée et calculent le terme correctif dans (2.21) et (2.23) en tenant compte de la politique de choix utilisée. On note que la précision de la probabilité  $\pi_1$  dépend du paramètre  $dt$  utilisé dans l'approximation numérique des sommes successives de l'équation (2.26) .

### Algorithme 2.3. Calcul de $\pi_1(\sigma | TRo \text{ et } M)$

**Entrées :**  $(\sigma, M), T, t'(T), [\tau_0 : \tau_{end}], dt$

**Sortie :**  $\pi_1$

**Initialisation :**  $\pi_1 \leftarrow 0$

1. calculer  $Cs(T)$  and  $C(T)$
2. **pour**  $t \leftarrow \max(\tau_0, t'(T) + a(T)) : \min(\tau_{end}, t'(T) + b(T))$  avec un pas de  $dt$
3.  $T_{next} \leftarrow$  la prochaine transition du 3<sup>ème</sup> type dans  $(\sigma, M)$
4. **si**  $T_{next} \neq \mathcal{E}$ ,
5.  $t_{next} \leftarrow$  la date de franchissement de la dernière transition qui tir avant  $T_{next}$
6.  $r_1 \leftarrow$  **Algorithme 2.3**  $((\sigma, M), T_{next}, t'(T_{next}), [t_{next} : \tau_{end}])$
7. **sinon**  $r_1 \leftarrow 1$ ,

8. **fin si**
9.  $u \leftarrow t - t'(T), r_2 \leftarrow f_T(u)$
10. **si**  $C(T) \neq \emptyset$ , corriger  $r_2$  par (2.21) ou (2.23) selon la politique de choix
11. **fin si**
12. **pour** chaque  $T' \in Cs(T)$
13.  $u' \leftarrow \tau(T') - t, r_2 \leftarrow r_2 \times f_{T'}(u')$
14. **fin pour**
15.  $\pi_1 \leftarrow \pi_1 + r_1 \times r_2 \times dt$
16. **fin pour**

On rappelle que  $\text{Prob}((\sigma, M) \in \Gamma_E^{-1}(TR_O) | TR_O)$  dépend en plus de  $\pi_0(M)$  et d'un 3<sup>ème</sup> terme  $\pi_2(\sigma | TR_O \text{ et } M)$  détaillé dans l'équation (2.19).

Pour calculer  $\pi_2(\sigma | TR_O \text{ et } M)$ , on considère  $E(M(\tau_{end}))$  l'ensemble des transitions  $T$  qui sont validées au dernier marquage  $M(\tau_{end})$  atteint par la trajectoire  $(\sigma, M)$  mais qui ne sont pas franchies avant  $\tau_{end}$ .

- pour une politique de choix par compétition,  $\pi_2(\sigma | TR_O \text{ et } M)$  est calculée comme suit :

$$\pi_2(\sigma | TR_O \text{ et } M) = \prod_{T \in E(M(\tau_{end}))} \left(1 - \int_0^{\tau_{end} - t'(T)} f_T(u) du\right) \quad (2.27)$$

où  $t'(T)$  et  $f_T(u)$  sont respectivement, la date de validation et la densité de probabilité de la durée de franchissement de la transition  $T$ .

On note que le conflit effectif entre les transitions de  $E(M(\tau_{end}))$  n'est pas résolu (car la résolution dépend des dates de tir prévues mais les transitions ne sont pas franchies).

- pour une politique de choix par présélection, le conflit entre les transitions de  $E(M(\tau_{end}))$  est a priori résolu en fonction des poids. Par conséquent, ces poids doivent être considérés dans l'évaluation de  $\pi_2(\sigma | TR_O \text{ et } M)$ . De plus, il est nécessaire de faire une partition de  $E(M(\tau_{end}))$  en plusieurs sous-ensembles distincts définis en fonction des transitions  $T$  qui sont en conflit effectif au marquage  $M(\tau_{end})$ . L'ensemble  $E(M(\tau_{end}))$  est ainsi défini par (2.28) :

$$E(M(\tau_{end})) = \cup_{T \in E(M(\tau_{end}))} \mathbf{C}(T) \quad (2.28)$$

où  $\mathbf{C}(T)$  représente l'ensemble des transitions, y compris  $T$ , en conflit effectif avec  $T$  au marquage  $M(\tau_{end})$ . Compte tenu des poids normalisés  $\omega'(T)$  calculés par (2.22) pour chaque ensemble  $\mathbf{C}(T)$ , le calcul de  $\pi_2(\sigma | TR_0 \text{ et } M)$  est donné par (2.29) :

$$\pi_2(\sigma | TR_0 \text{ et } M) = \prod_{T \in E(M(\tau_{end}))} \left( 1 - \int_0^{\tau_{end} - t'(T)} \left( \sum_{T' \in \mathbf{C}(T)} \omega'(T') f_{T'}(u) \right) du \right) \quad (2.29)$$

où  $t'(T)$  et  $f_{T'}(u)$  sont respectivement, la date de validation et la densité de probabilité de la durée de franchissement de la transition  $T'$  dans  $\mathbf{C}(T)$ . On note que, dans (2.29) toutes les transitions appartenant à  $\mathbf{C}(T)$  sont supposées être validées au même instant.

Par conséquent, la probabilité d'une trajectoire  $(\sigma, M) \in \Gamma^{-1}(TR_0)$  est obtenue en effectuant la somme de toutes les probabilités de l'ensemble étendu des trajectoires  $\Gamma_E^{-1}(TR_0)$  qui coïncident avec  $(\sigma, M)$  :

$$Prob((\sigma, M) \in \Gamma^{-1}(TR_0) | TR_0) = \sum_{\Phi(\sigma') = \sigma} Prob((\sigma', M) | TR_0) \quad (2.30)$$

où  $\Phi(\sigma')$  est une fonction qui transforme toute trajectoire étendue temporisée de la forme (2.16) en une trajectoire temporisée de la forme (2.15) en faisant abstraction des dates de validations des transitions.

**Remarque 2.2.** Dans ce travail, on considère que l'instant initial  $\tau_0$  coïncide avec la date de validation de la 1<sup>ère</sup> transition dans la séquence  $\sigma$ .

## 2.6. Probabilités des défauts

Nous avons déterminé les comportements compatibles avec les mesures et avons proposé un calcul de la probabilité de ces comportements à partir des mesures collectées dans une fenêtre de temps. Il est désormais possible, compte tenu de ces informations, d'évaluer la probabilité de chaque type de défaut.

### 2.6.1. Probabilités des défauts structurels

La probabilité d'un défaut structurel est détaillée dans la proposition 2.3. Cette probabilité est calculée en s'appuyant sur les mesures obtenues lors du fonctionnement du système ainsi que sur l'information temporelle incluse dans celles-ci.

**Proposition 2.3.** On considère un SED modélisé par un RdPS partiellement observable  $\langle G, L, H, M_I \rangle$  qui satisfait les deux hypothèses H2.1 et H2.2, et une trajectoire de mesures  $TR_O$  collectée durant l'intervalle de temps  $[\tau_0, \tau_{end}]$ . La probabilité que le défaut structurel  $\mathbf{R}$  se soit produit lorsque la trajectoire  $TR_O$  est observée, est donnée par :

$$\text{Prob}(F_R|TR_O) = \sum_{(\sigma, M) \in \Phi_R(TR_O)} \text{Prob}((\sigma, M) | TR_O) \quad (2.31)$$

avec  $\Phi_{R(TR_O)} = \{(\sigma, M) \in \Gamma_E^{-1}(TR_O) \text{ telle que } F_R(\sigma) = 1\}$ .

**Démonstration.** La proposition utilise la probabilité des trajectoires compatibles donnée par (2.18). Seules les trajectoires qui incluent le défaut structurel  $\mathbf{R}$  et qui appartiennent à l'ensemble  $\Phi_{R(TR_O)}$  sont considérées. La somme de leurs probabilités représente la probabilité d'occurrence du défaut dans l'intervalle  $[\tau_0, \tau_{end}]$  suivant les mesures  $TR_O$ .

**Exemple 2.4.** Reprenons l'exemple 2.1 avec les mêmes paramètres, ainsi qu'une trajectoire de mesures  $TR_O = (0, 0) (e, 8) (0, 8)$  collectée dans l'intervalle de temps  $[0 : 8]$  avec  $M(0) = M_I = (2 \ 0 \ 0)^T$ . La probabilité des trajectoires compatibles calculée par (2.18) est donnée dans le tableau 2.2. Si le défaut structurel  $\mathbf{R} = \{(T_1 T_1, \ll)\}$  est considéré alors  $\text{Prob}(\mathbf{R}|TR_O) = 0.56$  car seulement les trajectoires  $\sigma_b$  sont concordantes avec  $\mathbf{R}$ . Un ensemble de simulations de Monte-Carlo est utilisé pour confirmer ce résultat : 100 trajectoires qui coïncident avec  $TR_O$  (avec une erreur maximale tolérée de 0,02 UT) sont simulées. La probabilité des trajectoires fautives est de 0.56.

**Tableau 2.2.** Probabilité des trajectoires compatibles

$\sigma$		Probabilité	$\sum$ Probabilités
$\sigma_a$	$\sigma_1$	0.22	0.44
	$\sigma_2$	0.22	
$\sigma_b$	$\sigma_3$	0.23	0.56
	$\sigma_4$	0.05	
	$\sigma_5$	0.23	
	$\sigma_6$	0.05	

### 2.6.2. Probabilités des défauts temporels

Une méthode numérique similaire à celle détaillée dans la section 2.5 est utilisée pour le calcul de la probabilité d'un défaut défini par des contraintes temporelles. L'évaluation de la probabilité qu'une trajectoire de mesures satisfait ou non ces contraintes temporelles est basée sur :

- la probabilité  $\text{Prob}((\sigma, M)|TR_0)$  de chaque trajectoire temporisée partiellement mesurée  $(\sigma, M) \in \Gamma_E^{-1}(TR_0)$  calculée par (2.18)
- la probabilité que cette trajectoire satisfait les contraintes temporelles  $\text{Prob}((\sigma, M) \text{ satisfait } \mathbf{TC} | (\sigma, M))$

Afin d'évaluer  $\text{Prob}((\sigma, M) \text{ satisfait } \mathbf{TC} | (\sigma, M))$ , il faut tout d'abord évaluer la probabilité d'une trajectoire de mesures donnée  $TR_0$  et un ensemble de contraintes temporelles  $\mathbf{TC}$  avec (2.32) :

$$\text{Prob}((\sigma, M | TR_0 \text{ et } \mathbf{TC}) = \frac{\pi((\sigma, M) | TR_0 \text{ et } \mathbf{TC})}{\sum_{(\sigma', M') \in \Gamma_E^{-1}(TR_0)} \pi((\sigma', M') | TR_0 \text{ et } \mathbf{TC})} \quad (2.32)$$

où  $\pi((\sigma, M) | TR_0 \text{ et } \mathbf{TC})$  est définie par (2.33) :

$$\pi((\sigma, M) | TR_0 \text{ et } \mathbf{TC}) = \pi_0(M) \times \pi_1(\sigma | TR_0, M \text{ et } \mathbf{TC}) \times \pi_2(\sigma | TR_0 \text{ et } M) \quad (2.33)$$

où  $\pi_0(M)$  et  $\pi_2(\sigma|TR_O$  et  $M)$  sont définies et calculées dans la section 2.5. La probabilité  $\pi_1(\sigma|TR_O, M$  et  $TC)$  est évaluée telle que :

1. les dates de franchissement des transitions respectent l'ordre chronologique de la trajectoire  $\sigma$  :  $\tau_0 \leq t_{1,1} \leq \dots \leq t_{1,h_1-1} \leq \tau_1 \leq \dots \leq \tau_K \leq t_{K+1,1} \leq \dots \leq t_{K+1,h_{K+1}-1} \leq \tau_{end}$
2. les  $K$  dates de franchissement  $t_{k,h_k}$  correspondent aux  $\tau_k$  mesures et satisfont :  $t_{k,h_k} \in [\tau_k : \tau_k + dt]$  pour  $k = 1, \dots, K$
3. les durées de franchissement de toutes les transitions satisfont les contraintes temporelles  $TC$ . Pour chaque transition  $T(t'_{k,h}, t_{k,h})$ , la durée de franchissement  $u_{k,h}$  doit satisfaire les contraintes minimales et maximales de  $TC$  notées  $\delta(k, h)$  et  $\Delta(k, h)$ , et on définit la fonction  $\theta_{k,h}(u_{k,h})$  suivante telle que :

$$\begin{aligned} \theta_{k,h}(u_{k,h}) &= 1 \text{ si } u_{k,h} \in [\delta(k, h), \Delta(k, h)] \\ \theta_{k,h}(u_{k,h}) &= 0, \text{ sinon} \end{aligned} \quad (2.34)$$

Par conséquent,  $\pi_1(\sigma|TR_O, M$  et  $TC)$  est évaluée avec l'équation (2.35) similaire à (2.26). Cette expression est approchée grâce à l'algorithme 2.4. On note que cet algorithme est similaire à celui utilisé pour évaluer  $\pi_1(\sigma|TR_O$  et  $M)$  en incluant à chaque étape l'évaluation des  $TC$  avec la fonction  $\theta_T(u)$  :

$$\begin{aligned} \pi_1(\sigma|TR_O, M \text{ et } TC) &= \int_{\tau_0}^{\tau_1} g'_{1,1}(u_{1,1}) \left( \int_{t_{1,1}}^{\tau_1} g'_{1,2}(u_{1,2}) \dots \right. \\ &\quad \left. \dots \left( \int_{t_{1,h_1-1}}^{\tau_1} g'_{1,h_1-1}(u_{1,h_1-1}) \left( \int_{\tau_1}^{\tau_2} g'_{2,1}(u_{2,1}) \dots \right. \right. \right. \\ &\quad \left. \left. \left. \dots \left( \int_{t_{K+1,h_{K+1}-1}}^{\tau_{end}} g'_{K+1,h_{K+1}-1}(u_{K+1,h_{K+1}-1}) \cdot dt_{K+1,h_{K+1}-1} \right) \dots \right. \right. \\ &\quad \left. \left. \dots dt_{2,1} \right) dt_{1,h_1-1} \right) \dots dt_{1,2} \right) dt_{1,1} \end{aligned} \quad (2.35)$$

avec :

$$g'_{k,h}(u_{k,h}) = f'_{k,h}(u_{k,h}) \times \theta_{k,h}(u_{k,h}) \left( \prod_{(i,j) \in Cs(k,h)} f'_{i,j}(u_{i,j}) \times \theta_{i,j}(u_{i,j}) \right) \quad (2.36)$$

**Algorithme 2.4.** Calcul de  $\pi_1(\sigma | TR_0, M \text{ et } TC)$ 

**Entrées :**  $(\sigma, M), T, t'(T), [\tau_0 : \tau_{end}], dt, TC$

**Sortie :**  $\pi_1$

**Initialisation :**  $\pi_1 \leftarrow 0$

1. calculer  $Cs(T)$  and  $C(T)$
2. **pour**  $t \leftarrow \max(\tau_0, t'(T) + a(T)) : \min(\tau_{end}, t'(T) + b(T))$  avec un pas de  $dt$
3.  $T_{next} \leftarrow$  la prochaine transition du 3<sup>ème</sup> type dans  $(\sigma, M)$
4. **si**  $T_{next} \neq \mathcal{E}$ ,
5.  $t_{next} \leftarrow$  la date de franchissement de la dernière transition qui tir avant  $T_{next}$
6.  $r_1 \leftarrow$  **Algorithme 2.4**  $((\sigma, M), T_{next}, t'(T_{next}), [t_{next} : \tau_{end}], TC)$
7. **sinon**  $r_1 \leftarrow 1$ ,
8. **fin si**
9.  $u \leftarrow t - t'(T), r_2 \leftarrow f_T(u) \times \theta_T(u)$
10. **si**  $C(T) \neq \emptyset$ , corriger  $r_2$  par (2.21) ou (2.23) selon la politique de choix
11. **fin si**
12. **pour** chaque  $T' \in Cs(T)$
13.  $u' \leftarrow \tau(T') - t, r_2 \leftarrow r_2 \times f_{T'}(u') \times \theta_{T'}(u')$
14. **fin pour**
15.  $\pi_1 \leftarrow \pi_1 + r_1 \times r_2 \times dt$
16. **fin pour**

Enfin,  $\text{Prob}((\sigma, M) \text{ satisfait } TC | (\sigma, M))$  est obtenue selon  $\text{Prob}((\sigma, M) \text{ satisfait } TC | (\sigma, M)) = \pi((\sigma, M) | TR_0 \text{ et } TC) / \pi((\sigma, M) | TR_0)$ . Compte tenu du fait que  $\pi_0(M)$  et  $\pi_2(\sigma | TR_0 \text{ et } M)$  sont inchangées :

$$\text{Prob}((\sigma, M) \text{ satisfait } TC | (\sigma, M)) = \frac{\pi_1(\sigma | TR_0, M \text{ et } TC)}{\pi_1(\sigma | TR_0 \text{ et } M)} \quad (2.37)$$



La probabilité que  $TR_O$  satisfait les contraintes temporelles  $\mathbf{TC}$  est la somme pondérée des probabilités  $\text{Prob}((\sigma, M) \text{ satisfait } \mathbf{TC} | (\sigma, M))$  pour toutes les trajectoires compatibles de l'ensemble  $\Gamma_E^{-1}(TR_O)$ . Cette probabilité est définie par (2.38) :

$$\begin{aligned} \text{Prob}(TR_O \text{ satisfait } \mathbf{TC}) &= \sum_{(\sigma, M) \in \Gamma_E^{-1}(TR_O)} \text{Prob}((\sigma, M) | TR_O) \\ &\quad \times \text{Prob}((\sigma, M) \text{ satisfait } \mathbf{TC} | (\sigma, M)) \end{aligned} \quad (2.38)$$

Finalement, la probabilité qu'un défaut temporel se soit produit dans un intervalle de temps  $[\tau_0, \tau_{end}]$  sachant que  $TR_O$  est la trajectoire compatible réalisée, est donnée par (2.39) :

$$\text{Prob}(F_{TC} | TR_O) = 1 - \text{Prob}(TR_O \text{ satisfait } \mathbf{TC}) \quad (2.39)$$

**Exemple 2.5.** Reprenons l'exemple 2.1 avec les mêmes paramètres ainsi qu'une trajectoire de mesures  $TR_O = (0, 0) (e, 8) (0, 8)$  collectée dans l'intervalle de temps  $[0 : 8]$  avec  $M(0) = M_I = (2 \ 0 \ 0)^T$ . Les contraintes temporelles considérées sont :  $\mathbf{TC}(T_1) = [2 : 8]$  et  $\mathbf{TC}(T_2) = [2 : 3]$ . Un défaut temporel est donc défini par (2.40) :

$$F_{TC}(\sigma) = (d_1 \notin [2 : 8]) \vee (d_2 \notin [2 : 3]) \quad (2.40)$$

Les deux trajectoires compatibles avec  $TR_O$  sont :  $(\sigma_a, M_I)$  avec  $\sigma_a = T_1(t_{1,1})T_2(8)$  d'une probabilité de 0.44 et  $(\sigma_b, M_I)$  avec  $\sigma_b = T_1(t_{1,1})T_1(t_{1,2})T_2(8)$  d'une probabilité de 0.56. Grâce à (2.35) et l'algorithme 2.4, on obtient le résultat suivant :  $\text{Prob}((\sigma_a, M_I) \text{ satisfait } \mathbf{TC} | (\sigma_a, M_I)) = \text{Prob}((\sigma_b, M_I) \text{ satisfait } \mathbf{TC} | (\sigma_b, M_I)) = 0.35$ . Finalement,  $\text{Prob}(TR_O \text{ satisfait } \mathbf{TC}) = 0.35$  et  $\text{Prob}(F_{TC} | TR_O) = 0.65$ .

## 2.7. Cas d'étude

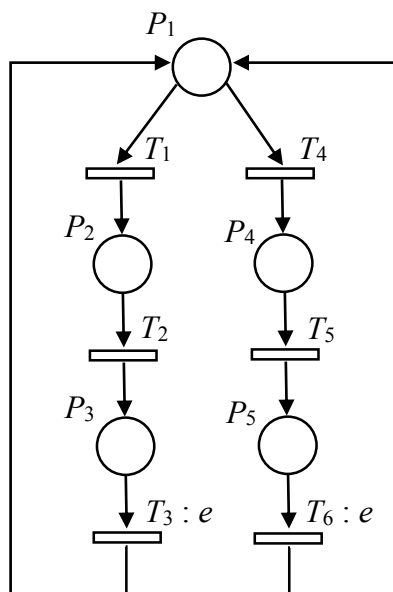


Figure 2.5. POTPN d'un processus d'impression 3D

On considère un procédé de fabrication à base d'impression tridimensionnelle aussi connu sous le nom d'impression 3D. L'imprimante traite deux types de produits en fonction de deux processus A et B avec des spécifications différentes. Les deux processus sont modélisés par le POTPN représenté sur la figure 2.5. Le marquage initial est supposé être connu.

Le processus A est représenté par les trois transitions  $T_1$ ,  $T_2$  et  $T_3$ , tandis que les transitions  $T_4$ ,  $T_5$  et  $T_6$  détaillent le processus B. Pour chaque processus, les opérations se résument en trois phases successives : la préparation qui est représentée par les transitions  $T_1$  et  $T_4$ , l'impression 3D représentée par les transitions  $T_2$  et  $T_5$  et la gélification représentée par les transitions  $T_3$  et  $T_6$ . Chaque transition a une densité de probabilité uniforme avec un support fini détaillé dans le tableau 2.3. On remarque que les temps moyens d'exécution des deux processus sont légèrement différents : la durée globale du processus B est un peu plus longue en moyenne (248.5 s) que la durée globale du processus A (233.5 s). Cette différence sera utilisée pour évaluer la probabilité de chaque processus.

Une surveillance de ce procédé de fabrication est effectuée grâce à la configuration de capteurs suivante : seules les deux transitions  $T_3$  et  $T_6$  sont mesurées et délivrent un label identique :  $L(T_3) = L(T_6) = e$ . Le marquage n'est pas mesuré :  $H = (0 \ 0 \ 0 \ 0 \ 0)^T$ . On considère la trajectoire de mesures qui correspond à

l'exécution d'un seul processus :  $TR_O = (0,0) (e, \tau) (0, \tau)$ . A partir de cette trajectoire on peut faire les trois constatations suivantes : 1) le procédé de fabrication est surveillé durant l'intervalle de temps  $[0 : \tau]$  ; 2) le label  $e$  est collecté à l'instant  $\tau$  ; 3) aucun marquage n'est mesuré pendant l'intervalle de temps  $[0 : \tau]$ . Deux scénarios sont considérés par la suite et pour chaque scénario les deux cas suivants sont étudiés pour chaque processus : une seule exécution simultanée (c'est-à-dire  $M_I = (1 \ 0 \ 0 \ 0 \ 0)^T$ ) et deux exécutions simultanées (c'est-à-dire  $M_I = (2 \ 0 \ 0 \ 0 \ 0)^T$ ). On note que, ce POTPN satisfait les trois hypothèses H2.1, H2.2 et H2.3.

➤ 1<sup>er</sup> scénario : le procédé de fabrication est supposé délivrer des produits uniquement fabriqués par le processus A. On considère le défaut structurel correspondant une exécution du processus B. Celui-ci est modélisé par le motif  $\mathbf{R} = \{(T_4T_5T_6), \subset\}$  et donc :  $F_R(\sigma) = ((T_4T_5T_6) \subset \sigma)$ . On note que dans un contexte non temporisé, le motif  $\mathbf{R}$  n'est pas diagnosticable alors que dans le cas temporisé la probabilité de  $\mathbf{R}$  est obtenue grâce à la date de mesure de l'événement  $e$ . Par conséquent, le système devient diagnosticable. Si le marquage initial est  $M_I = (1 \ 0 \ 0 \ 0 \ 0)^T$  alors le nombre maximum de franchissements successifs de transitions silencieuses est 2 et 2 séquences de franchissements  $T_1T_2T_3$ , et  $T_4T_5T_6$  sont compatibles avec  $TR_O$ . La figure 2.6 représente la probabilité de ces trajectoires compatibles avec  $TR_O$  lorsque  $\tau$  varie dans l'intervalle  $[225 \text{ s} : 255 \text{ s}]$ . La probabilité que le défaut structurel modélisé par le motif  $\mathbf{R}$  apparaisse, coïncide avec la probabilité de la trajectoire  $(M_I, T_4T_5T_6)$ . Le calcul de ces probabilités est effectué grâce à l'équation (2.26) et l'algorithme 2.3. La simulation de Monte-Carlo (les lignes en pointillées) utilisée dans cet exemple permet de confirmer le résultat du diagnostic. Les simulations sont effectuées afin d'obtenir un ensemble de 100 trajectoires dans un intervalle de  $[0 : \tau]$  menant à une trajectoire de mesures  $(0,0) (e, \tau') (0, \tau')$  similaire à  $TR_O$  avec  $\tau' \in [\tau : \tau+1]$  c'est-à-dire qu'une différence d'au plus 1s est tolérée sur la date de mesure de  $e$ .

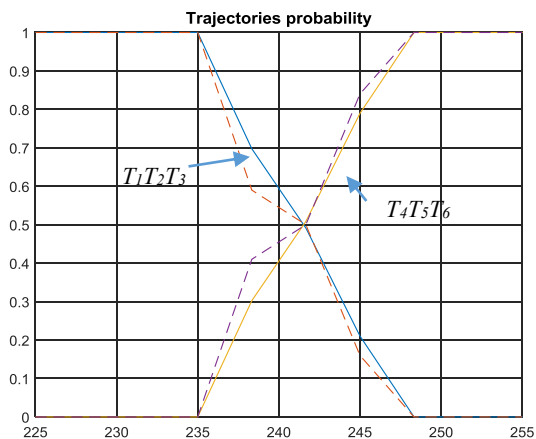
Si le marquage initial est  $M_I = (2 \ 0 \ 0 \ 0 \ 0)^T$  alors le nombre maximum de franchissements successifs de transitions silencieuses est 4 ainsi 28 séquences de franchissement non temporisées sont compatibles avec  $TR_O$  mais seules quelques-unes sont de probabilité non nulle. La figure 2.7 représente la probabilité des trajectoires compatibles avec  $TR_O$  sachant que  $\tau$  varie dans l'intervalle  $[225 \text{ s} : 255 \text{ s}]$ . La probabilité que le défaut structurel modélisé par le motif  $\mathbf{R}$  apparaisse, coïncide avec la probabilité de la trajectoire  $(M_I, T_4T_4T_5T_5T_6)$ . Le calcul de ces probabilités est confirmé par les simulations Monte-Carlo. On peut également vérifier que, malgré le grand nombre de trajectoires compatibles lorsque  $M_I = (2 \ 0 \ 0 \ 0 \ 0)^T$ , la probabilité d'un défaut structurel  $\mathbf{R}$  ne dépend pas de l'exécution unique ou multiple du processus. Ce résultat était attendu car les deux jetons sont en concurrence et le comportement défectueux est défini dans ce scénario comme une exécution complète du cycle à droite.

➤ 2<sup>ème</sup> scénario : le procédé de fabrication est supposé délivrer des produits fabriqués par le processus A ou B. Un défaut temporel est constaté sur les opérations de l'impression 3D si ces dernières ne satisfont pas les spécifications temporelles représentées dans tableau 2.3. On considère le défaut temporel :  $F_{TC}(\sigma) = (d_2 \notin TC(T_2)) \vee (d_5 \notin TC(T_5))$ . Si  $M_I = (1 \ 0 \ 0 \ 0 \ 0)^T$ . La figure 2.8 représente la probabilité des trajectoires compatibles avec  $TRo$  qui ne satisfont pas les contraintes temporelles  $TC$  sachant que  $\tau$  varie dans l'intervalle [225 s : 255 s]. La figure 2.9 représente la probabilité qu'un défaut temporel apparaisse. Le calcul de ces probabilités est effectué grâce à l'équation (2.35) et l'algorithme 2.4. Des simulations de Monte-Carlo permettent de confirmer ce résultat du diagnostic.

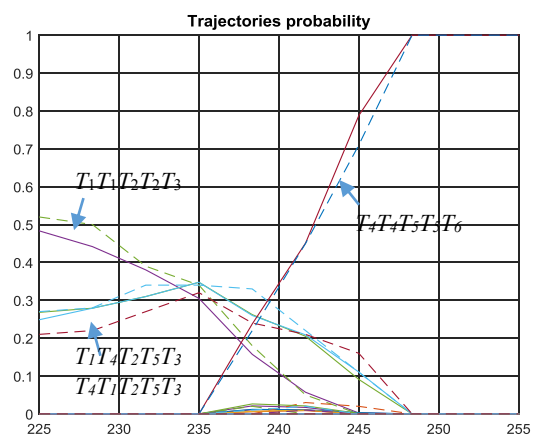
Si  $M_I = (2 \ 0 \ 0 \ 0 \ 0)^T$ , la figure 2.10 représente la probabilité des trajectoires compatibles avec  $TRo$  qui ne satisfont pas les contraintes temporelles  $TC$  et la figure 2.11 représente la probabilité qu'un défaut temporel apparaisse. Le calcul de ces probabilités est confirmé par des simulations Monte-Carlo. On peut également remarquer que, la probabilité d'occurrence du défaut temporel dépend de l'exécution unique ou multiple du processus. Avec deux produits traités simultanément, la probabilité du défaut temporel est d'au moins 0.4. Même si le produit délivré à la date  $\tau$  satisfait les contraintes temporelles  $TC$ , le second produit en cours d'exécution a toujours une probabilité non nulle de violer ces contraintes.

**Tableau 2.3.** La signification des opérations

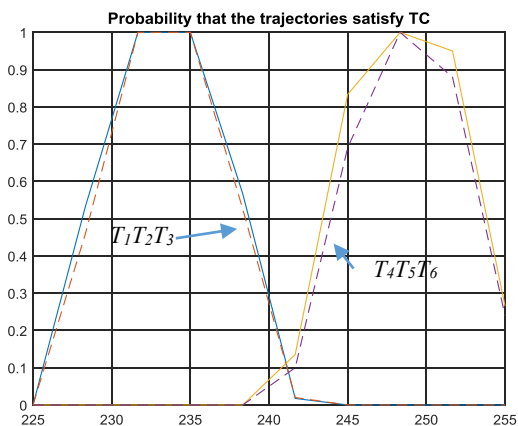
$T$	Significations	Durée [a : b]	Tolérance [ $\delta$ : $\Delta$ ]
$T_1$	Préparation du processus A	[10 : 12]	[10 : 12]
$T_2$	Impression 3D pour le processus A	[10 : 30]	[15 : 25]
$T_3$	Gélification pour le processus A	[200 : 205]	[200 : 205]
$T_4$	Préparation du processus B	[10 : 12]	[10 : 12]
$T_5$	Impression 3D pour le processus B	[25 : 45]	[30 : 40]
$T_6$	Gélification pour le processus B	[200 : 205]	[200 : 205]



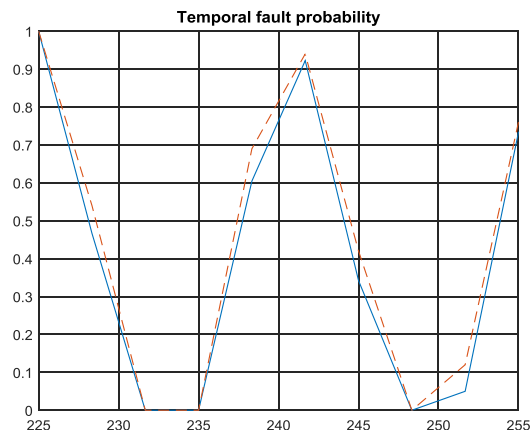
**Figure 2.6.** Probabilité des trajectoires compatibles et probabilité du défaut structurel  $\mathbf{R}$  pour  $M_I = (1\ 0\ 0\ 0\ 0)^T$



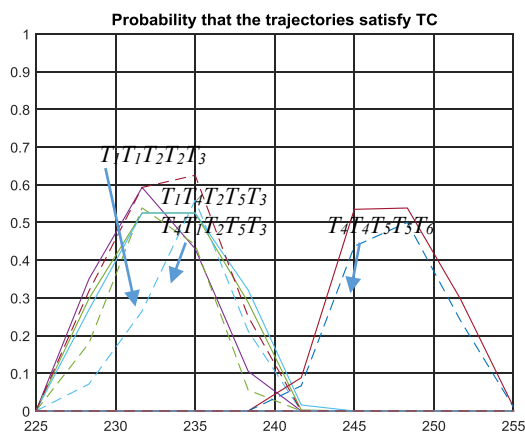
**Figure 2.7.** Probabilité des trajectoires compatibles et probabilité du défaut structurel  $\mathbf{R}$  pour  $M_I = (2\ 0\ 0\ 0\ 0)^T$



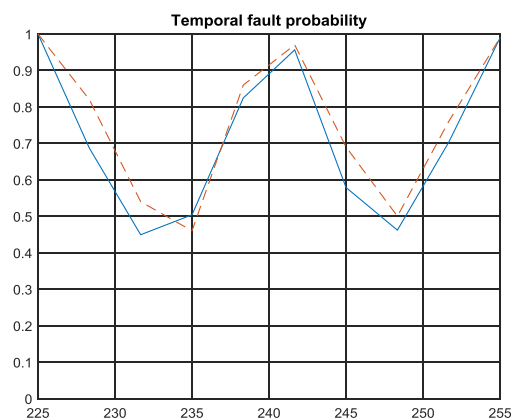
**Figure 2.8.** Probabilité que la trajectoire compatible satisfait  $TC$  pour  $M_I = (1\ 0\ 0\ 0\ 0)^T$



**Figure 2.9.** Probabilité du défaut temporel pour  $M_I = (1\ 0\ 0\ 0\ 0)^T$



**Figure 2.10.** Probabilité que la trajectoire compatible satisfait  $TC$  pour  $M_I = (2 \ 0 \ 0 \ 0 \ 0)^T$



**Figure 2.11.** Probabilité du défaut temporel pour  $M_I = (2 \ 0 \ 0 \ 0 \ 0)^T$

## 2.8. Conclusion

Dans ce chapitre, nous nous sommes intéressés au diagnostic des SED par le calcul des probabilités des trajectoires compatibles. Pour cela, les RdP stochastiques partiellement mesurés sont employés pour modéliser le système avec ses capteurs. Ces réseaux sont caractérisés par une mesure partielle à la fois des transitions et des marquages. Ainsi, à partir d'une trajectoire de mesures obtenue lors du fonctionnement du système et composée d'événements et de marquages mesurés, l'objectif est de détecter l'occurrence des défauts structurels et temporels. Cette méthode est particulièrement utile pour les SED qui se comportent selon une dynamique non markovienne. La première contribution a consisté à déterminer l'ensemble des comportements compatibles avec les mesures. Nous avons montré que cette détermination peut être effectuée soit grâce à une résolution d'inégalités matricielles linéaires soit à l'aide d'un algorithme incrémental représentant une diminution significative de la complexité des calculs par rapport à la première méthode. Ensuite, nous nous sommes intéressés au calcul des probabilités de chaque comportement, compatible avec les mesures. Finalement, la probabilité des défauts structurels et temporels est obtenue grâce à l'exploitation de ces probabilités. Ce résultat, nous a permis d'étendre la définition de défaut pour représenter des comportements plus complexes incluant des contraintes temporelles. Le résultat est satisfaisant mais l'effort de calcul reste important. Afin de lever ces difficultés, nous proposons dans le chapitre suivant un nouvel outil pour diagnostiquer les RdP temporisés partiellement mesurés. Contrairement à la détection directe et instantanée,

---

cet outil permet de détecter et confirmer des comportements dégradés du système en se basant sur des mesures moyennes de certains paramètres.

# Chapitre 3. Diagnostic par carte de contrôle

3.1. Introduction .....	91
3.2. Diagnostic et défauts temporels.....	91
3.2.1. Définitions.....	91
3.2.2. Modélisation des défauts temporels.....	92
A. Hypothèses de travail .....	92
B. Variation des supports des lois des distributions des durées.....	93
3.3. Carte de contrôle .....	95
3.3.1. Cartes de contrôle classiques de Shewhart .....	97
A. La carte de la moyenne / étendue ( $\bar{X}, R$ ) .....	98
B. La carte de la moyenne / écart-type ( $\bar{X}, S$ ) .....	98
C. La carte de la médiane / étendue ( $\tilde{X}, R$ ) .....	98
3.3.2. Carte de contrôle à somme cumulée (CuSum).....	100
3.3.3. Carte de contrôle à Moyenne Mobile pondérée exponentiellement (EWMA).....	101
3.3.4. Carte de contrôle à Moyenne Mobile (MM) .....	102
3.4. Table de signatures et diagnostic .....	106
3.4.1. Conception de la table de signatures .....	106
3.4.2. Détection et localisation des défauts temporels.....	107
3.5. Exemple .....	108
3.6. Conclusion.....	112





### 3.1. Introduction

La complexité croissante des SED qui sont de plus en plus exigeants en termes de contraintes temporelles [Rachidi et al. 2017 ; Rachidi et al. 2019], nécessite une exploitation profonde des informations pouvant affecter l'évolution temporelle du processus. Parmi les outils d'analyses des données, les cartes de contrôles statistiques [Saporta 2006, Benneyan 2001 ; Ali et al. 2016] ont été largement exploitées en milieu industriel. Dans ce chapitre, nous proposons d'adapter le principe des cartes de contrôle de type Moyenne Mobile (Moving Average control chart / Moving Mean chart ou (MM)) pour assurer la surveillance des évolutions temporelles des SED. Pour se faire, les RdPS T-temporisés partiellement mesurés (RdPSTPO) sont utilisés comme outil de modélisation [Ru et Hadjicostis 2009].

L'objectif principal de cette étude est de pouvoir détecter des défauts temporels grâce à l'application d'une carte de contrôle MM [Zhou et Tang 2016]. Les défauts temporels sont définis comme des variations inacceptables de la moyenne mobile des durées de franchissement des transitions. Notre approche est basée sur la détection de ces défauts temporels en utilisant une carte de contrôle MM applicable lorsque nous avons suffisamment de mesures des durées de franchissement. Contrairement à la détection directe et instantanée, qui est parfois difficile à mettre en œuvre, le principal avantage d'une détection à base de carte de contrôle MM, est de pouvoir déterminer les variations lentes des durées moyennes et de confirmer les changements lorsqu'ils se sont réellement produits. Les défauts temporels (en particuliers les dérives) sont mis en évidence par l'utilisation d'une carte de contrôle MM sur une fenêtre glissante de taille fixe. Une table de signatures est ensuite introduite pour isoler le défaut parmi plusieurs candidats.

### 3.2. Diagnostic et défauts temporels

#### 3.2.1. Définitions

Dans ce chapitre, on considère les RdPS T-temporisés partiellement mesurés munis d'une politique de choix par présélection, d'une politique de service avec serveur non-réentrant et enfin d'une politique de mémoire qui prend en compte uniquement les durées relatives à la dernière sensibilisation du système (enabling memory) (voir Chapitre 1, Section 1.5). Les définitions suivantes sont également introduites :

**Un chemin**, dans un réseau de Petri, est une suite alternée de places et de transitions telle que :  $Path(T(1), T(h)) = T(1)P(1)T(2)P(2) \dots T(i)P(i) \dots T(h-1)P(h-1)T(h)$  où  $P(i) \in T(i)^\circ$  et  $T(i+1) = P(i)^\circ$  pour  $i \in \{1, \dots, h-1\}$ .  $h$  est la taille du chemin.

**La date de tir** de la transition  $T(i)$  est notée  $\tau(T(i))$ .

**La durée d'un chemin**  $d(T(1), T(h))$  est le temps écoulé entre les dates de franchissement des deux transitions extrêmes de ce chemin :  $d(T(1), T(h)) = \tau(T(h)) - \tau(T(1))$ .

**Un chemin silencieux** noté  $(Path_{sil})$  est un chemin où toutes les places et transitions sont silencieuses.

**Un chemin semi-silencieux** noté  $(Path_s)$  est un chemin où seules les dates des deux transitions extrêmes  $T(1)$  et  $T(h)$  sont mesurées.

### 3.2.2. Modélisation des défauts temporels

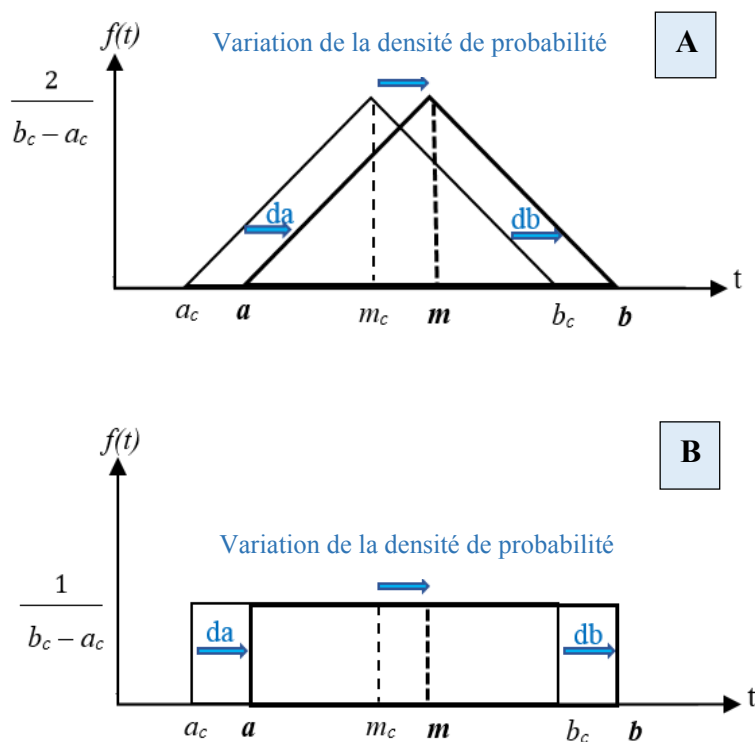
#### A. Hypothèses de travail

L'objectif de ce travail est de détecter une translation du support des lois de probabilité qui définissent les durées de tir des transitions dans un RdPSTPO. Cette variation est détectée à l'aide d'une carte de contrôle MM appliquée aux durées de franchissement des transitions. Les hypothèses suivantes seront respectées tout au long de ce chapitre :

- Une seule marque circule dans le réseau considéré.
- Les RdPSTPO considérés sont réinitialisables (cette condition est nécessaire pour pouvoir obtenir plusieurs tirs d'une même transition).
- En cas de conflit entre les transitions, une politique de présélection est appliquée avec un poids  $\omega_j = 1$  pour toutes les transitions  $T_j$  en conflit.
- Le type de la distribution des durées ainsi que son support initial sont supposés connus.
- Le type de la distribution des durées est invariant.
- Deux types de défauts n'apparaissent pas simultanément.
- Les seuils de détection de la carte de contrôle sont imposés selon les tolérances souhaitées.
- Un seul chemin semi-silencieux est compatible avec la mesure de deux étiquettes consécutives  $(\ell_{obs_k}, \tau_k)$   $(\ell_{obs_{k+1}}, \tau_{k+1})$ .

## B. Variation des supports des lois des distributions des durées

La variation de la moyenne des durées de franchissement des transitions est le défaut auquel on s'intéresse dans cette étude. Les durées de franchissement des transitions sont des VA définies par leurs densités de probabilité. Une translation du support de la loi conduit à une variation de la moyenne de la durée de tir qui est détectée, dans ce qui suit, par une carte de contrôle. Dans la figure 3.1 on suppose une translation d'une loi triangulaire symétrique ou uniforme bornée de support  $[a_c, b_c]$  et de moyenne  $m_c$  vers une autre loi de même type avec des paramètres modifiés  $a$ ,  $b$  et  $m$ .



**Figure 3.1.** Variation de support de la loi : A) triangulaire symétrique ; B) uniforme bornée

Soit un RdPSTPO. Les durées de franchissement d'une transition donnée  $T$  sont réparties selon des lois définies de densités de probabilité connues de support  $[a_c, b_c]$ . Le calcul de la moyenne et de l'écart-type des durées de  $N$  tirs consécutifs de la même transition, permet de déduire une variation du support de la loi. Par la suite, le nouveau support de la densité de probabilité  $[a, b]$  est calculé. Ce résultat est donné par la proposition suivante.

**Proposition 3.1.** Les durées de  $N$  franchissements consécutifs de la transition  $T$  d'une loi uniforme bornée sont mesurées. La moyenne  $m$  et l'écart-type  $\sigma$  de ces mesures permettent d'estimer le support  $[a, b]$  de la loi par l'équation (3.1) :

$$\begin{aligned} a &= m - \sigma \times 3^{1/2} \\ b &= m + \sigma \times 3^{1/2} \end{aligned} \quad (3.1)$$

**Démonstration.** La moyenne et l'écart-type de  $N$  mesures des durées de franchissement de la transition  $T$  d'une loi uniforme bornée, sont calculées par l'équation (3.2) :

$$\begin{aligned} m &= (a + b) / 2 \\ \sigma &= (b - a) / (2 \times 3^{1/2}) \end{aligned} \quad (3.2)$$

A partir de l'équation (3.2) on obtient directement (3.1).

Ensuite, les variations  $da$  et  $db$  des deux extrémités du support sont obtenues par l'équation (3.3) :

$$\begin{aligned} da &= a - a_c = m - \sigma \times 3^{1/2} - a_c \\ db &= b - b_c = m + \sigma \times 3^{1/2} - b_c \end{aligned} \quad (3.3)$$

**Proposition 3.2.** Les durées de  $N$  franchissements consécutifs de la transition  $T$  d'une loi triangulaire symétrique sont mesurées. La moyenne  $m$  et l'écart-type  $\sigma$  de ces mesures permettent d'estimer le support  $[a, b]$  de la loi par l'équation (3.4) :

$$\begin{aligned} a &= m - \sigma \times 6^{1/2} \\ b &= m + \sigma \times 6^{1/2} \end{aligned} \quad (3.4)$$

**Démonstration.** La moyenne et l'écart-type de  $N$  mesures des durées de franchissement de la transition  $T$  d'une loi triangulaire symétrique, sont calculées par l'équation (3.5) :

$$\begin{aligned} m &= (a + b) / 2 \\ \sigma &= (b - a) / (2 \times 6^{1/2}) \end{aligned} \quad (3.5)$$

Ensuite, les variations  $da$  et  $db$  des deux extrémités du support sont obtenues par l'équation (3.6) :

$$\begin{aligned} da &= m - \sigma \times 6^{1/2} - a_c \\ db &= m + \sigma \times 6^{1/2} - b_c \end{aligned} \quad (3.6)$$

### 3.3. Carte de contrôle

La carte de contrôle est une technique statistique importante utilisée dans le contrôle statistique des processus (Statistical Process Control ou (SPC)). Elle est fréquemment utilisée pour améliorer la qualité et la fiabilité des produits ainsi que la maîtrise de procédés. La mise en place d'une carte de contrôle se réalise à partir d'échantillonnage, les données ainsi collectées étant soumises à une analyse rétrospective pour déterminer les situations hors contrôle.

Parmi les différents types de cartes de contrôle, on peut citer les cartes de contrôle classiques de Shewhart [Shewhart 1931] qui sont utilisées principalement pour détecter les grandes variations, la carte de contrôle à somme cumulée (Cumulative Sum ou (CuSum)) [Pages 1954 ; Box et Luceno 1997 ; Annadi et al. 1995], la carte de contrôle à Moyenne Mobile pondérée exponentiellement (Exponentially Weighted Moving Average ou (EWMA)) [Roberts 1959 ; Lucas et Saccucci 1990 ; Montgomery 1991 ; Zhang 2006] ainsi que la carte de contrôle à Moyenne Mobile (Moving Mean ou Moving Average ou (MM)). Ces trois cartes de contrôle (CuSum, MM et EWMA) sont spécialement conçues pour détecter les variations de faible et moyenne amplitude dans un processus. À mesure que l'on prélève des échantillons on obtient une image de l'évolution du processus (figure 3.2). Le processus est considéré sous contrôle si les mesures restent entre les limites de contrôle ( $S_{sup}$  - limite de contrôle supérieure et  $S_{inf}$  - limite de contrôle inférieure). Lorsqu'une mesure sort du domaine de contrôle, une alarme est générée pour prévenir l'opérateur.

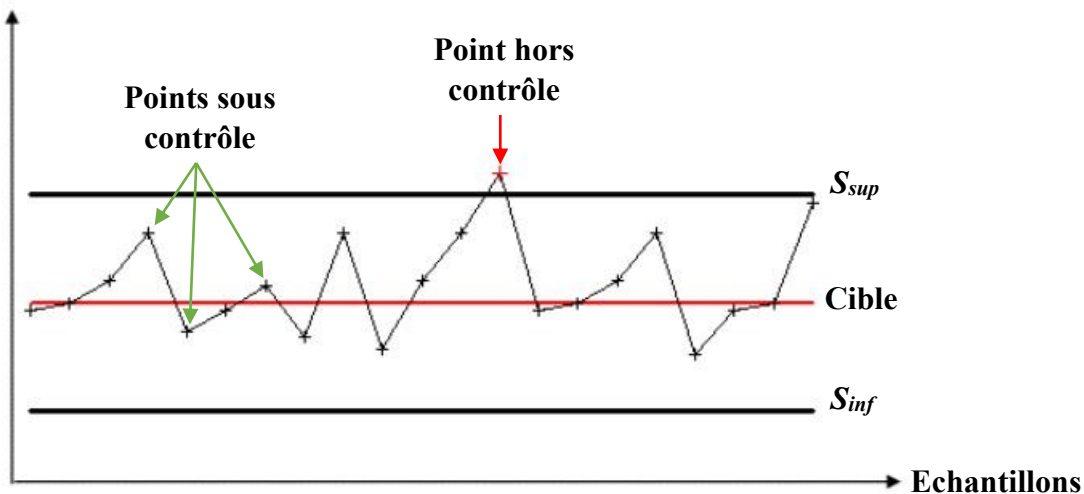


Figure 3.2. Image d'une carte de contrôle

On regroupe les données  $X_{i,j}$  dans le tableau de Maîtrise Statistique des Procédés (MSP) suivant :

Tableau 3.1. Tableau MSP de  $m \times n$  mesures  $X_{i,j}$

Echantillons	Mesures
1	$X_{1,1}$ $X_{1,2}$ $X_{1,3}$ ... $X_{1,n}$
2	$X_{2,1}$ $X_{2,2}$ $X_{2,3}$ ... $X_{2,n}$
⋮	⋮ ⋮ ⋮ ... ⋮
$m$	$X_{m,1}$ $X_{m,2}$ $X_{m,3}$ ... $X_{m,n}$

A partir de ce tableau, il est possible de calculer un certain nombre de caractéristiques importantes. Pour chaque échantillon  $i$  du tableau de données MSP, on peut calculer la moyenne  $\bar{X}_i$ , l'étendue  $R_i$ , l'écart-type  $S_i$  et la médiane empirique  $\tilde{X}_i$ . A partir de ces quantités et grâce aux équations suivantes, on peut aussi calculer sur une fenêtre de  $N$  valeurs consécutives :

- La moyenne des moyennes :

$$\bar{\bar{X}} = \frac{1}{m} \sum_{i=1}^m \bar{X}_i \quad (3.7)$$

- La moyenne des étendues :

$$\bar{R} = \frac{1}{m} \sum_{i=1}^m R_i \quad (3.8)$$

- La moyenne des écart-types :

$$\bar{S} = \frac{1}{m} \sum_{i=1}^m S_i \quad (3.9)$$

- La moyenne des médianes empiriques :

$$\bar{\tilde{X}}_i = \frac{1}{m} \sum_{i=1}^m \tilde{X}_i \quad (3.10)$$

### 3.3.1. Cartes de contrôle classiques de Shewhart

Les cartes de contrôle de Shewhart sont des outils graphiques, utilisées pour suivre le comportement et les résultats d'un processus au cours du temps. Les cartes de contrôle classiques de Shewhart qui sont généralement utilisées pour la détection d'une dérive en position sont :

- La carte de la moyenne / étendue, i.e. carte de contrôle  $(\bar{X}, R)$
- La carte de la moyenne / écart-type, i.e. carte de contrôle  $(\bar{X}, S)$
- La carte de la médiane / étendue, i.e. carte  $(\tilde{X}, R)$

Par exemple, une carte de contrôle de Shewhart de type  $(\bar{X}, R)$  regroupe une carte des moyennes des échantillons et une carte des étendues des échantillons, permettant par conséquent de suivre à la fois la moyenne et la dispersion du processus afin d'identifier la présence d'anomalies. Les limites de contrôle  $S_{inf(\bar{X},R)}$  et  $S_{sup(\bar{X},R)}$  de cette carte s'obtiennent comme suit :



### A. La carte de la moyenne / étendue ( $\bar{X}, R$ )

Cette carte de contrôle ( $\bar{X}, R$ ) est définie par ces deux limites de contrôle calculées comme suit :

$$S_{inf(\bar{X},R)} = \bar{\bar{X}} - A_{(\bar{X},R)}(n) \times \bar{R} \quad (3.11)$$

$$S_{sup(\bar{X},R)} = \bar{\bar{X}} + A_{(\bar{X},R)}(n) \times \bar{R} \quad (3.12)$$

où  $A_{(\bar{X},R)}(n)$  est un coefficient listé dans le tableau 3.2.

### B. La carte de la moyenne / écart-type ( $\bar{X}, S$ )

Cette carte de contrôle ( $\bar{X}, S$ ) est définie par ces deux limites de contrôle calculées comme suit :

$$S_{inf(\bar{X},S)} = \bar{\bar{X}} - A_{(\bar{X},S)}(n) \times \bar{S} \quad (3.13)$$

$$S_{sup(\bar{X},S)} = \bar{\bar{X}} + A_{(\bar{X},S)}(n) \times \bar{S} \quad (3.14)$$

où  $A_{(\bar{X},S)}(n)$  est un coefficient listé dans le tableau 3.2.

### C. La carte de la médiane / étendue ( $\tilde{X}, R$ )

Cette carte de contrôle ( $\tilde{X}, R$ ) est définie par ces deux limites de contrôle calculées comme suit :

$$S_{inf(\tilde{X},R)} = \bar{\tilde{X}} - A_{(\tilde{X},R)}(n) \times \bar{R} \quad (3.15)$$

$$S_{sup(\tilde{X},R)} = \bar{\tilde{X}} + A_{(\tilde{X},R)}(n) \times \bar{R} \quad (3.16)$$

où  $A_{(\tilde{X},R)}(n)$  est un coefficient listé dans le tableau 3.2.

**Tableau 3.2.** Tableau des Coefficients intervenant dans le calcul des limites de contrôle des cartes de Shewhart

n	$A_{(\bar{x},R)}(n)$	$A_{(\bar{x},S)}(n)$	$A_{(\bar{x},R)}(n)$
2	1.8800	2.6587	-
3	1.0233	1.9544	1.1972
4	0.7286	1.6281	-
5	0.5768	1.4273	0.6962
6	0.4832	1.2871	-
7	0.4193	1.1819	0.5123
8	0.3725	1.0991	-
9	0.3367	1.0317	0.4140
10	0.3083	0.9754	-
11	0.2851	0.9274	0.3519
12	0.2658	0.8859	-
13	0.2494	0.8495	0.3087
14	0.2354	0.8173	-
15	0.2231	0.7885	0.2766
16	0.2123	0.7626	-
17	0.2028	0.7391	0.2518
18	0.1943	0.7176	-
19	0.1866	0.6979	0.2319
20	0.1796	0.6797	-
21	0.1733	0.6629	0.2155
22	0.1675	0.6473	-
23	0.1621	0.6327	0.2018
24	0.1572	0.6191	-
25	0.1526	0.6063	0.1901

Toutefois, les cartes de contrôle de Shewhart ne sont pas efficaces pour détecter de petites variations dans les variables du processus. En effet la conclusion d'une carte de contrôle de Shewhart est simplement basée sur l'échantillon actuel, ce qui signifie qu'elle n'accorde aucune attention aux données antérieures qui ont entraîné une relative mauvaise performance pour de petites variations dans le processus. En revanche, des versions multivariées d'autres cartes de contrôle connues ont été proposées pour une détection et précision plus importantes.

### 3.3.2. Carte de contrôle à somme cumulée (CuSum)

La carte CuSum a été introduite à l'origine par Pages en 1954. Le principe de cette carte est de sommer les écarts entre les moyennes et la valeur cible de telle manière que si le procédé s'éloigne significativement de cette valeur cible, le cumul des écarts va croître, et permettra donc de détecter une dérive. Tant que ces écarts restent au voisinage de la valeur cible, le processus est dit sous contrôle, sinon il est hors de contrôle.

Soit  $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_k$  une suite de variables aléatoires indépendantes correspondant aux moyennes calculées sur des échantillons de  $n$  variables aléatoires  $X_{j,k}$  de moyenne  $\mu_0$  et d'écart type  $\sigma_0$ . A partir de cette suite, on définit deux nouvelles suites à valeurs positives :

$$Y_k^+ = \max \{0, \bar{X}_k - (\mu_0 + H) + Y_{k-1}^+\} \quad (3.17)$$

$$Y_k^- = \max \{0, (\mu_0 - H) - \bar{X}_k + Y_{k-1}^-\} \quad (3.18)$$

où  $H = H_0 \cdot \sigma_0$  est une valeur de référence avec  $H_0$  une constante qui dépend de  $n$ .

La suite de variables aléatoires  $Y_1^+, \dots, Y_k^+$  permet de détecter une dérive positive de la moyenne, alors que la suite  $Y_1^-, \dots, Y_k^-$  permet, quant à elle, de détecter une dérive négative.

En général on prend,  $Y_0^+ = Y_0^- = 0$ . Une dérive du procédé sera détectée à l'étape  $k$  si :

$$\max (Y_k^-, Y_k^+) > S_{sup} \quad (3.19)$$

où  $S_{sup} = S_{sup_0} \cdot \sigma_0$  est un seuil et où  $S_{sup_0}$  est une constante qui dépend de  $n$ .

### 3.3.3. Carte de contrôle à Moyenne Mobile pondérée exponentiellement (EWMA)

Une carte de contrôle EWMA (figure 3.3) aussi appelée carte de contrôle avec mémoire, considère l'historique de l'évolution de la caractéristique suivie. Elle a été introduite par Roberts en 1959 pour traiter particulièrement les dérives de faibles et moyennes amplitudes. De manière similaire à la carte CuSum, la carte EWMA utilise également les informations passées, cependant des poids pondèrent les mesures selon une exponentielle décroissante qui décroît selon l'ancienneté des mesures. Le calcul de la carte EWMA est effectué grâce à l'équation (3.20) :

$$Y_i = (1 - \lambda) \cdot Y_{i-1} + \lambda \cdot \bar{X}_i \quad (3.20)$$

où  $i$  est le numéro de l'échantillon et  $\lambda$  est une constante de lissage telle que :  $0 < \lambda \leq 1$ . Le rôle du paramètre  $\lambda$  est de donner plus au moins de poids aux nouvelles mesures par rapport à la valeur moyenne (généralement une valeur de l'ordre de 0.5 est choisie).  $Y_0$  est la valeur initiale qui est égale à la moyenne cible ou à la moyenne des données initiales dans le cas où les informations sur la moyenne cible ne sont pas disponibles. Les observations  $X_i$  enregistrées séquentiellement peuvent être des valeurs observées individuellement à partir du processus, bien qu'il s'agisse souvent de moyennes d'échantillons obtenues à partir d'un plan d'échantillonnage.

Pour une carte de contrôle EWMA, il est nécessaire de définir des limites de contrôle :

$$S_{inf} = \mu_0 - K \cdot \sigma_0 \quad (3.21)$$

$$S_{sup} = \mu_0 + K \cdot \sigma_0 \quad (3.22)$$

avec  $K$  est une constante strictement positive ( $K > 0$ )

Une carte EWMA considère que le procédé est sous contrôle à l'instant  $i$  si  $Y_i \in ]S_{inf}, S_{sup}[$ . Si on remplace dans l'équation (3.20) le terme  $Y_{i-1}$  en fonction de  $Y_{i-2}$ , puis  $Y_{i-2}$  en fonction de  $Y_{i-3}$ , ainsi de suite, on obtient (3.23) :

$$Y_i = (1 - \lambda)^i \cdot Y_0 + \lambda \sum_{j=0}^{i-1} (1 - \lambda)^j \cdot \bar{X}_{i-j} \quad (3.23)$$

On voit ainsi clairement que  $Y_i$  est une combinaison linéaire de la variable aléatoire initiale  $Y_0$  pondérée par un coefficient  $(1 - \lambda)^i$ , et des variables aléatoires  $\bar{X}_1, \dots, \bar{X}_i$  pondérées par les coefficients  $\lambda (1 - \lambda)^{i-1}, \lambda (1 - \lambda)^{i-2}, \dots, \lambda (1 - \lambda)^0$ . C'est pour cette raison que la série  $Y_1, \dots, Y_i$  est appelée une série de moyenne mobile pondérée exponentiellement (EWMA).

On remarquera que plus  $\lambda$  est proche de 0, plus la série  $Y_1, \dots, Y_i$  est une version « lissée » de la série  $\bar{X}_1, \dots, \bar{X}_i$ . A la limite, pour  $\lambda = 0$ , on a  $Y_i = Y_{i-1} = \dots = Y_0$ . A l'inverse, plus  $\lambda$  est proche de 1, plus la série  $Y_1, \dots, Y_i$  ressemble à la série  $\bar{X}_1, \dots, \bar{X}_i$ . A la limite, pour  $\lambda = 1$ , on a  $Y_i = \bar{X}_i, i \geq 1$ .

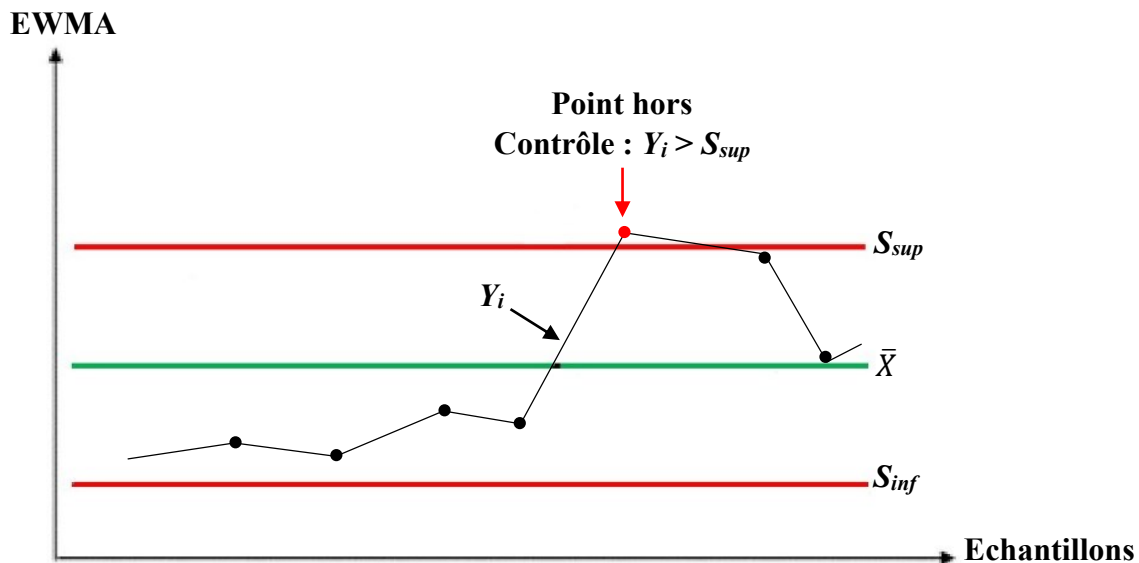


Figure 3.3. Carte de contrôle EWMA

### 3.3.4. Carte de contrôle à Moyenne Mobile (MM)

Le filtrage opéré par les cartes EWMA limite l'influence des valeurs extrêmes, mais a pour inconvénient de masquer une augmentation brusque ou retarder la détection par rapport à une carte de Shewhart dans le cas d'une dérive de forte amplitude. Pour pallier les inconvénients de cette carte tout en détectant rapidement les dérives de petites et moyennes amplitudes d'un processus, la carte de contrôle MM moins complexe que la carte EWMA a été employée dans notre travail.

De manière similaire à la carte EWMA, la carte de contrôle MM, non pondérée exponentiellement, permet de détecter les variations de la moyenne des séries de données. Les seuils de détection supérieur et inférieur  $S_{sup}$  et  $S_{inf}$  sont définis en fonction des tolérances souhaitées du système.

Dans ce travail, on s'intéresse à l'application de la carte de contrôle MM sur les durées de tir des transitions dans un chemin. La moyenne  $m_n$  et l'écart type  $\sigma_n$  des durées de tirs d'une transition donnée  $T$  sont calculés sur une fenêtre temporelle glissante de  $N$  valeurs consécutives. La surveillance par la carte de contrôle MM

début dès que le nombre d'acquisitions, supposées correspondre à un comportement sain, atteint la dimension de la fenêtre de scrutation  $N$ . Pour  $n > N$ , le calcul peut se simplifier en remplaçant le terme le plus ancien par le plus récent grâce à l'équation (3.24) :

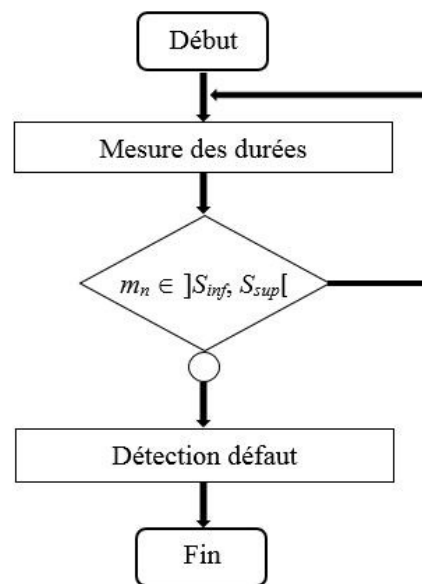
$$m_n = m_{n-1} + \frac{(d_n - d_{(n-N)})}{N} \quad \text{si } n > N \quad (3.24)$$

où  $d_n$  est la mesure la plus récente et  $d_{n-N}$  la plus ancienne.

Pour un système temporel surveillé par la carte de contrôle MM, on considère que le système possède un comportement sain dans le cas où  $m_n \in ]S_{inf}, S_{sup}[$ , dans le cas contraire, il a un comportement défectueux (figure 3.4). Les seuils de détection sont définis par l'équation (3.25) :

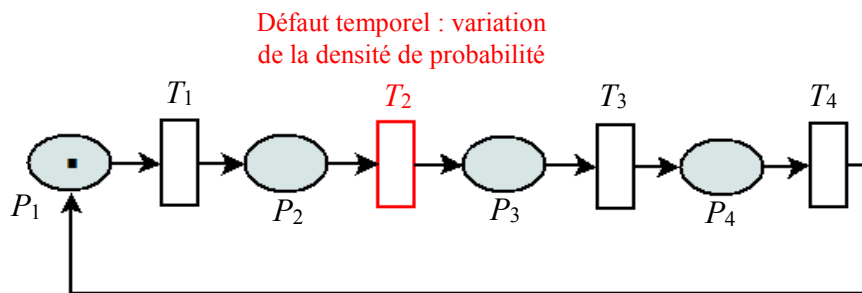
$$\begin{aligned} S_{sup} &= m_c + k \times \sigma_c \\ S_{inf} &= m_c - k \times \sigma_c \end{aligned} \quad (3.25)$$

où  $m_c$  et  $\sigma_c$  sont la moyenne et l'écart type des durées de tirs de la transition  $T$  dans le cas d'un comportement nominal et  $k$  est un paramètre d'entrée permettant d'ajuster les seuils de détection.



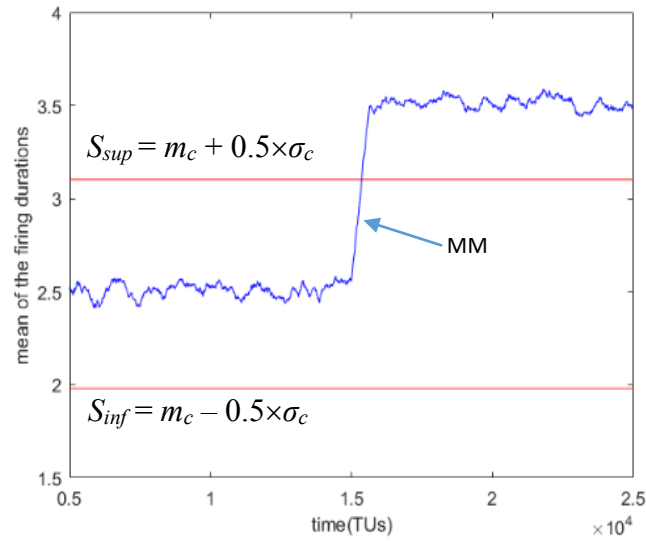
**Figure 3.4.** Utilisation d'une carte de contrôle MM

**Exemple 3.1.** On considère l'exemple de RdPSTPO représenté sur la figure 3.5 avec le marquage initial donné par  $M_I = (1 \ 0 \ 0 \ 0)^T$ . Toutes les transitions sont mesurées. Afin d'illustrer notre approche, des densités de probabilité uniformes bornées et triangulaires symétriques sont successivement considérées pour toutes les transitions. Les supports sont définis par  $a_{c1} = 1, b_{c1} = 2$  pour  $T_1$  ;  $a_{c2} = 2, b_{c2} = 3$  pour  $T_2$  ;  $a_{c3} = 4, b_{c3} = 5$  pour  $T_3$  ;  $a_{c4} = 0, b_{c4} = 2$  pour  $T_4$ . Si les densités de probabilité sont uniformes bornées (respectivement triangulaires symétriques), le support  $[a_2, b_2]$  de la transition  $T_2$  est calculé grâce à la proposition 3.1 (respectivement proposition 3.2). Dans les simulations suivantes, on suppose que la transition  $T_2$  subit un changement de support de  $[2, 3]$  à  $[3, 4]$  entre les instants 5000 UT et 25000 UT. Cette variation correspond à un saut de  $m$  de 2.5 à 3.5 UT.

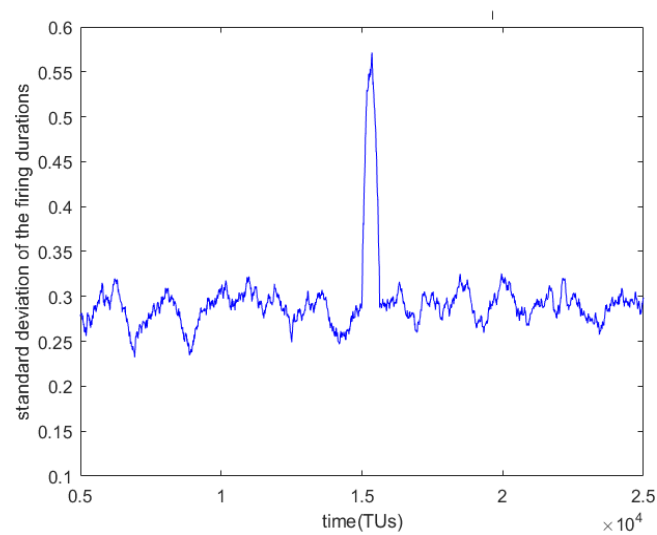


**Figure 3.5.** Exemple de RdPSTPO

L'application de la carte de contrôle MM sur la transition  $T_2$  nous permet de détecter les défauts temporels. Par exemple, sur la figure 3.6 (densités de probabilité uniforme bornée), un saut de 1 UT est observé. Cette variation, correspondant au saut de moyenne de 2.5 à 3.5 UT, traduit le passage d'un comportement sain, c'est-à-dire avant dépassement du seuil supérieur  $S_{sup}$  de la carte MM, vers un comportement défaut. Grâce à la proposition 3.1, la variation des extrémités du support de la loi peut être évaluée. Dans le cas de densités de probabilité triangulaires symétriques, des résultats similaires sont obtenus à partir de la proposition 3.2.



**Figure 3.6.** La variation de la moyenne des durées de franchissement de  $T_2$  avec une densité de probabilité uniforme bornée



**Figure 3.7.** La variation de l'écart-type des durées de franchissement de  $T_2$  avec une densité de probabilité uniforme bornée



### 3.4. Table de signatures et diagnostic

Dans le but de localiser les défauts liés aux variations de la moyenne des durées de franchissement des transitions, nous avons conçu une table de signatures. L'idée est de détecter puis de localiser les transitions défauts à partir des durées de certains chemins. La détection est effectuée à l'aide de la proposition 3.3, la localisation du défaut est basée sur la proposition 3.4.

#### 3.4.1. Conception de la table de signatures

Pour chaque chemin semi-silencieux, (chemin pour lesquels seules les dates des transitions extrêmes sont mesurées), on relève les transitions potentiellement responsables d'une augmentation de la durée du chemin concerné (transitions qui appartiennent à ce chemin).

La table de signatures contient la liste des transitions en colonne et l'ensemble des chemins semi-silencieux concernés en ligne. Chaque chemin semi-silencieux est de la forme :  $Path_s(T(1), T(h)) = T(1) P(1) T(2) P(2) \dots T(i) P(i) \dots T(h-1) P(h-1) T(h)$ .  $Tabsig(Path_s, T) = 1$  si une modification dans la durée de franchissement de la transition  $T$  provoque une variation de la durée  $d(T(1), T(h))$  du chemin  $Path_s(T(1), T(h))$ ,  $Tabsig(Path_s, T) = 0$  sinon. La construction de  $Tabsig$  est obtenue grâce à l'algorithme 3.1 :

#### Algorithme 3.1. Construction de la table de signatures $Tabsig$

**Entrées :** matrice d'incidence  $W$ , l'ensemble des chemins semi-silencieux  $\{Path_s(T(1), T(h))\}$

**Sortie :**  $Tabsig$  : la table de signatures

**Initialisation :**  $Tabsig \leftarrow [ ]$

1. **pour** tous les chemins semi-silencieux :  $Path_s(T(1), T(h))$
2.     **Pour** toutes les transitions silencieuses  $T(j), j \in \{2, \dots, h\}$  incluses dans  $Path_s(T(1), T(h))$
3.              $Tabsig(Path_s(T(1), T(h)), T(j)) \leftarrow 1$
4.     **fin pour**
5. **fin pour**

### 3.4.2. Détection et localisation des défauts temporels

Considérons maintenant une trajectoire mesurée donnée par :  $\Gamma(\sigma, M_I) = (e_{obs\ 1}, \tau_1) \dots (e_{obs\ K}, \tau_K)$ .  $\Gamma(\sigma, M_I)$  est suffisamment grande pour que les transitions issues de chaque chemin semi-silencieux apparaissent plusieurs fois. Selon l'hypothèse considérée, chaque sous-séquence élémentaire de  $\Gamma(\sigma, M_I) : (e_{obs\ K}, \tau_K) \dots (e_{obs\ K+1}, \tau_{K+1})$ ,  $k = 1, \dots, K-1$  correspond à un unique chemin semi-silencieux, à savoir  $Path_s$ . La moyenne des durées des  $N$  tirs consécutifs de  $Path_s$  est calculée à l'aide de la carte de contrôle MM. Si une variation est détectée, cette variation est certainement due à un changement de support de l'une des transitions incluses dans  $Path_s$ . L'ensemble des transitions défauts possibles est finalement obtenu en corrélant les différents chemins semi-silencieux dont la moyenne des durées varie de manière significative.

La proposition 3.3 est utilisée pour détecter la présence d'une transition défaut dans un chemin semi-silencieux donné. Ensuite, en fonction des mesures collectées, la proposition 3.4 permet de localiser la transition défaut.

**Proposition 3.3.** On considère un RdPSTPO,  $T(1)$  et  $T(h)$  sont deux transitions non silencieuses. En supposant l'existence d'un seul chemin semi-silencieux entre  $T(1)$  et  $T(h) : Path_s(T(1), T(h)) = T(1) P(1) T(2) P(2) \dots T(i) P(i) \dots T(h-1) P(h-1) T(h)$ , alors  $d(T(1), T(h)) = \alpha(T(h)) - \alpha(T(1))$  ne dépend que des durées de franchissement de l'ensemble des transitions  $\{T(2) \dots T(h)\}$ .

**Démonstration.** Soit  $DT(T(h))$  la durée de franchissement de la transition  $T(h)$ . Sous l'hypothèse qu'une seule marque circule dans le réseau, en particulier dans le chemin  $Path_s(T(1), T(h))$ ,  $DT(T(h))$  peut s'écrire comme suit :

$$DT(T(h)) = \alpha(T(h)) - \alpha(T(h-1)) \quad (3.26)$$

puis :

$$DT(T(h)) = \alpha(T(h)) - (DT(T(h-1)) + \alpha(T(h-2))) \quad (3.27)$$

où  $DT(T(h-1)) = \alpha(T(h-1)) - \alpha(T(h-2))$ . D'une manière similaire :

$$DT(T(h)) = \alpha(T(h)) - DT(T(h-1)) - (DT(T(h-2)) + \alpha(T(h-3))) \quad (3.28)$$

et

$$DT(T(h)) = \alpha(T(h)) - DT(T(h-1)) - DT(T(h-2)) - \dots - DT(T(2)) - \alpha(T(1)) \quad (3.29)$$

alors

$$\tau(T(h)) - \tau(T(1)) = DT(T(2)) + \dots + DT(T(h-1)) + DT(T(h)) \quad (3.30)$$

Ainsi,  $d(T(1), T(h))$  ne dépend que de la durée des  $h-1$  dernières transitions du chemin  $Path_s(T(1), T(h))$ .

**Proposition 3.4.** On considère un RdpSTPO avec sa table de signatures. Soit la trajectoire mesurée  $\Gamma(\sigma, M_I) = (e_{obs\ 1}, \tau_1) \dots (e_{obs\ K}, \tau_K)$ . Soit  $P_{défaut}$  l'ensemble des chemins semi-silencieux pour lesquels la moyenne des durées mesurées contient une variation significative :  $P_{défaut} = \{ Path_s(T(1), T(h)) / d(T(1), T(h)) \notin ]S_{inf}, S_{sup}[ \}$ . Soit  $T_{défaut}$  l'ensemble des transitions défauts possibles.  $T_{défaut}$  satisfait :

$$T_{défaut} = (\cap \{ Path_s(T(2), T(h)) \}) \text{ pour tous les } Path_s(T(1), T(h)) \in P_{défaut} \quad (3.31)$$

**Démonstration.** Supposons que  $T$  soit la transition défaut, la durée de franchissement de cette transition présente donc une variation. Ainsi, tous les chemins semi-silencieux contenant  $T$  doivent présenter une variation de leurs durées selon la proposition 3.3, et tout chemin semi-silencieux ne contenant pas  $T$  ne doit pas avoir de variation : alors,  $T \in Path_s(T(2), T(h))$  si et seulement si  $Path_s(T(1), T(h)) \in P_{défaut}$ .

Deux cas peuvent se produire :

- a)  $T_{défaut}$  est un singleton et la localisation est complète
- b)  $T_{défaut}$  contient plusieurs transitions défauts et la localisation est partielle

### 3.5. Exemple

On considère le RdpSTPO marqué représenté sur la figure 3.8, et doté du marquage initial  $M_I = (1\ 0\ 0\ 0\ 0)^T$ . Seules les transitions  $T_1$ ,  $T_3$  et  $T_4$  sont mesurées. Aucune place n'est mesurée (les places et transitions non mesurées sont grisées). Des densités de probabilité uniformes bornées sont considérées pour toutes les transitions.

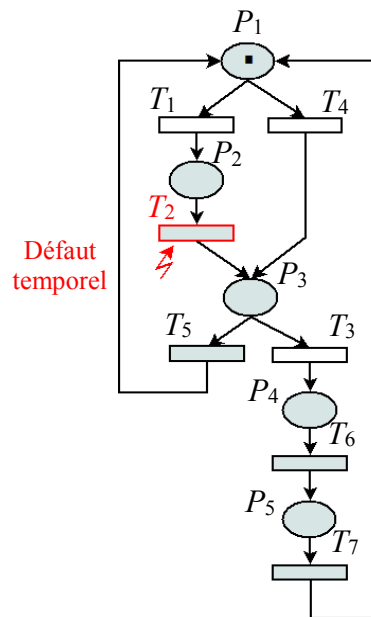
Les supports des lois modélisant les durées de franchissement des transitions sont définis comme suit :  $a_{c1} = 1$ ,  $b_{c1} = 3$  pour  $T_1$  ;  $a_{c2} = 2$ ,  $b_{c2} = 4$  pour  $T_2$  ;  $a_{c3} = 0$ ,  $b_{c3} = 3$  pour  $T_3$  ;  $a_{c4} = 2$ ,  $b_{c4} = 4$  pour  $T_4$  ;  $a_{c5} = 2$ ,  $b_{c5} = 4$

pour  $T_5$  ;  $a_{c6} = 3$ ,  $b_{c6} = 4$  pour  $T_6$  ;  $a_{c7} = 1$ ,  $b_{c7} = 5$  pour  $T_7$ . La transition  $T_2$  subit une variation de son support de loi de  $[2, 4]$  à  $[3, 5]$  entre 5000 UT à 27000 UT.

L'ensemble  $P_{mes}$  des chemins semi-silencieux mesurés est :

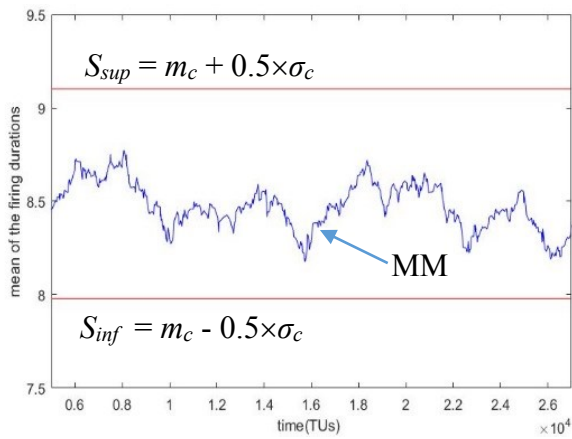
$$P_{mes} = \{Path_s(T_3, T_1), Path_s(T_3, T_4), Path_s(T_1, T_3), Path_s(T_1, T_4), Path_s(T_1, T_1), Path_s(T_4, T_3), Path_s(T_4, T_1), Path_s(T_4, T_4)\}.$$

avec :  $Path_s(T_3, T_1) = T_3 T_6 T_7 T_1$ ,  $Path_s(T_3, T_4) = T_3 T_6 T_7 T_4$ ,  $Path_s(T_1, T_3) = T_1 T_2 T_3$ ,  $Path_s(T_1, T_4) = T_1 T_2 T_5 T_4$ ,  $Path_s(T_1, T_1) = T_1 T_2 T_5 T_1$ ,  $Path_s(T_4, T_3) = T_4 T_3$ ,  $Path_s(T_4, T_1) = T_4 T_5 T_1$ ,  $Path_s(T_4, T_4) = T_4 T_5 T_4$ .

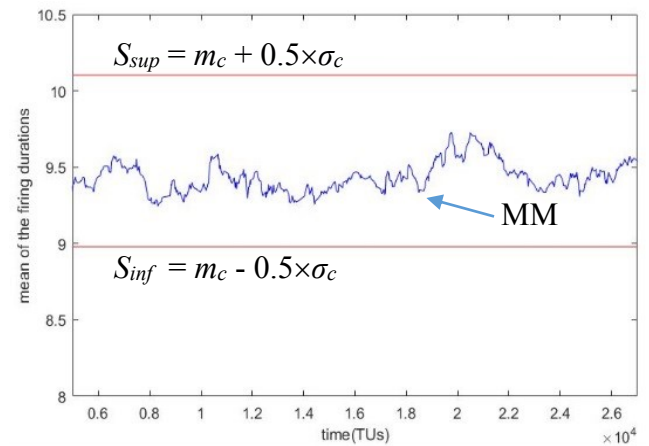


**Figure 3.8.** Exemple de RdPSTPO

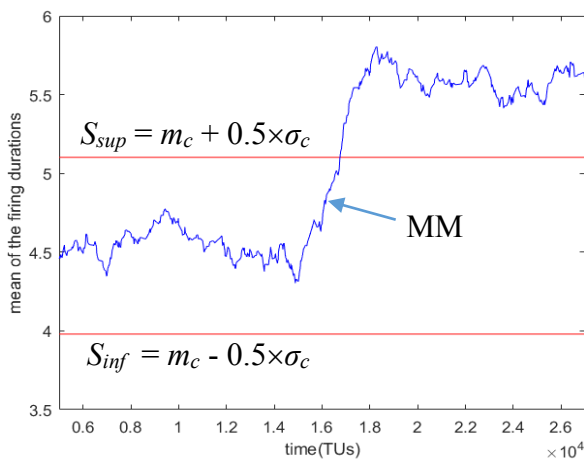
Les différentes durées  $d(T_k, T_{k'})$  liées aux chemins semi-silencieux  $Path_s(T_k, T_{k'})$ , sont enregistrées. Des cartes de contrôle MM sont appliquées sur chacune de ces durées des chemins de  $P_{mes}$ , figure 3.9 a) à 3.9 h).



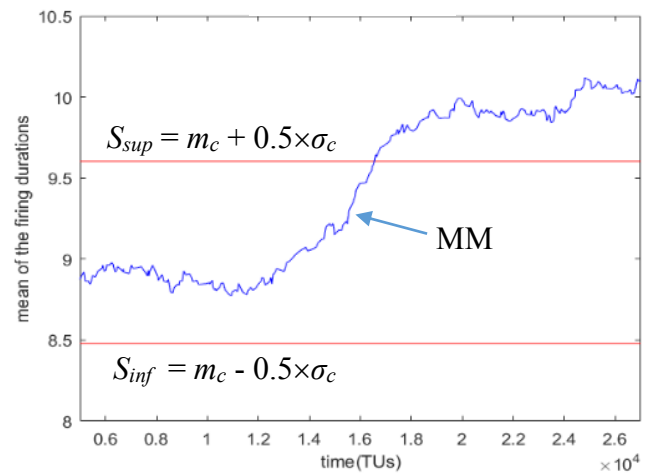
**Figure 3.9 a).** La carte de contrôle MM appliquée sur :  $Path_s(T_3, T_1)$



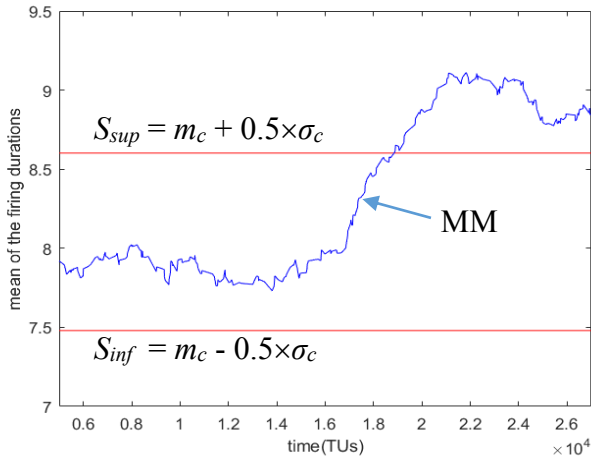
**Figure 3.9 b).** La carte de contrôle MM appliquée sur :  $Path_s(T_3, T_4)$



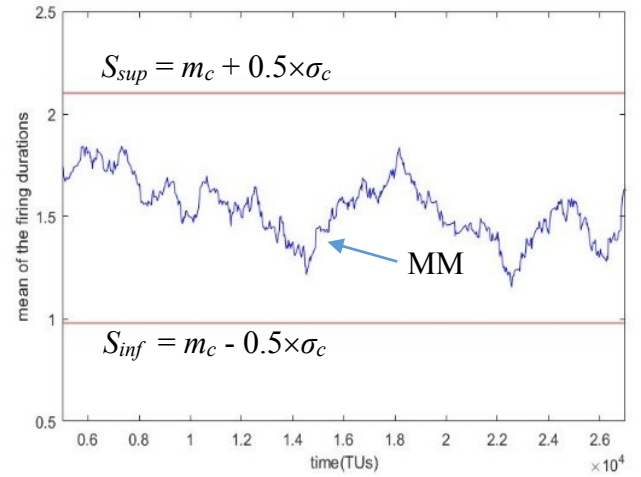
**Figure 3.9 c).** La carte de contrôle MM appliquée sur :  $Path_s(T_1, T_3)$



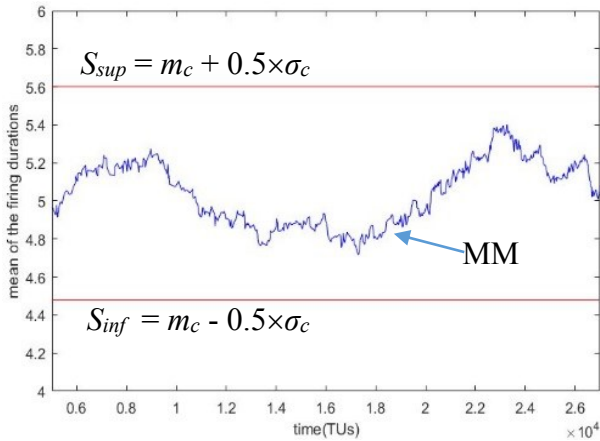
**Figure 3.9 d).** La carte de contrôle MM appliquée sur :  $Path_s(T_1, T_4)$



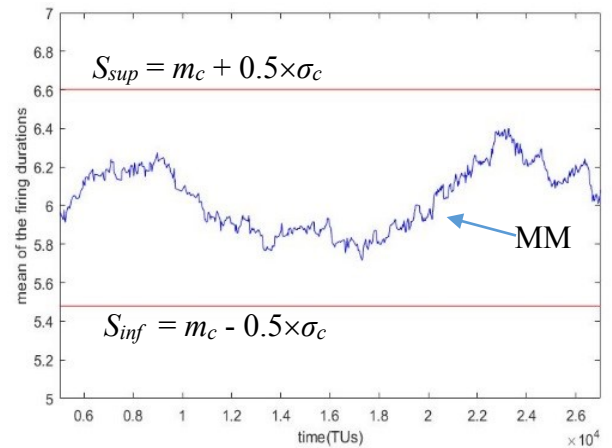
**Figure 3.9 e).** La carte de contrôle MM appliquée sur :  $Path_s(T_1, T_1)$



**Figure 3.9 f).** La carte de contrôle MM appliquée sur :  $Path_s(T_4, T_3)$



**Figure 3.9 g).** La carte de contrôle MM appliquée sur :  $Path_s(T_4, T_1)$



**Figure 3.9 h).** La carte de contrôle MM appliquée sur :  $Path_s(T_4, T_4)$

On constate une variation de la moyenne des trois durées :  $d(T_1, T_3)$ ,  $d(T_1, T_4)$  et  $d(T_1, T_1)$  dans l'intervalle de temps [5000, 27000]. Cette variation est détectée grâce au seuil supérieur placé sur les cartes de contrôle MM.

La table de signatures correspondant à l'exemple de la figure 3.8 est détaillée dans le tableau 3.3.

**Tableau 3.3.** La table de signatures de l'exemple de la figure 3.8

	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$
$Path_s(T_3, T_1)$	1	0	0	0	0	1	1
$Path_s(T_3, T_4)$	0	0	0	1	0	1	1
$Path_s(T_1, T_3)$	0	1	1	0	0	0	0
$Path_s(T_1, T_4)$	0	1	0	1	1	0	0
$Path_s(T_1, T_1)$	1	1	0	0	1	0	0
$Path_s(T_4, T_3)$	0	0	1	0	0	0	0
$Path_s(T_4, T_1)$	1	0	0	0	1	0	0
$Path_s(T_4, T_4)$	0	0	0	1	1	0	0

Les trois lignes surlignées correspondent aux chemins de l'ensemble  $P_{\text{défaut}}$ . La transition défaut est localisée grâce à la proposition 3.4 :

$$\{T_2\} = Path_s(T_1, T_3) \cap Path_s(T_1, T_4) \cap Path_s(T_1, T_1)$$

### 3.6. Conclusion

Dans ce chapitre, nous avons introduit les bases du SPC et présenté les principales cartes de contrôle pour la moyenne, l'étendue, l'écart-type et la médiane. Nous avons proposé une approche pour la détection et la localisation des défauts temporels dans les SED avec des densités de probabilité arbitraires. Dans notre travail, les défauts temporels correspondent à une variation de la moyenne des durées de franchissement des transitions dans des RdPSTPO, qui sont détectés par l'application d'une carte de contrôle MM. La carte MM a été choisie parmi les différentes cartes introduites dans ce chapitre pour sa simplicité et sa rapidité de détection pour des dérives de faibles et moyennes amplitudes. La localisation de la transition présentant un défaut temporel, parmi plusieurs transitions candidates, a été réalisée grâce à une table de signatures.

---

## Conclusion Générale et Perspectives

---

Les travaux que nous avons réalisés ont porté sur l'étude et le développement de méthodes de surveillance des systèmes à événements discrets avec contraintes temporelles. Dans le but d'analyser et de surveiller ces systèmes, nous nous sommes focalisés sur l'importance de l'information temporelle pour une caractérisation précise de l'état du système. Nous avons utilisé le modèle des Réseaux de Petri Stochastiques Temporisés Partiellement Observés permettant de prendre en compte les aspects temporels ainsi que les comportements aléatoires du système dus à l'occurrence de défaillances dès l'étape de modélisation. De plus, le modèle considéré permet une représentation de l'instrumentation du système. Nous avons privilégié des approches permettant d'apporter des mesures correctives en cas de défaut afin d'éviter des conséquences qui peuvent devenir dangereuses, conduire à l'indisponibilité du système et provoquer des dégâts humains et matériels importants.

La première étape a été d'introduire les contraintes temporelles dans le modèle RdP. Des contraintes de temps complexes entre les tâches ont été extraites de l'outil de modélisation de processus métier et reportées dans le RdP grâce à des règles de transformation que nous avons élaborées.

Une première approche de détection et de diagnostic a été proposée pour tirer profit des informations temporelles disponibles dans le système (les dates d'acquisitions des mesures), et discerner avec précision les différents comportements qui expliquent les mesures afin de diagnostiquer les défauts. En se basant sur cette exploitation des informations temporelles et sur les dates d'occurrence des événements, nous avons calculé la probabilité d'occurrence des défauts (temporels et structurels). Nous avons évalué la probabilité de violation des contraintes temporelles pour les trames mesurées. Par conséquent, nous avons pu renforcer le diagnostic des systèmes et discerner les différents comportements qui coïncident avec les mesures.



Une seconde approche de diagnostic a ensuite été proposée avec l'utilisation d'une carte de contrôle de type Moyenne Mobile pour détecter les dérives lentes des moyennes des durées de franchissement des transitions. On a pu constater que de faibles variations non détectables ponctuellement peuvent être repérées grâce à cette carte de contrôle MM. De plus, l'implémentation de cette méthode est rapide et facile : les équations mathématiques et les calculs sont simples en comparaison des méthodes présentées au chapitre 2. Finalement, une table de signatures a été proposée pour localiser la transition en défaut temporel parmi plusieurs transitions candidates.

Au terme de ces travaux de thèse, plusieurs perspectives de recherche se dégagent visant à améliorer et étendre les résultats obtenus :

➤ A court terme :

- Combiner des défauts structurels et temporels dans le même modèle tout en considérant des lois de probabilité continues ou discrètes très générales.
- Alléger les différentes hypothèses de travail employées dans le diagnostic des défauts temporels par utilisation d'une carte de contrôle MM, notamment :
  1. le cas où plusieurs marques circulent dans le réseau
  2. l'apparition de défauts caractérisés par un changement dans le type de la distribution des durées
  3. l'occurrence de plusieurs défauts dans une trajectoire de mesures
- Évaluer la probabilité de violation des contraintes temporelles futures pour des applications de pronostic.

➤ A plus long terme on pourra envisager d'étendre notre approche basée sur les RdP Stochastiques Partiellement Observés à un modèle de RdP Stochastiques Généralisés (RdPSG) [Marsan et al. 1994] incluant des transitions immédiates.

## Bibliographie

- Ali, S., Pievatolo, A. et Gob, R., 2016. An Overview of Control Charts for High-quality Processes. *Quality and reliability engineering international*.
- Alur, R. et Dill, D.L., 1994. A theory of timed automata. *Theoretical computer science*, 126(2), pp.183–235.
- Ammour, R., 2017. *Contribution au Diagnostic et Pronostic des Systèmes à Événements Discrets Temporisés par des Réseaux de Petri Stochastiques*. Ph.D. Thesis, Université du Havre.
- Ammour, R., Leclercq, E., Sanlaville, E. et Lefebvre, D., 2015. Estimation of the fault occurrence dates in DESs with partially observed stochastic Petri nets. *5th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS)*. Atlanta, USA. pp.254–259.
- Ammour, R., Leclercq, E., Sanlaville, E. et Lefebvre, D., 2016. Faults prognosis using partially observed stochastic Petri nets. In *13th International Workshop on Discrete Event Systems, (WODES)*. IEEE. Xian, China. pp.472–477.
- Ammour, R., Leclercq, E., Sanlaville, E. et Lefebvre, D., 2017a. Fault prognosis of timed stochastic discrete event systems with bounded estimation error. *Automatica*, 82, pp.35–41.
- Ammour, R., Leclercq, E., Sanlaville, E. et Lefebvre, D., 2017b. Faults prognosis using partially observed stochastic Petri-nets : an incremental approach. *Discrete Event Dynamic Systems*, pp.1–21.
- Annadi, H.P., Keats, J.B., Runger, G.C. et Montgomery, D.C., 1995. An Adaptive Sample Size CUSUM Control Chart. *International Journal of Production Research*, 33(6), pp.1605–1616.
- Barilla J., Lokajicek, M., Pisakova, H. et Simr, P., 1963. Applying Petri nets to modeling the chemical stage of radiobiological mechanism. *Journal of Physics and Chemistry of Solids*, 78, pp.127–136.
- Basile, F., Cabasino, M.P. et Seatzu, C., 2013. Marking estimation of Time Petri nets with unobservable transitions. In *18th IEEE Int. Conf. on Emerging Technologies & Factory Automation*. Cagliari, Italy.
- Basile, F., Cabasino, M.P. et Seatzu, C., 2014. K-diagnosability of Time labeled Petri nets. *12<sup>th</sup> IFAC-IEEE Workshop on Discrete Event Systems*, pp.14–16. Cachan, France.
- Basile, F., Cabasino, M.P. et Seatzu, C., 2015. State estimation and fault diagnosis of Time labeled Petri net systems with unobservable transitions. *IEEE Transactions on Automatic Control*. 60(4), pp.997–1009.
- Basile, F., Chiacchio, P. et De Tommasi, G., 2008. Sufficient conditions for diagnosability of Petri nets. *the 9th International Workshop on Discrete Event Systems*, pp.370–375. Göteborg, Sweden.

- Basile, F., Chiacchio, P. et De Tommasi, G., 2009. An efficient approach for online diagnosis of discrete event systems. *IEEE Transactions on Automatic Control*, 54(4), pp.748–759.
- Basile, F., Chiacchio, P. et De Tommasi, G., 2012. On K-diagnosability of Petri nets via integer linear programming. *Automatica*, 48(9), pp.2047–2058.
- Bause, F. et Kritzinger, P.S., 2002. *Stochastic Petri Nets*: Verlag, Verlag Vieweg, Wiesbaden, 26.
- Benneyan, J.C., 2001. *Design, Use, and Performance of Statistical Control Charts for Clinical Process Improvement*. International journal of six sigma, vol 4.
- Berthomieu, B. et Diaz, M., 1991. Modeling and verification of time dependent systems using time Petri nets. *IEEE transactions on software engineering*, 17(3), pp.259–273.
- Berthomieu, B. et Menasche, M., 1983 *An Enumerative Approach For Analyzing Time Petri Nets*. In Proceedings of IFIP, pp.41–46. Elsevier Science Publishers. Rio de Janeiro, Brazil.
- Berthomieu, B. et Vernadat, F., 2003. State Class Contructions for Branching Analysis of Time Petri nets. Springer-Verlag, pp.442–457.
- Billington, J. et Diaz, M., 1999. *Application of Petri nets to communication networks: Advances in Petri nets*, Springer Science & Business Media.
- Bobbio, A., 1990. System Modelling with Petri nets. *Systems reliability assessment: proceedings of the Ispra courses (On Reliability and Risk Analysis)*, vol 6, pp.103–143. Springer, Dordrecht.
- Bobbio, A., Puliafito, A., Telek, M. et Trivedi, K. S. 1998. Recent developments in non-markovian stochastic Petri nets. *Journal of Circuits, Systems, and Computers*, 8(1), pp.119–158.
- Boel, R. et Jiroveanu, G., 2013. The on-line diagnosis of time Petri nets. *Lecture Notes in Control and Information Sciences*, vol 433. Springer, London.
- Boussif, A., 2016. *Contributions to Model-Based Diagnosis of Discrete-Event Systems*. Ph.D. Thesis, Université de Lille.
- Boussif, A., Ghazel, M. et Klai, K., 2015. Combining enumerative and symbolic techniques for diagnosis of discrete event systems. *9th International Workshop on Verification and Evaluation of Computer and Communication Systems*, pp.47–119. Bucarest, Romania.
- Boussif, A., Liu, B. et Ghazel, M., 2017. An Experimental Comparison of Three Diagnosis Techniques for Discrete Event Systems. In *DX'17, 28th International Workshop on Principles of Diagnosis*. Brescia, Italy.
- Bouyer, P., Chevalier, F. et D'Souza, D., 2005. Fault diagnosis using timed automata. In *International Conference on Foundations of Software Science and Computation Structures*. Edinburgh, UK. Springer, pp.219–233.

- Box, G. E. et Luceno, A., 1997. *Statistical Control by Monitoring and Feedback Adjustment*. John Wiley & Sons, Incorporated.
- Burks, A.W. et Wang, H., 1957. The Logic of Automata - Part I. *Journal of the ACM (JACM)*, 4(2), pp.193–218.
- Cabasino, M. P., Giua, A., Lafortune, S. et Seatzu, C., 2009b. Diagnosability analysis of unbounded Petri nets. In *Proceedings of the IEEE Conference on Decision and Control*. Shanghai, China, pp.1267–1272.
- Cabasino, M.P., Giua, A. et Seatzu, C., 2009a. Diagnosability of bounded Petri nets. In *Proceedings of the IEEE Conference on Decision and Control*. Shanghai, China, pp.1254–1260.
- Cabasino, M.P., Giua, A. et Seatzu, C., 2010. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9), pp.1531–1539.
- Cabasino, M.P., Giua, A. et Seatzu, C., 2014. Diagnosability of discrete-event systems using labeled Petri nets. *IEEE Transactions on Automation Science and Engineering*, 11(1), pp.144–153.
- Campos, J., Chiola, G., Colom, J.M. et Silva, M., 1989. Tight Polynomial Bounds for Steady-State Performance of Marked Graphs. *Workshop on Petri Nets and Performance Models PNPM*, Kyoto, Japan.
- Cassandras, C.G. et Lafortune, S., 2009. *Introduction to discrete event systems*, Springer Science & Business Media.
- Chen, Y.-L. et Provan, G., 1997. Modeling and diagnosis of timed discrete event systems-a factory automation example. In *American Control Conference. Proceedings of the 1997*. New Mexico, USA, pp.31–36.
- Cong, X., Fanti, M. P., Mangini, A. M. et Li, Z., 2017., Decentralized Fault Diagnosis by Petri Nets and Integer Linear Programming. In *20th IFAC World Congress*. Toulouse, France, pp.14188–14193.
- Diaz, M., 2013. *Petri nets: fundamental models, verification and applications*, John Wiley & Sons.
- DiCesare, F., Harhalakis, G., Proth, J. M., Silva, M. et Vernadat, F. B., 1993. *Practice of Petri nets in manufacturing*, Springer, Netherlands.
- Distefano, S., Longo, F. et Scarpa, M., 2017. Marking dependency in non-Markovian stochastic Petri nets. *Performance Evaluation*, 110, pp.22–47.
- Dotoli, M., Fanti, M. P., Mangini, A. M. et Ukovich, W., 2009. On-line fault detection in discrete event systems by Petri nets and integer linear programming. *Automatica*, 45(11), pp.2665–2672.
- Frutos Escrig, D., Valero, R. V. et Marroquin, A.O., 2000. *Decidability of Properties of Timed-Arc Petri Nets*. In *Proceedings of ICATPN*, Springer-Verlag, pp.187–206.

- German, R., 2000. *Performance analysis of communication systems with non-Markovian stochastic Petri nets*, John Wiley & Sons, Inc.
- Ghazel, M., 2005. *Surveillance des systèmes à événements discrets à l'aide des réseaux de Petri Temporels*. Thèse de doctorat, Ecole Centrale de Lille.
- Girault, C. et Valk, R., 2013. *Petri nets for systems engineering: a guide to modeling, verification, and applications*, Springer Science & Business Media.
- Giua A. et Seatzu, C., 2007. Marking estimation of Petri nets with silent transitions. *IEEE Transactions on Automatic Control*, 52(9), pp.1695–1699.
- Giua, A. et Seatzu, C., 2005. Fault detection for discrete event systems using Petri nets with unobservable transitions. In *Decision and Control and European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. Seville, Spain, pp.6323–6328.
- Gougam, H.E., Subias, A. et Pencolé, Y., 2013. Supervision patterns: Formal diagnosability checking by petri net unfolding. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 4(PART 1), pp.73–78.
- Gougam, H.-E.E., Subias, A. et Pencolé, Y., 2014. Discriminability Analysis of Supervision Patterns by Net Unfoldings. *IFAC Proceedings Volumes*, 47(2), pp.459–464.
- Haddad, S., Mokdad, L. et Moreaux, P., 2006. *A new approach to the evaluation of non-markovian stochastic Petri nets*, Springer.
- Horváth, A., Paolieri, M., Ridi, L. et Vicario, E., 2012. Transient analysis of non-Markovian models using stochastic state classes. *Performance Evaluation*, 69(7), pp.315–335.
- Jéron, T., Marchand, H., Pinchinat, S. et Cordier, M.O., 2006. Supervision Patterns in Discrete Event Systems Diagnosis. *Workshop on Discrete Event Systems. IEEE Computer society*, pp.262–268.
- Jiang, S. et Kumar R., 2004. Failure diagnosis of discrete-event systems with linear-time temporal logic specifications. *IEEE Transactions on Automatic Control*, 49(6), pp.934–945.
- Jiang, S. et Kumar, R., 2006. Diagnosis of repeated failures for discrete event systems with linear-time temporal-logic specifications. *IEEE Transactions on Automation Science and Engineering*, 3(1), pp.47–59.
- Jiang, S., Huang, Z., Chandra, V. et Kumar, R., 2001. A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 46(8), pp.1318–1321.
- Jiroveanu, G. et Boel, R.K., 2010. The diagnosability of Petri net models using minimal explanations. *Automatic Control, IEEE Transactions on*, 55(7), pp.1663–1668.
- Jiroveanu, G., De Schutter, B. et Boel, R.K., 2006. On-line diagnosis for time Petri nets. In *Discrete Event Systems, in 8th International Workshop on*. IEEE, Michigan, USA. pp.313–318.

- 
- Kleene, S.C., 1951. *Representation of events in nerve nets and finite automata*, RAND PROJECT AIR FORCE SANTA MONICA CA.
- Kumar, A. et Barton, R.R., 2016. Controlled violation of temporal process constraints – Models, algorithms and results. *Information Systems*, Elsevier.
- Lefebvre, D. et Delherm, C., 2007. Diagnosis of DES with Petri net models. *IEEE Transactions on Automation Science and Engineering*, 4(1), pp.114–118.
- Lefebvre, D. et Leclercq, E., 2015. Diagnosability of Petri nets with observation graphs. *Discrete Event Dynamic Systems*. 26(3), pp.539–559.
- Lefebvre, D., 2012. Diagnosis with Petri nets according to partial events and states observation. In *Proc. IFAC Safeprocess '12*, Mexico City, Mexico.
- Lefebvre, D., 2014a. On-Line Fault Diagnosis With Partially Observed Petri Nets. *IEEE Transactions on Automatic Control*, 59(7), pp.1919–1924.
- Lefebvre, D., 2014b. Fault diagnosis and prognosis with partially observed petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(10), pp.1413–1424.
- Lefebvre, D., 2017a. Detection of Temporal Anomalies for Partially Observed Timed PNs. *Mathematical Problems in Engineering*, vol 7. DOI: 10.1155/2017/2821078.
- Lefebvre, D., 2017b. Probability of faults for partially observed Timed PNs with temporal constraints. In *Networking, Sensing and Control (ICNSC), 2017 IEEE 14<sup>th</sup> International Conference*. IEEE, pp.103–108. Calabria, Southern Italy.
- Lefebvre, D., Rachidi, S., Leclercq, E. et Pigné, Y., 2018a. Diagnosis of structural and temporal faults for k bounded non-markovian stochastic Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems SMCA*, DOI: 10.1109/TSMC.2018.2875726.
- Lefebvre, D., Rachidi, S., Leclercq, E. et Pigné, Y., 2018b. Temporal fault diagnosis for k-bounded non-markovian SPN. *5th International Conference on Control, Decision and Information Technologies CoDIT*. Thessaloniki, Greece.
- Li, B., Khelif-Bouassida, M. et Toguyéni, A., 2015a. On-the-fly diagnosability analysis of labeled Petri nets using T-invariants. *IFAC-PapersOnLine*, 48(7), pp.64–70.
- Li, B., Liu, B. et Toguyéni, A., 2015b. On-the-fly diagnosability analysis of labeled petri nets using minimal explanations. *IFAC-PapersOnLine*, 48(21), pp.326–331.
- Lin, F., 1994. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems*, 4(2), pp.197–212.
- Liu, B., 2014. *An Efficient Approach for Diagnosability and Diagnosis of DES Based on Labeled Petri Nets, Untimed and Timed Contexts*. Ph.D. Thesis, Université de Lille.

- 
- Liu, B., Ghazel, M. et Toguyeni, A., 2013. Evaluation à la volée de la diagnosticabilité des systèmes à événements discrets temporisés. *Journal Européen des Systèmes Automatisés*, 47(1–3), pp.227–242.
- Liu, B., Ghazel, M. et Toguyéni, A., 2014. Toward an efficient approach for diagnosability analysis of DES modeled by labeled Petri nets. *13th European Control Conference ECC'2014*. Strasbourg, France.
- Liu, B., Ghazel, M. et Toguyéni, A., 2017. On-the-Fly and Incremental Technique for Fault Diagnosis of Discrete Event Systems Modeled by Labeled Petri Nets. *Asian Journal of Control*.
- Lucas, J.M. et Saccuci, M.S., 1990. Exponentially Weighted Moving Average Control Schemes: Properties and Enhancement. *Technometrics*, vol 32, pp.1–29.
- Marsan, M. A., Balbo, G., Conte, G., Donatelli, S., et Franceschinis, G., 1994. *Modelling with generalized stochastic Petri nets*, John Wiley & Sons, Inc.
- Marsan, M.A., 1988. Stochastic Petri nets: an elementary introduction. In *European Workshop on Applications and Theory in Petri Nets*. Springer. Venice, Italy, pp.1–29.
- Merlin, P.M., 1975. *A study of the recoverability of computing systems*. Ph.D thesis, University of California.
- Molloy, M.K., 1982. Performance analysis using stochastic Petri nets. *IEEE Transactions on computers*, 31(9), pp.913–917.
- Montgomery, D.C., 1991. Introduction to Statistical Quality Control. *2nd Edition*. John Wiley & Sons, New York.
- Moore, E.F., 1956. Gedanken-experiments on sequential machines. *Automata studies*, vol 34, pp.129–153.
- Murata, T., 1989. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), pp.541–580.
- Pages, E.S., 1954. Continuous Inspection Schemes. *Biometrika*, vol 41, pp.100115.
- Petri, C.A., 1962. *Communicating with Automata*. Ph.D Thesis Hamburg.
- Prock, J., 1991. A new technique for fault detection using Petri nets. *Automatica*, 27(2), pp.239–245.
- Rabin, M.O., 1963. Probabilistic Automata. *Information and Control*, vol 6, pp.230–245.
- Rachidi, S., Leclercq, E., Pigné, Y. et Lefebvre, D., 2017. PN modeling of discrete event systems with temporal constraints. *21th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania.
- Rachidi, S., Leclercq, E., Pigné, Y. et Lefebvre, D., 2019. Timed diagnosis for manufacturing systems and logistics with markovian and non-markovian stochastic Petri nets. *PART I: Journal of Systems and Control Engineering*. Article soumis pour publication.

- Ramadge, P.J.G. et Wonham, W.M., 1989. The control of discrete event systems. *Proceedings of the IEEE*, 77(1), pp.81–98.
- Ramchandani, C., 1973. Analysis of asynchronous concurrent systems by timed petri nets. PhD thesis, Massachusetts Institute of Technology.
- Ramírez-Treviño, A., Ruiz-Beltrán, E., Rivera-Rangel, I. et Lopez-Mellado, E., 2007. Online fault diagnosis of discrete event systems. A Petri net-based approach. *IEEE Transactions on Automation Science and Engineering*, 4(1), pp.31–39.
- Reisig, W., 2013. *Elements of distributed algorithms: modeling and analysis with Petri nets*, Springer Science & Business Media.
- René, D. et Alla, H., 2010. *Discrete, Continuous, and Hybrid Petri nets*. Springer Science & Business Media.
- Roberts, S.W., 1959. Control chart tests based on geometric moving averages. *Technometrics*, vol 1, pp.239–250.
- Ross, S. M., 2014. *Introduction to probability models*, 11th Edition, Academic Press Elsevier.
- Ru, Y. et Hadjicostis, C.N., 2009. Fault Diagnosis in Discrete Event Systems Modeled by Partially Observed Petri Nets. *Discrete Event Dynamic Systems*, 19(4), pp.551–575.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. et Teneketzis, D., 1995. Diagnosability of discrete-event systems. *IEEE Transactions on automatic control*, 40(9), pp.1555–1575.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K. et Teneketzis, D. C., 1996. Failure diagnosis using discrete-event models. *IEEE transactions on control systems technology*, 4(2), pp.105–124.
- Saporta, G., 2006. Probabilités, analyse des données et statistique. *Editions TECHNIP*, pp.284–2006.
- Shallit, J.O., 2008. A Second Course in Formal Languages and Automata Theory. *Cambridge University Press*, ISBN 978-0-521-86572-2, 2008.
- Shewhart, W.A., 1931. *Economic Control of Quality of Manufactured Product*, New York: 1931. Reprinted by ASQC, Milwaukee, 1980.
- Sifakis, J., 1977a. Use of Petri nets for performance evaluation in measuring modelling and evaluationg computer systems. *North Holland*, pp.75–93.
- Sifakis, J., 1977b. Etude du comportement permanent des réseaux de Petri temporisés. *Journée AFCET sur les réseaux de Petri*. Paris, France.
- Sifakis, J., 1980. Use of Petri nets for performance evaluation. *Acta Cybernetica*, 4(2), pp.185–202.
- Tripakis, S., 2002. Fault Diagnosis for Timed Automata. *FTRTFT - Formal Techniques in Real-Time and Fault-Tolerant Systems*, pp.205–221.



- Trivedi, K. S., Bobbio, A., Ciardo, G., German, R., Puliafito, A. et Telek, M., 1995. Non-markovian petri nets. In *ACM SIGMETRICS Performance Evaluation Review*. ACM, pp.263–264.
- Ushio, T., Onishi, I. et Okuda, K., 1998. Fault detection based on Petri net models with faulty behaviors. In *Systems, Man, and Cybernetics. IEEE International Conference on*. San Diego, USA, pp.113–118.
- Valette, R., 1992. Les réseaux de Petri. Support de cours. France.
- Van Der Aalst, W.M.P., Arthur, H.M., Hofstede, T. et Weske, M., 2003. Business Process Management : A Survey. *LNCS Springer-Verlag*, pp.1–12.
- Villemeur, A., 1988. *Sureté de fonctionnement des systèmes industriels : fiabilité-facteurs humains, informatisation*, Eyrolles.
- Wang, X., Mahulea, C. et Silva, M., 2013. Fault diagnosis graph of time Petri nets. In *European Control Conference (ECC)*. IEEE. Zurich, Switzerland, pp.2459–2464.
- Wang, X., Mahulea, C. et Silva, M., 2015. Diagnosis of Time Petri Nets Using Fault Diagnosis Graph. *IEEE Transactions on Automatic Control*, 60(9), pp.2321–2335.
- Wen, Y., Li, C. et Jeng, M., 2005. A polynomial algorithm for checking diagnosability of Petri nets. In *Systems, Man and Cybernetics, IEEE International Conference*. Waikoloa, HI, USA, pp.2542–2547.
- Yoo, T.-S. et Lafortune, S., 2002. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on automatic control*, 47(9), pp.1491–1495.
- Zad, S.H., Kwong, R. et Wonham, W.M., 2003. Fault diagnosis indiscrete-event systems: framework and model reduction. *Transactions on Automatic Control*. IEEE, 48(7), pp.1199–1212.
- Zad, S.H., Kwong, R.H. et Wonham, W.M., 2005. Fault diagnosis in discrete-event systems: Incorporating timing information. *IEEE Transactions on Automatic Control*, 50(7), pp.1010–1015.
- Zaytoon, J. et Lafortune, S., 2013. Overview of fault diagnosis methods for Discrete Event Systems. *Annual Reviews in Control*, 37(2), pp.308–320.
- Zhang, F., 2006. AEWMA Control Charts for Monitoring Variability Sources of Solder Joints Quality. *IEEE Transactions on components and Packaging Technologies*, 29(1), pp.80–88.
- Zhou, Z.G. et Tang, P., 2016. Improving time series anomaly detection based on Exponentially Weighted Moving Average (EWMA) of season-trend model residuals. *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. Beijing, China.
- Zuberek, W.M., 1980. Timed Petri nets and preliminary performance evaluation. *7th IEEE Annual Symp. On Computer Architecture*, New York, USA, pp.89–96.



---

## Diagnostic des défauts dans les systèmes à événements discrets soumis à des contraintes temporelles

**Résumé.** Cette thèse porte sur le diagnostic des défauts dans des Systèmes à Événements Discrets (SED) pour lesquels l'occurrence des événements est soumise à des contraintes temporelles. Les domaines d'application potentiels sont nombreux et variés allant de la production manufacturière aux systèmes de transport en passant par les réseaux de communication et les systèmes d'information. La complexité croissante de ces systèmes nécessite l'élaboration de méthodes de surveillance de plus en plus efficaces et performantes pour garantir leur sécurité, leur disponibilité ainsi que le maintien de leurs propriétés dans le temps. Les réseaux de Petri Stochastiques Temporisés Partiellement Observés (RdPSTPO) sont utilisés pour modéliser le système ainsi que les défauts qui peuvent l'affecter. L'étude concerne particulièrement les défauts qui se traduisent par la violation des contraintes temporelles. Une exploitation pertinente des informations temporelles, en vue du diagnostic, constitue la contribution majeure de la thèse. En effet, une meilleure utilisation de ces informations permet de discerner avec précision les différents comportements qui expliquent les mesures. De plus, la probabilité d'occurrence des défauts est calculée en fonction des dates des mesures collectées. Deux approches sont développées : la première est dédiée aux défauts ponctuels et la seconde aux dérives lentes dans les systèmes cycliques.

**Mots clés.** Systèmes à événements discrets, Réseaux de Petri temporels, Détection des défauts, Diagnostic des défauts, Carte de contrôle.

---

## Faults Diagnosis in discrete event systems with temporal constraints

**Abstract.** This PhD thesis deals with the fault diagnosis of Discrete Event Systems (DES) for which the occurrence of events is constrained by temporal specifications. The domain of application is large and varied ranging from manufacturing systems to transportation systems, communication networks and information systems. The increasing complexity of these systems requires the development of effective and efficient monitoring methods to ensure their security, availability and the maintenance of their properties over the time. For that purpose, Partially Observed Stochastic Timed Petri Nets (POSTPN) are used to model the system and the different faults that may affect it. The study is particularly concerned with faults that result after the violation of temporal constraints. Our main contribution in this thesis is the relevant exploitation of the timed information for the fault diagnosis. In fact, a better use of these temporal informations allow to discern with precision the different behaviors that are consistent with the measurements. In addition, the probability of faults occurrence is evaluated according to the dates of collected measurements. Two approaches are developed: the first one is used for single faults and the second one for slow drifts in cyclical systems.

**Keywords.** Discrete event systems, Temporal Petri nets, Faults detection, Faults diagnosis, Control chart.