

ÉCOLE DOCTORALE D'ANGERS

2006

Thèse de DOCTORAT

Spécialité : Automatique et Informatique Appliquée

Présentée et soutenue publiquement par

Samir HAMACI

le 9 décembre 2005

à l'ISTIA - Université d'Angers

**Étude de graphes d'événements
temporisés avec multiplieurs dans l'algèbre
(min, +).**

Jury

Président	:	Jean-Jaques Lesage, Professeur	ENS Cachan
Rapporteurs	:	Hassane Alla, Professeur	INPG Grenoble
		Jean-Claude Gentina, Professeur	École Centrale de Lille
		Jean-Jacques Loiseau, Directeur de Recherche	CNRS, IRCCyN, Nantes
Examineurs	:	Jean-Louis Boimond, Professeur	Université d'Angers
		Jean-Louis Ferrier, Professeur	Université d'Angers

Directeurs de thèse : Jean-Louis Boimond

Laboratoire : LABORATOIRE D'INGÉNIERIE DES SYSTÈMES AUTOMATISÉS. 62, avenue Notre Dame
du Lac, F-49000 ANGERS

**ÉTUDE DES GRAPHES D'ÉVÉNEMENTS TEMPORISÉS
AVEC MULTIPLIEURS DANS L'ALGÈBRE (min, +).**

Samir HAMACI



Université d'Angers

Samir HAMACI

Étude des graphes d'événements temporisés avec multiplieurs dans l'algèbre

(min, +).

xviii+

Résumé

Les graphes d'événements temporisés avec multiplieurs forment une sous classe des réseaux de Petri dont le comportement ne peut pas être décrit par des équations linéaires dans l'algèbre $(\min,+)$ de par la présence des poids sur les arcs.

Ce mémoire est consacré à l'étude des systèmes dynamiques, modélisés par des graphes d'événements temporisés discrets, voire hybrides avec multiplieurs. Nous abordons le problème de la représentation entrée-sortie de ces graphes dans l'algèbre $(\min,+)$. Pour pallier le problème de non-linéarité, une approche de modélisation à base d'opérateurs est proposée. La commande en juste-à-temps est calculée pour les graphes d'événements temporisés discrets avec multiplieurs dans un dioïde d'opérateurs. Une extension de cette commande aux graphes d'événements temporisés hybrides avec multiplieurs est présentée. Est abordé ensuite, l'analyse des performances des graphes d'événements temporisés discret avec multiplieurs. Nous proposons deux méthodes de linéarisation de ces graphes, dans le but d'obtenir des représentations linéaires dans l'algèbre $(\min,+)$, ceci afin d'appliquer certains résultats de base de la théorie spectrale, habituellement utilisée pour évaluer les performances des graphes d'événements temporisés ordinaires.

Mots-clés : graphes d'événements temporisés discrets avec multiplieurs, graphes d'événements temporisés hybrides avec multiplieurs, dioïde, commande en juste-à-temps, algèbre $(\min,+)$, linéarisation, temps de cycle.

Abstract

Timed event graphs with multipliers form a subclass of Petri nets whose behavior can not be described by linear equations in $(\min,+)$ algebra. This nonlinearity is due to the presence of weights on arcs.

This report is devoted to the study of dynamic systems modelled by discrete timed event graphs with multipliers and hybrid timed event graphs with multipliers. We tackle the problem of input-output representation of these graphs in $(\min,+)$ algebra. To mitigate the problem of nonlinearity, an approach of modelling, based on operators, is proposed. A just in time control is calculated for discrete timed event graphs with multipliers. An extension of this control to the hybrid Timed event graphs with multipliers is presented. Then, the performance analysis of timed event graphs with multipliers, is tackled. We propose two linearization methods of these graphs, in the purpose to obtain a linear representation in $(\min,+)$ algebra, and to apply some basic results of the spectral theory, usually used to evaluate the performances of ordinary timed event graphs.

Keywords: discrete timed event graphs with multipliers, hybrid timed event graphs with multipliers, dioïd, just in time control, $(\min,+)$ algebra, linearization, cycle time.

Remerciements

Le travail présenté dans ce mémoire a été préparé au Laboratoire d'Ingénierie des Systèmes Automatisés dirigé par le Professeur **Jean-Louis Ferrier**. Je lui exprime ma profonde reconnaissance pour la confiance qu'il m'a témoigné en m'acceptant au sein de ce laboratoire.

Je tiens à exprimer particulièrement ma gratitude à **Jean-Louis Boimond**, Professeur à l'université d'Angers et directeur de cette thèse, pour son immense disponibilité et ses nombreux et précieux conseils. Ce travail n'aurait jamais abouti sans son aide. Qu'il trouve ici l'expression de mon respect et de ma profonde reconnaissance.

J'exprime toute ma reconnaissance à **Hassane Alla**, Professeur à l'INPG de Grenoble, **Jean-Jacques Loiseau**, Directeur de recherche CNRS à l'IRCCyN et **Jean-Claude Gentina**, Professeur à l'École Centrale de Lille, pour l'honneur qu'ils m'ont fait en acceptant de rapporter cette thèse.

Je voudrais également remercier **Jean-Jacques Lesage**, Professeur à l'École Normale Supérieure de Cachan, de l'intérêt qu'il a porté à ces travaux en acceptant de les examiner.

Mes remerciements vont également à **Sébastien Lahaye**, Maître de Conférences à l'université d'Angers, pour ses conseils et son intérêt permanent à ces travaux.

Enfin, je remercie également les collègues de laboratoire, en particulier les doctorants, pour leur témoignage d'amitié et pour l'ambiance de travail qu'ils ont su faire régner durant ces trois années.

*A mes chers parents,
mes frères, mes soeurs,
et ceux qui recommandent le bien et s'interdisent le mal.*

Sommaire

Table des matières	1
Notations	5
Introduction	7
1 Outils algébriques	11
2 Modélisation des graphes d'événements temporisés discrets, continus, hybrides avec multiplieurs	27
3 Sur la commande optimale des graphes d'événements temporisés discrets et hybrides avec multiplieurs	67
4 Analyse de performances des graphes d'événements temporisés discrets avec multiplieurs ..	83
5 Conclusion et Perspectives	115
Bibliographie	119

Table des matières

Table des matières	1
Notations	5
Introduction	7
1 Outils algébriques	11
1.1 Dioïde, moduloïde	11
1.1.1 Dioïde : définitions et exemples	11
1.1.2 Structure ordonnée d'un dioïde	13
1.1.3 Moduloïde	15
1.2 Théorie de la résiduation sur les dioïdes	16
1.2.1 Applications résiduables sur les dioïdes	16
1.2.2 Résiduation dans le cas matriciel	19
1.2.3 Résolution d'équations au point fixe dans un dioïde	19
1.3 Eléments de la théorie des systèmes linéaires	21
1.3.1 Signaux et systèmes	21
1.3.2 Systèmes élémentaires	23
1.3.3 Opérations sur les systèmes	24
1.4 Conclusion	25
2 Modélisation des graphes d'événements temporisés discrets, continus, hybrides avec multi- multiplicateurs	27
2.1 Graphes d'événements temporisés discrets	28
2.1.1 Réseaux de Petri : définitions et notations	29
2.1.2 Invariants d'un réseau de Petri	30
2.1.3 Graphes d'événements temporisés	33
2.1.4 Propriétés des graphes d'événements temporisés	34
2.1.5 Représentation d'état des graphes d'événements temporisés	35
2.1.6 Forme récurrente explicite de la représentation d'état	38
2.1.7 Conditions initiales des graphes d'événements temporisés	40
2.2 Graphes d'événements temporisés discrets avec multiplicateurs	41
2.2.1 Présentation des graphes d'événements temporisés discrets avec multiplicateurs	41

2.2.2	Equations récurrentes des graphes d'événements temporisés discrets avec multiplieurs	44
2.2.3	Approche opératoirelle pour modéliser les graphes d'événements temporisés discrets avec multiplieurs	45
2.2.4	Exemple	49
2.3	Graphes d'événements temporisés continus avec multiplieurs	50
2.3.1	Propriété des graphes d'événements temporisés continus avec multiplieurs	51
2.3.2	Equations récurrentes des graphes d'événements temporisés continus avec multiplieurs	53
2.3.3	Approche opératoirelle pour modéliser les graphes d'événements temporisés continus avec multiplieurs	54
2.3.4	Exemple	58
2.4	Graphes d'événements temporisés hybrides avec multiplieurs	59
2.4.1	Equations récurrentes des graphes d'événements temporisés hybrides avec multiplieurs	60
2.4.2	Approche opératoirelle pour modéliser les graphes d'événements temporisés hybrides avec multiplieurs	62
2.4.3	Exemple	64
2.5	Conclusion	65
3	Sur la commande optimale des graphes d'événements temporisés discrets et hybrides avec multiplieurs	67
3.1	Relation entrée-sortie dans le dioïde $\mathcal{D}_{\min}[\delta]$	69
3.2	Commande en juste-à-temps des graphes d'événements temporisés discrets avec multiplieurs	71
3.2.1	Equations <i>backward</i> dans le dioïde $\mathcal{D}_{\min}[\delta]$	71
3.2.2	Exemple	74
3.3	Commande en juste-à-temps des graphes d'événements temporisés hybrides avec multiplieurs	75
3.3.1	Modèle d'état <i>backward</i> hybride	75
3.3.2	Exemple	78
3.4	Conclusion	82
4	Analyse de performances des graphes d'événements temporisés discrets avec multiplieurs	83
4.1	Quelques rappels	84
4.2	Propriétés spectrales et asymptotiques des matrices de $\mathbb{R}_{\min}^{n \times n}$	87

4.3	Evaluation de performances des Graphes d'événements temporisés dans l'algèbre $(min, +)$	90
4.4	Evaluation de performances des graphes d'événements temporisés discrets avec multi- plieurs	94
4.5	Méthode de linéarisation des graphes d'événements temporisés discrets avec multiplieurs avec des conditions sur le marquage initial	98
4.5.1	Linéarisation exacte des graphes d'événements temporisés discrets avec multi- plieurs	98
4.5.2	Linéarisation approchée par ajout/retrait de jetons	100
4.5.3	Exemple	101
4.6	Méthode de linéarisation des graphes d'événements temporisés discrets avec multiplieurs avec une condition sur le T-invariant	106
4.6.1	Principe de linéarisation	107
4.6.2	Exemple	111
4.7	Conclusion	114
5	Conclusion et Perspectives	115
	Bibliographie	119

Notations

SED : système à événements discret.

\mathcal{P} : ensemble non vide de places d'un réseau de Petri.

\mathcal{T} : ensemble non vide de transitions d'un réseau de Petri.

m_p : nombre de jetons initialement contenus dans la place p .

τ_p : durée minimale de séjour d'un jeton dans la place p .

GET : graphe d'événements temporisé.

GETDM : graphe d'événements temporisé discret avec multiplieurs.

GETCM : graphe d'événements temporisé continu avec multiplieurs.

GETHM : graphe d'événements temporisé hybride avec multiplieurs.

θ : vecteur T-invariant associé au graphe d'événements temporisé discret avec multiplieurs.

$ppcm$: plus petit commun multiple.

M_{qp} : poids de l'arc allant de la place p vers la transition n_q .

$M_{pq'}$: poids de l'arc allant de la transition $n_{q'}$ vers la place p .

\mathcal{N}_p : marque valuée associée à la place p .

\mathcal{N}_ρ : marque valuée associée au circuit ρ .

\oplus : addition dans un dioïde.

\otimes : multiplication dans un dioïde.

\mathbb{Z}_{max} : algèbre $(\mathbb{Z} \cup \{-\infty\}, max, +)$.

$\overline{\mathbb{Z}}_{max}$: algèbre $(\mathbb{Z} \cup \{\pm\infty\}, max, +)$.

\mathbb{Z}_{min} : algèbre $(\mathbb{Z} \cup \{+\infty\}, min, +)$.

$\overline{\mathbb{Z}}_{min}$: algèbre $(\mathbb{Z} \cup \{\pm\infty\}, min, +)$.

\preceq : relation d'ordre liée à un dioïde.

a^* : étoile de kleene.

$x_q(t)$: variable compteur associée à la transition x_q d'un graphe d'événements temporisé discret.

$n_q(t)$: variable compteur associée à la transition n_q d'un graphe d'événements temporisé discret avec multiplieurs.

$f^\#$: application résiduée de f .

- f^\flat : application dualement résiduée de f .
- L_a : produit à gauche par a , $L_a(x) = a \otimes x$.
- R_a : produit à droite par a , $R_a(x) = x \otimes a$.
- $a \bowtie b$: notation utilisée pour représenter $L_a^\sharp(b)$.
- $b \bowtie a$: notation utilisée pour représenter $R_a^\sharp(b)$.
- $\mathcal{D}_{\min}[[\delta]]$: dioïde d'opérateurs discrets permettant de modéliser les graphes d'événements temporisés discrets avec multiplieurs.
- $\gamma^\nu, \nu \in \mathbb{Z} \cup \{\pm\infty\}$: opérateur de stock sur les compteurs discrets.
- $\delta^\tau, \tau \in \mathbb{Z} \cup \{\pm\infty\}$: opérateur de retard sur les compteurs discrets.
- $\mu_m, m \in \mathbb{Q}$: opérateur multiplieur sur les compteurs discrets.
- $\mathcal{C}_{\min}[[\delta]]$: dioïde d'opérateurs continus permettant de modéliser les graphes d'événements temporisés continus avec multiplieurs.
- $\gamma_c^\nu, \nu \in \mathbb{R} \cup \{+\infty\}$: opérateur de stock sur les compteurs continus.
- $\delta_c^\tau, \tau \in \mathbb{R}$: opérateur de retard sur les compteurs continus.
- $\mu_{m_c}, m \in \mathbb{R}$: opérateur multiplieur sur les compteurs continus.
- $\omega_V, V \in \mathbb{R}^+ \cup \{+\infty\}$: opérateur limitateur de flux associé aux transitions continus.
- P : (Projection canonique) opérateur modélisant l'interface continue-discrète (CD) d'un graphe d'événements temporisé hybride avec multiplieurs.
- I : (Injection canonique) opérateur modélisant l'interface discrète-continue (DC) d'un graphe d'événements temporisé hybride avec multiplieurs.
- $\overline{\mathbb{R}}_{min}^{\mathbb{R}}$: ensemble de signaux continus de \mathbb{R} dans $\overline{\mathbb{R}}_{min}$.
- $\overline{\mathbb{Z}}_{min}^{\mathbb{R}}$: ensemble de signaux discrets de \mathbb{Z} dans $\overline{\mathbb{Z}}_{min}$.
- U_{opt} : commande optimale vis-à-vis du critère de juste-à-temps.
- TC : temps de cycle d'un graphe d'événements temporisé.
- λ : taux de production d'un graphe d'événements temporisé.
- TC_m : temps de cycle d'un graphe d'événements temporisé discret avec multiplieurs.
- λ_{m_q} : fréquence de franchissement associée à la transition n_q d'un graphe d'événements temporisé discret avec multiplieurs.
- $\lceil a \rceil$: partie entière supérieure de a .
- $\lfloor a \rfloor$: partie entière inférieure de a .

Introduction

Un *système dynamique* est un modèle permettant de décrire l'évolution, au cours du temps, d'un ensemble d'objets en interaction [Willems, 1991]. Le terme *système* se réfère à un ensemble de variables d'états (dont la valeur évolue au cours du temps) et aux interactions entre ces variables. Selon J.Willems, un système dynamique peut être représenté par un triplet $(\mathfrak{S}, \mathcal{Q}, \mathcal{C})$, où :

- \mathfrak{S} est l'axe des temps (en général \mathbb{Z} ou \mathbb{R}),
- \mathcal{Q} est l'espace dans lequel évoluent les variables du système,
- \mathcal{C} est l'ensemble des trajectoires correspondant aux évolutions possibles du système.

Dans la littérature, ces systèmes dynamiques sont classés en trois catégories : les systèmes à événements discrets, les systèmes continus et les systèmes hybrides.

La première classe de systèmes regroupe les *systèmes à événements discrets* (SED) et recouvre plusieurs domaines d'applications tels que les systèmes de production manufacturière, la logistique, les systèmes de transport, les réseaux de communication, les circuits logiques, etc. Ces systèmes se définissent par des espaces d'états discrets dans lesquels toutes les variables d'états prennent leurs valeurs dans un ensemble dénombrable, et des évolutions, nommées trajectoires, basées sur une succession des états et des transitions. Plusieurs concepts (théories, méthodes, outils, modèles et langages) ont été élaborés afin de maîtriser la complexité croissante de la conception et du développement de ces systèmes.

Une approche courante pour l'étude de ces systèmes consiste à ignorer la valeur explicite du temps et à s'intéresser uniquement à l'ordre d'occurrence des événements. Les modèles non temporisés ainsi obtenus, sont généralement élaborés à l'aide d'automates à états finis, de grafsets, de réseaux de Petri (RdP), etc. Pour certaines applications, les informations temporelles sont cruciales et doivent être intégrées au modèle de manière à spécifier les trajectoires d'événements et leur temps. Ces informations peuvent être représentées par des modèles temporisés non stochastiques (automates temporisés, réseaux de Petri temporisés, etc.) où les durées des activités sont connues *a priori*, ou stochastiques (chaîne de Markov, réseaux de files d'attentes, réseaux de Petri stochastiques) basés sur des hypothèses statistiques.

Au début des années 80, une nouvelle théorie, permettant d'étudier une catégorie de SED, a vu le jour. Cette théorie concerne une sous classe de réseaux de Petri, appelés *graphes d'événements temporisés* (GET), dont l'évolution de l'état est représenté par des équations linéaires dans une structure algébrique particulière, appelée *l'algèbre des dioïdes*. Le comportement dynamique des systèmes carac-

térisés par des phénomènes de saturation et de synchronisation peut être décrit par de telles récurrences.

La deuxième classe de systèmes dynamiques regroupe les *systèmes continus*. Ces systèmes mettent en jeu des phénomènes essentiellement *naturels* dont le comportement dynamique se décrit par des équations différentielles ou aux différences, linéaires ou non linéaires.

Dans le cas d'un système de production, lorsque le nombre de pièces qui y circulent est élevé, le nombre des marquages accessibles peut être important. En effet, l'usage des modèles RdP discrets est limité, d'où la nécessité de disposer d'un nouvel outil de modélisation. Pour ces systèmes, des modèles réseaux de Petri continus à vitesses constantes [David and Alla, 1987, Alla and David, 1998] et à vitesses variables [Zerhouni and Alla, 1992] ont été proposés.

Si l'utilisation des modèles discrets et des modèles continus suffit pour modéliser certains systèmes, il s'avère que, dans de nombreux cas, la combinaison des deux caractères (continu et discret) dans un même modèle est indispensable ; d'où la notion de *modèle hybride* introduite dans [David and Alla, 1992, Branicky, 1995, Champagnat et al., 1998]. Par exemple, dans un système de production où une matière liquide est conditionnée, le produit est initialement continu pour devenir discret à la fin du processus (liquide mis en bouteilles par exemple) [Audry et al., 1994].

Lorsque la taille du modèle est importante, les techniques développées dans le cas discret atteignent leur limite. L'utilisation des graphes d'événements temporisés à arcs pondérés [Teruel et al., 1992], [Chao et al., 1993, Munier, 1993], encore appelés graphes d'événements temporisés avec multiplieurs, peut réduire la taille du modèle. De plus, pour un système de production, l'emploi de ces graphes permet de modéliser simplement l'assemblage et le désassemblage de lots de produits effectués durant la fabrication.

Les travaux présentés dans ce mémoire visent à étendre certains résultats établis sur la modélisation, la commande et l'analyse de performances des graphes d'événements temporisés discrets (classiquement étudiés dans le cadre de l'algèbre des dioïdes) aux graphes d'événements temporisés discrets, voire hybride, avec multiplieurs.

Ce mémoire est scindé en quatre chapitres :

- Dans le premier chapitre nous présentons les outils algébriques nécessaires à la modélisation et à la commande des graphes d'événements temporisés discrets ou hybrides avec multiplieurs. Quelques résultats de base sur l'algèbre des dioïdes sont rappelés. La théorie de la résiduation est ensuite présentée. La dernière partie de ce chapitre est consacrée à un ensemble de rappels sur la théorie

des systèmes utiles pour modéliser les graphes étudiés.

- Le second chapitre a trait à la modélisation des graphes avec multipliers dans un dioïde d'opérateurs. Nous présentons le dioïde d'opérateurs discrets $\mathcal{D}_{\min}[\delta]$ qui sera, par la suite, utilisé comme la structure algébrique pour l'étude et l'analyse de graphes d'événements temporisés discrets avec multipliers. Afin d'aborder la modélisation des graphes d'événements temporisés hybrides avec multipliers, le dioïde d'opérateurs continus $\mathcal{C}_{\min}[\delta]$, utilisé pour la modélisation de la partie continue de ces graphes, sera ensuite présenté. Le résultat principal de ce chapitre est la modélisation proposée des graphes d'événements temporisés hybrides avec multipliers. Cette modélisation se base sur une composition d'opérateurs, elle est, à notre connaissance, originale. Cette originalité réside dans l'introduction de deux opérateurs modélisant les interfaces continue-discrète et discrète-continue.
- Le troisième chapitre aborde le problème de commande des graphes d'événements temporisés discrets avec multipliers, dans un dioïde d'opérateurs. Seul le problème de poursuite de trajectoire, antérieurement résolu pour les systèmes linéaires dans le dioïde $(\min, +)$, est dans ce mémoire étendu au cas des systèmes non linéaires n'admettant pas une représentation $(\min, +)$ linéaire. L'objectif est de synthétiser une trajectoire d'entrée (commande) telle que la réponse en sortie suive "au mieux" une trajectoire de consigne correspondant au comportement désiré en sortie du système. En s'inspirant des travaux de Jan Komenda [Komenda et al., 1998], qui considère seulement la classe des graphes d'événements temporisés hybrides à interface discrète-continue, une extension de cette commande à toutes les classes des graphes d'événements temporisés hybrides avec multipliers est proposée.
- Le quatrième chapitre est consacré à l'analyse de performances des graphes d'événements temporisés discrets avec multipliers. Pour analyser les performances de ces graphes, nous utilisons certains résultats de base de la théorie spectrale, utilisés pour évaluer les performances des graphes d'événements temporisés ordinaires. Pour cela, deux nouvelles méthodes de linéarisation sont proposées. Ces méthodes ont pour but de linéariser le modèle mathématique associé aux graphes d'événements temporisés discrets avec multipliers afin d'obtenir des GET ordinaires équivalents.
 - La première méthode est inspirée des travaux proposés par Guy Cohen dans [Cohen et al., 1998c]. Elle a pour principe de linéariser les graphes d'événements temporisés discrets avec multipliers sous réserve de vérifier une condition de linéarisation sur le marquage initial. Dans le cas où cette condition est satisfaite, les performances du GETDM peuvent être déterminées de façon

exacte. Dans le cas où cette condition n'est pas vérifiée, nous procédons à un ajout ou à un retrait de jetons dans le graphe afin de satisfaire la condition de linéarisation pour toutes les places de ce dernier. Cette technique d'analyse nous permet d'encadrer les performances du GET discret avec multiplicateurs entre deux bornes : une valeur supérieure obtenue par l'ajout d'un nombre minimal de jetons dans le graphe, et une valeur inférieure obtenue par le retrait d'un nombre minimal de jetons.

- La deuxième méthode est inspirée des travaux, sur la linéarisation, proposés par Benoît Trouillet dans [Trouillet et al., 2002]. Elle a trait aux graphes d'événements temporisés discrets avec multiplicateurs admettant un T-invariant dont au moins une de ses composantes est égale à un. Cette composante est associée à une transition dite de *normalisation*. Cette technique de linéarisation a pour principe d'exprimer chaque variable d'état du graphe, appelée *compteur*, en fonction du compteur associé à la transition de *normalisation*. Cette dernière présente un intérêt au sens où elle permet de linéariser les équations non linéaires associées aux transitions du graphe.
- En conclusion, nous faisons une synthèse sur le travail réalisé et proposons des perspectives de recherche.

CHAPITRE 1

Outils algébriques

Ce premier chapitre a pour objectif d'expliciter les outils algébriques utilisés dans ce mémoire. Sans être exhaustif, nous présentons, dans la première partie, un ensemble de définitions, de notations et résultats relatifs à l'algèbre des dioïdes. Nous rappelons qu'un dioïde de par la propriété d'idempotence de la loi additive, notée \oplus , est une structure algébrique que l'on peut munir d'une relation d'ordre. Du fait que la loi additive et la loi multiplicative, notée \otimes , d'un dioïde n'ont pas nécessairement d'inverse, la deuxième partie est consacrée à la théorie de la résiduation qui est une alternative au problème d'inversion d'applications définies sur des dioïdes. La troisième partie de ce chapitre présente quelques éléments de la théorie des systèmes linéaires dans les dioïdes. Nous donnons un ensemble de définitions et notations relatives aux signaux et systèmes utilisés par la suite.

1.1 Dioïde, moduloïde

Cette section rappelle certaines propriétés propres à la structure algébrique d'un dioïde. Pour une présentation plus détaillée, on renvoie le lecteur à l'ouvrage [Baccelli et al., 1992] (accessible gratuitement à la page <http://www-rocq.inria.fr/metalau/cohen/SED/book-online.html>) et à la thèse de doctorat de Stéphane Gaubert [Gaubert, 1992].

1.1.1 Dioïde : définitions et exemples

Définition 1.1. On appelle *demi-anneau* un ensemble \mathcal{D} muni de deux lois internes \oplus et \otimes telles que :

- la loi additive \oplus est associative et commutative, elle admet un élément neutre, noté ε , appelé élément zéro, tel que $\forall a \in \mathcal{D}, a \oplus \varepsilon = \varepsilon \oplus a = a$;
- la loi multiplicative \otimes est associative, elle admet un élément neutre, noté e , appelé élément identité, tel que $\forall a \in \mathcal{D}, a \otimes e = e \otimes a = a$;
- la loi multiplicative \otimes est distributive à gauche et à droite par rapport à la loi additive \oplus :
$$\forall a, b, c \in \mathcal{D}, (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \quad \text{et} \quad c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b) ;$$
- l'élément zéro est absorbant pour la multiplication : $\forall a \in \mathcal{D}, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.

Si la loi additive est idempotente, c'est-à-dire $\forall a \in \mathcal{D}, a \oplus a = a$ alors $(\mathcal{D}, \oplus, \otimes)$ est un demi anneau idempotent, également appelé dioïde.

On dit qu'un dioïde est commutatif si la loi \otimes est commutative. Comme dans l'algèbre classique, on note $a \otimes b$ par ab lorsqu'il n'y a pas d'ambiguïté. Notons que le terme $a^{\otimes k}$, également noté a^k lorsqu'il n'y a pas d'ambiguïté, $k \in \mathbb{N}$, désigne le produit $\underbrace{a \otimes a \dots \otimes a}_{k \text{ fois}}$, sachant que $a^0 = e$.

Exemple 1.2. (Dioïdes $\mathbb{R}_{min}, \mathbb{R}_{max}$)

- $\mathbb{R}_{min} = (\mathbb{R} \cup \{+\infty\}, min, +)$ est un dioïde commutatif où, la loi additive \oplus correspond à l'opération *min* usuelle ; la loi multiplicative \otimes est équivalente à l'addition usuelle. L'élément zéro de \mathbb{R}_{min} est $\varepsilon = +\infty$ et l'élément identité est $e = 0$. Ce dioïde est appelé *l'algèbre (min, +)*.
- $\mathbb{R}_{max} = (\mathbb{R} \cup \{-\infty\}, max, +)$ est un dioïde commutatif où la loi additive \oplus correspond à l'opération *max* usuelle ; la loi multiplicative \otimes est équivalente à l'addition usuelle et pour lequel $\varepsilon = -\infty$ et $e = 0$. Ce dioïde est appelé *l'algèbre (max, +)*.

Définition 1.3 (Dioïde matriciel). L'ensemble des matrices carrées de dimension n , à coefficients dans un dioïde $(\mathcal{D}, \oplus, \otimes)$, est un dioïde matriciel, noté $(\mathcal{D}^{n \times n}, \oplus, \otimes)$, où les opérations sont définies, à partir des opérations du dioïde $(\mathcal{D}, \oplus, \otimes)$ de manière analogue à l'algèbre classique, de la façon suivante :

$$\forall A, B \in \mathcal{D}^{n \times n};$$

$$A \oplus B : (A \oplus B)_{ij} = A_{ij} \oplus B_{ij}, \quad \forall i, j = 1, \dots, n,$$

$$A \otimes B : (A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}, \quad \forall i, j = 1, \dots, n.$$

L'élément identité de $\mathcal{D}^{n \times n}$ est la matrice, notée Id_n , composée de e sur la diagonale et de ε partout ailleurs. L'élément zéro est la matrice composée exclusivement de ε et est noté ε .

La somme et le produit de deux matrices de dimensions compatibles, pas nécessairement carrées, peuvent être définis de la façon suivante :

$$A \in \mathcal{D}^{n \times p}, \quad B \in \mathcal{D}^{n \times p}; \quad A \oplus B : (A \oplus B)_{ij} = A_{ij} \oplus B_{ij}, \quad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, p,$$

$$C \in \mathcal{D}^{n \times p}, \quad D \in \mathcal{D}^{p \times q}; \quad C \otimes D : (C \otimes D)_{ij} = \bigoplus_{k=1}^p C_{ik} \otimes D_{kj}, \quad \forall i = 1, \dots, n, \quad \forall j = 1, \dots, q.$$

Pour que ces matrices puissent être manipulées comme des éléments d'un dioïde matriciel, il faut, en toute rigueur, considérer qu'elles appartiennent au dioïde de matrices carrées de dimension $max(n, p, q) \times max(n, p, q)$, en les complétant, si nécessaire, pour cela de lignes et/ou de colonnes constituées de l'élément ε .

Définition 1.4 (Dioïde de séries formelles). Soit $(\mathcal{D}, \oplus, \otimes)$ un dioïde, on définit une série formelle en q variables, notées p_1 à p_q , à coefficients dans \mathcal{D} comme une application Π de \mathbb{Z}^q dans \mathcal{D} : $\forall \kappa = (\kappa_1, \dots, \kappa_q) \in \mathbb{Z}^q$, $\Pi(\kappa)$ représente le coefficient de $p_1^{\kappa_1} \dots p_q^{\kappa_q}$. Une autre représentation équivalente de la série Π est :

$$\Pi = \bigoplus_{\kappa \in \mathbb{Z}^q} \Pi(\kappa_1, \dots, \kappa_q) p_1^{\kappa_1} \dots p_q^{\kappa_q}.$$

L'ensemble des séries formelles muni des opérations :

$$\Pi \oplus \Psi : (\Pi \oplus \Psi)(\kappa) = \Pi(\kappa) \oplus \Psi(\kappa),$$

$$\Pi \otimes \Psi : (\Pi \otimes \Psi)(\kappa) = \bigoplus_{i+j=\kappa} \Pi(i) \otimes \Psi(j),$$

est un dioïde noté $\mathcal{D}[[p_1, \dots, p_q]]$.

On appelle *support* de la série formelle Π l'ensemble :

$$\text{Supp}(\Pi) = \{\kappa \in \mathbb{Z}^q \mid \Pi(\kappa) \neq \varepsilon\}.$$

Une série formelle à support fini est appelée *polynôme*. Une série formelle dont le support est un singleton (de \mathbb{Z}^q) est appelée *monôme*.

Définition 1.5 (Sous dioïde). Soient $(\mathcal{D}, \oplus, \otimes)$ un dioïde et un ensemble $\mathcal{C} \subset \mathcal{D}$, $(\mathcal{C}, \oplus, \otimes)$ est un sous dioïde de $(\mathcal{D}, \oplus, \otimes)$ si :

- $\varepsilon \in \mathcal{C}$ et $e \in \mathcal{C}$,
- \mathcal{C} est fermé pour les lois \oplus et \otimes .

Exemple 1.6 (Dioïdes \mathbb{Z}_{max} , \mathbb{Z}_{min}). Les dioïdes $\mathbb{Z}_{max} = (\mathbb{Z} \cup \{-\infty\}, max, +)$ et $\mathbb{Z}_{min} = (\mathbb{Z} \cup \{+\infty\}, min, +)$ sont respectivement des sous dioïdes de \mathbb{R}_{max} et \mathbb{R}_{min} .

1.1.2 Structure ordonnée d'un dioïde

L'idempotence de la loi additive \oplus permet de définir une relation d'ordre, notée \preceq ou \succeq , dans un dioïde. En effet, soit $(\mathcal{D}, \oplus, \otimes)$ un dioïde, alors la relation \preceq définie par :

$$a \preceq b \Leftrightarrow b = a \oplus b$$

est une *relation d'ordre*. En effet :

- $a \oplus a = a$ du fait de l'idempotence de la loi \oplus , d'où $a \preceq a$ (réflexivité),
- si $a \preceq b$ et $b \preceq a$ alors $b = a \oplus b$ et $a = b \oplus a$, d'où $a = b$ (antisymétrie),

- si $a \preceq b$ et $b \preceq c$ alors $b = a \oplus b$ et $c = b \oplus c$, d'où $c = b \oplus c = (a \oplus b) \oplus c = a \oplus (b \oplus c) = a \oplus c$.
On en déduit que $a \preceq c$ (transitivité).

Cette relation d'ordre (partielle) est compatible avec les lois \oplus et \otimes , c'est-à-dire :

- $a \preceq b \Rightarrow \forall c \in \mathcal{D}, a \oplus c \preceq b \oplus c$,
- $a \preceq b \Rightarrow \forall c \in \mathcal{D}, a \otimes c \preceq b \otimes c$ et $c \otimes a \preceq c \otimes b$.

La relation d'ordre est *totale* si :

$$\forall a, b \in \mathcal{D}, a \succeq b \text{ ou } b \succeq a.$$

Une condition nécessaire et suffisante pour que l'ordre d'un dioïde soit total, s'écrit de façon évidente comme suit :

$$\forall a, b \in \mathcal{D}, a \oplus b = a \text{ ou } b.$$

Exemple 1.7. L'ordre total \succeq défini dans \mathbb{R}_{max} coïncide avec l'ordre usuel, par exemple, $5 \succeq 4$, en effet $max(4, 5) = 5$. Par contre, l'ordre \succeq défini dans \mathbb{R}_{min} est total et est l'inverse de l'ordre usuel \geq , par exemple, $4 \succeq 5$, en effet $min(4, 5) = 4$.

Définition 1.8 (Dioïde complet). Un dioïde $(\mathcal{D}, \oplus, \otimes)$ est *complet* si :

- il est fermé pour les sommes infinies,
- la loi \otimes distribue sur les sommes infinies, c'est-à-dire si pour tout $a \in \mathcal{D}$ et tout sous-ensemble $\mathcal{B} \subseteq \mathcal{D}$,

$$a \otimes \left(\bigoplus_{b \in \mathcal{B}} b \right) = \bigoplus_{b \in \mathcal{B}} (a \otimes b) \quad \text{et} \quad \left(\bigoplus_{b \in \mathcal{B}} b \right) \otimes a = \bigoplus_{b \in \mathcal{B}} (b \otimes a).$$

Un dioïde complet admet un élément maximum $\bigoplus_{a \in \mathcal{D}} a$, que l'on notera T pour "Top".

L'élément T est absorbant pour l'addition, autrement dit, $\forall a \in \mathcal{D}, T \oplus a = T$.

Notons que, d'après la définition d'un dioïde, l'élément zéro ε est absorbant pour la multiplication pour tout élément de \mathcal{D} , aussi, on a : $T \otimes \varepsilon = \varepsilon \otimes T = \varepsilon$.

Exemple 1.9. On vérifie facilement que les dioïdes suivants sont complets :

- $(\mathbb{Z} \cup \{-\infty, +\infty\}, min, +)$ avec $\varepsilon = +\infty, e = 0$ et $T = -\infty$ est un dioïde noté par la suite : $\overline{\mathbb{Z}}_{min}$. On a dans $\overline{\mathbb{Z}}_{min}$ la règle suivante :

$$(-\infty) + (+\infty) = (+\infty).$$

- $(\mathbb{Z} \cup \{-\infty, +\infty\}, max, +)$ avec $\varepsilon = -\infty, e = 0$ et $T = +\infty$ est un dioïde noté par la suite : $\overline{\mathbb{Z}}_{max}$. On a dans $\overline{\mathbb{Z}}_{max}$ la règle suivante :

$$(-\infty) + (+\infty) = (-\infty).$$

Définition 1.10 (Dioïde distributif). $(\mathcal{D}, \oplus, \otimes)$ est un dioïde *distributif* si $\mathcal{D}, \oplus, \otimes$ est complet, et si $\forall b \in \mathcal{D}$ et pour tout sous ensemble \mathcal{B} de \mathcal{D} , on a :

$$\left(\bigwedge_{a \in \mathcal{B}} a \right) \oplus b = \bigwedge_{a \in \mathcal{B}} (a \oplus b),$$

et

$$\left(\bigoplus_{a \in \mathcal{B}} a \right) \wedge b = \bigoplus_{a \in \mathcal{B}} (a \wedge b).$$

1.1.3 Moduloïde

Définition 1.11 (Monoïde). Un *monoïde* (\mathcal{M}, \oplus) est un ensemble \mathcal{M} muni d'une loi de composition interne, notée \oplus , associative et possédant un élément neutre, noté ε . Si cette loi est commutative, le monoïde est commutatif.

Définition 1.12 (Moduloïde). Soit $(\mathcal{D}, \oplus, \otimes)$ un dioïde, on appelle *moduloïde* sur \mathcal{D} , un monoïde commutatif (\mathcal{M}, \oplus) muni d'une loi de composition externe " \cdot ", $\mathcal{D} \times \mathcal{M} \rightarrow \mathcal{M}$ vérifiant pour tout $a, b \in \mathcal{D}, \alpha_1, \alpha_2 \in \mathcal{M}$,

1. $(a \oplus b) \cdot \alpha_1 = a \cdot \alpha_1 \oplus b \cdot \alpha_1$,
2. $(a \otimes b) \cdot \alpha_1 = a \cdot (b \cdot \alpha_1)$,
3. $a \cdot (\alpha_1 \oplus \alpha_2) = a \cdot \alpha_1 \oplus a \cdot \alpha_2$,
4. $\varepsilon \cdot \alpha_1 = \varepsilon$,
5. $e \cdot \alpha_1 = \alpha_1$.

L'addition d'un moduloïde est idempotente, en effet :

$$\forall a \in (\mathcal{M}, \oplus, \cdot), \quad a \oplus a = e \cdot a \oplus e \cdot a = (e \oplus e) \cdot a = e \cdot a = a.$$

Comme pour un dioïde, l'idempotence de l'addition permet de doter un moduloïde d'une relation d'ordre \succeq définie par :

$$\alpha_1 \succeq \alpha_2 \Leftrightarrow \alpha_1 = \alpha_1 \oplus \alpha_2.$$

Exemple 1.13 (Moduloïde $\overline{\mathbb{R}}_{max}^{\mathbb{Z}}$ et $\overline{\mathbb{R}}_{min}^{\mathbb{Z}}$). L'ensemble des applications de \mathbb{Z} dans $\overline{\mathbb{R}}$ muni du max point-à-point est un monoïde commutatif, noté ici (\mathcal{M}, \oplus) . Ce monoïde, doté de plus de la loi de composition externe :

$$\begin{aligned} \overline{\mathbb{R}}_{max} \times \mathcal{M} &\rightarrow \mathcal{M} \\ (\alpha, x) &\mapsto \alpha \cdot x \quad : \quad t \in \mathbb{Z}, (\alpha \cdot x)(t) = \alpha \otimes x(t) = \alpha + x(t), \end{aligned}$$

est un moduloïde sur $\overline{\mathbb{R}}_{max}$, traditionnellement noté $\overline{\mathbb{R}}_{max}^{\mathbb{Z}}$.

Définition 1.14 (Sous-moduloïde). On appelle un *sous-moduloïde* $(\mathcal{M}, \oplus, \cdot)$ sur \mathcal{D} , toute partie non vide \mathcal{N} de \mathcal{M} , stable pour la loi \oplus et la loi externe " \cdot ", et telle que $(\mathcal{N}, \oplus, \cdot)$ forme un moduloïde sur \mathcal{D} .

Exemple 1.15. L'ensemble des applications *croissantes* de \mathbb{Z} dans $\overline{\mathbb{R}}$ muni des mêmes lois de composition que celles de $\overline{\mathbb{R}}_{max}^{\mathbb{Z}}$ est un sous-moduloïde de $\overline{\mathbb{R}}_{max}^{\mathbb{Z}}$.

1.2 Théorie de la résiduation sur les dioïdes

1.2.1 Applications résiduables sur les dioïdes

Les applications sur les ensembles ordonnés sont généralement non inversibles. D'un point de vue pratique, il est souvent suffisant de disposer, à défaut de la solution d'une équation, de la plus grande ou de la plus petite solution de l'inéquation correspondante. Ceci donne une alternative à l'inversion d'applications lorsque celles-ci ne sont pas bijectives. Dans la suite de cette section, nous supposons que \mathcal{C} et \mathcal{D} sont deux dioïdes complets, et que Π est une application de \mathcal{C} dans \mathcal{D} .

Définition 1.16 (Application résiduable, application dualement résiduable).

Une application $\Pi : (\mathcal{C}, \preceq) \rightarrow (\mathcal{D}, \preceq)$ isotone¹ est dite *résiduable*, si l'inéquation $\Pi(x) \preceq y$ admet une plus grande solution dans \mathcal{C} pour tout $y \in \mathcal{D}$. L'application Π est dite *dualement résiduable*, si l'inéquation $\Pi(x) \succeq y$ admet une plus petite solution dans \mathcal{C} pour tout $y \in \mathcal{D}$.

Les applications résiduables, ou dualement résiduables, sur les dioïdes complets ont de nombreuses propriétés, dont nous pouvons trouver les démonstrations dans [Baccelli et al., 1992, §4]. Rappelons ici quelques définitions issues de [Baccelli et al., 1992, Gaubert, 1992].

Définition 1.17 (Application semi-continue supérieurement, application semi-continue inférieurement).

- Une application $\Pi : \mathcal{C} \rightarrow \mathcal{D}$ est dite *semi-continue supérieurement* si pour tout sous ensemble I de \mathcal{C} ,

$$\bigwedge_{x \in I} \Pi(x) \in \mathcal{D} \text{ existe et } \Pi\left(\bigwedge_{x \in I} x\right) = \bigwedge_{x \in I} \Pi(x).$$

- L'application Π est dite *semi-continue inférieurement* si pour tout sous ensemble I de \mathcal{C} ,

$$\bigoplus_{x \in I} \Pi(x) \in \mathcal{D} \text{ existe et } \Pi\left(\bigoplus_{x \in I} x\right) = \bigoplus_{x \in I} \Pi(x).$$

¹Le qualificatif isotone est utilisé pour désigner les applications croissantes définies sur des ensembles ordonnés.

Théorème 1.18. [Baccelli et al., 1992]

Soit Π une application isotone d'un dioïde complet \mathcal{C} dans un dioïde complet \mathcal{D} . Sont équivalents :

- (i) Π est résiduable.
- (ii) Π est semi-continue inférieurement et $\Pi(\varepsilon_{\mathcal{C}}) = \varepsilon_{\mathcal{D}}$.
- (iii) Il existe une application unique, isotone et semi-continue supérieurement $\Pi^{\sharp} : \mathcal{D} \rightarrow \mathcal{C}$ telle que :

$$\Pi \circ \Pi^{\sharp} \preceq I_{\mathcal{D}},$$

$$\Pi^{\sharp} \circ \Pi \succeq I_{\mathcal{C}},$$

$I_{\mathcal{D}}$ et $I_{\mathcal{C}}$ étant respectivement les applications identités de \mathcal{D} et \mathcal{C} . L'application Π^{\sharp} est appelée application résiduée de Π .

De façon duale :

- (i) Π est dualement résiduable.
- (ii) Π est semi-continue supérieurement et $\Pi(T_{\mathcal{C}}) = T_{\mathcal{D}}$.
- (iii) Il existe une application unique, isotone et semi-continue inférieurement $\Pi^{\flat} : \mathcal{D} \rightarrow \mathcal{C}$ telle que

$$\Pi \circ \Pi^{\flat} \succeq I_{\mathcal{D}},$$

$$\Pi^{\flat} \circ \Pi \preceq I_{\mathcal{C}}.$$

L'application Π^{\flat} est appelée application dualement résiduée de Π .

Théorème 1.19. [Baccelli et al., 1992] Soient $\Pi : \mathcal{C} \rightarrow \mathcal{D}$ et $\Psi : \mathcal{D} \rightarrow \mathcal{E}$ deux applications résiduables. Ces applications satisfont les applications suivantes :

$$\Pi = \Pi \circ \Pi^{\sharp} \circ \Pi \text{ et } \Pi^{\sharp} = \Pi^{\sharp} \circ \Pi \circ \Pi^{\sharp}, \quad (1.1)$$

$$(\Psi \circ \Pi)^{\sharp} = \Pi^{\sharp} \circ \Psi^{\sharp}, \quad (1.2)$$

$$\Pi \text{ est injective} \Leftrightarrow \Pi^{\sharp} \circ \Pi = Id_{\mathcal{C}} \Leftrightarrow \Pi^{\sharp} \text{ est surjective}, \quad (1.3)$$

$$\Pi \text{ est surjective} \Leftrightarrow \Pi \circ \Pi^{\flat} = Id_{\mathcal{D}} \Leftrightarrow \Pi^{\flat} \text{ est injective}. \quad (1.4)$$

Définition 1.20 (Homomorphisme). Soient \mathcal{C} et \mathcal{D} deux dioïdes et $\Pi : \mathcal{C} \rightarrow \mathcal{D}$. L'application Π est un homomorphisme si $\forall a, b \in \mathcal{C}$:

$$\Pi(a \oplus b) = \Pi(a) \oplus \Pi(b) \text{ et } \Pi(\varepsilon_{\mathcal{C}}) = \varepsilon_{\mathcal{D}},$$

$$\Pi(a \otimes b) = \Pi(a) \otimes \Pi(b) \text{ et } \Pi(e) = e.$$

Définition 1.21 (Isomorphisme). L'application $\Pi : \mathcal{C} \rightarrow \mathcal{D}$ est un isomorphisme si $\Pi^{-1} : \mathcal{D} \rightarrow \mathcal{C}$ existe et si Π et Π^{-1} sont des homomorphismes.

Nous aurons besoin, par la suite, des applications résiduées sur des dioïdes complets. Les applications L_a et R_a correspondent à la multiplication, respectivement à gauche et à droite, par l'élément a de \mathcal{D} , on a :

$$L_a : \mathcal{D} \rightarrow \mathcal{D}$$

$$x \mapsto L_a(x) = a \otimes x,$$

et

$$R_a : \mathcal{D} \rightarrow \mathcal{D}$$

$$x \mapsto R_a(x) = x \otimes a.$$

Sachant que \mathcal{D} est un dioïde complet, L_a et R_a sont des applications semi-continues inférieurement et $L_a(\varepsilon) = R_a(\varepsilon) = \varepsilon$, d'après le théorème 1.18, L_a et R_a sont résiduables. Les applications résiduées sont notées respectivement L_a^\sharp et R_a^\sharp .

Notation 1. Les applications résiduées L_a^\sharp et R_a^\sharp sont également notées :

$$L_a^\sharp(x) = a \backslash x,$$

$$R_a^\sharp(x) = a / x.$$

Théorème 1.22. Les applications L_a^\sharp et R_a^\sharp vérifient les propriétés suivantes :

$$a(a \backslash x) \preceq x \qquad (x / a)a \preceq x \qquad (1.5)$$

$$a \backslash (ax) \succeq x \qquad (xa) / a \succeq x \qquad (1.6)$$

$$a(a \backslash (ax)) = ax \qquad ((xa) / a)a = xa \qquad (1.7)$$

$$a \backslash (x \wedge y) = a \backslash x \wedge a \backslash y \qquad (x \wedge y) / a = x / a \wedge y / a \qquad (1.8)$$

$$(a \oplus b) \backslash x = a \backslash x \wedge b \backslash x \qquad x / (a \oplus b) = x / a \wedge x / b \qquad (1.9)$$

$$(ab) \backslash x = b \backslash (a \backslash x) \qquad x / (ba) = (x / a) / b \qquad (1.10)$$

$$b(a \backslash x) \preceq (a / b) \backslash x \qquad (x / a)b \preceq x / (b \backslash a) \qquad (1.11)$$

$$(a \backslash x)b \preceq a \backslash (xb) \qquad b(x / a) \preceq (bx) / a \qquad (1.12)$$

Le lecteur peut trouver les preuves de ces propriétés dans [Baccelli et al., 1992, p.182-185], [Gaubert, 1992, §5.3].

1.2.2 Résiduation dans le cas matriciel

Il s'agit d'étendre les applications résiduées de l'addition et de la multiplication au cas des dioïdes matriciels, sachant que les résultats décrits dans la partie précédente ont été développés sur des dioïdes généraux (scalaires). Considérons à présent l'application L_A définie sur le dioïde matriciel $\mathcal{D}^{n \times n}$:

$$L_A : \mathcal{D}^{n \times n} \rightarrow \mathcal{D}^{n \times n}$$

$$X \mapsto L_A(X) = A \otimes X, \quad \text{avec } A \in \mathcal{D}^{n \times n},$$

on obtient les résultats suivants.

Théorème 1.23. *Soit \mathcal{D} un dioïde complet et $\mathcal{D}^{n \times n}$ le dioïde des matrices à valeurs dans \mathcal{D} . Soit $B \in \mathcal{D}^{n \times n}$, la plus grande solution de $AX \preceq B$ est la matrice $L_A^\sharp(B) \in \mathcal{D}^{n \times n}$, notée également $A \bowtie B$. Les éléments de cette matrice sont donnés par la relation :*

$$(A \bowtie B)_{ij} = \bigwedge_{l=1}^n A_{li} \bowtie B_{lj}, \quad \text{pour } i, j = 1, \dots, n.$$

Le théorème suivant exprime les applications résiduées de L_A et R_A dans le cas où les matrices considérées ne sont pas nécessairement de dimension carrée. Soient les applications :

$$L_A : \mathcal{D}^{p \times q} \rightarrow \mathcal{D}^{n \times q} \qquad R_{A'} : \mathcal{D}^{q \times p} \rightarrow \mathcal{D}^{q \times n}$$

$$X \mapsto AX \quad (A \in \mathcal{D}^{n \times p}), \qquad X \mapsto XA' \quad (A' \in \mathcal{D}^{p \times n}).$$

Les inéquations $AX \preceq B$ et $XA' \preceq B'$ admettent $L_A^\sharp(B)$ et $R_{A'}^\sharp(B')$ comme plus grandes solutions respectives. Les expressions de ces matrices s'obtiennent *via* le théorème suivant.

Théorème 1.24. *Soient $B \in \mathcal{D}^{n \times q}$, $B' \in \mathcal{D}^{q \times n}$, on a :*

$$(L_A^\sharp(B))_{ij} = (A \bowtie B)_{ij} = \bigwedge_{l=1}^n A_{li} \bowtie B_{lj}, \quad i = 1, \dots, p, \quad j = 1, \dots, q,$$

$$(R_{A'}^\sharp(B'))_{ij} = (B' \phi A')_{ij} = \bigwedge_{l=1}^n B'_{il} \phi A'_{jl}, \quad i = 1, \dots, q, \quad j = 1, \dots, p.$$

Pour la preuve de ce théorème, le lecteur peut se référer à [Baccelli et al., 1992, §4].

1.2.3 Résolution d'équations au point fixe dans un dioïde

Il s'agit de résoudre les équations suivantes :

$$\Pi(x) \oplus b = x, \tag{1.13}$$

$$\Pi(x) \wedge b = x. \tag{1.14}$$

Théorème 1.25. Soit \mathcal{D} un dioïde complet et $\Pi : \mathcal{D} \rightarrow \mathcal{D}$ une application semi-continue inférieurement (cf. définition 1.17). La plus petite solution de l'équation (1.13) est $\Pi^*(b)$, où

$$\Pi^* = \bigoplus_{n \in \mathbb{N}} \Pi^n \text{ avec } \Pi^0 = Id_{\mathcal{D}}, \quad \Pi^n = \underbrace{\Pi \circ \Pi \circ \dots \circ \Pi}_{n \text{ fois}}.$$

La preuve se fait en deux étapes : on doit vérifier que $\Pi^*(b)$ est solution de (1.13), ensuite on montre que toute solution de (1.13) est supérieure ou égale à $\Pi^*(b)$.

Preuve :

$$\begin{aligned} 1. \quad \Pi(\Pi^*(b)) \oplus b &= \Pi\left(\bigoplus_{n \in \mathbb{N}} \Pi^n(b)\right) \oplus b \\ &= \bigoplus_{n \in \mathbb{N}} \Pi(\Pi^n(b)) \oplus b \quad (\Pi \text{ est semi-continue inférieurement}) \\ &= \bigoplus_{n \geq 1} \Pi^n(b) \oplus Id_{\mathcal{D}}(b) = \Pi^*(b), \end{aligned}$$

donc $\Pi^*(b)$ est solution de l'équation (1.13).

2. $\forall x$ solution de l'équation (1.13), on a :

$$\begin{aligned} x = \Pi(x) \oplus b &= \Pi(\Pi(x) \oplus b) \oplus b = \Pi^2(x) \oplus \Pi(b) \oplus b \\ &\vdots \\ &= \Pi^n(x) \oplus \Pi^{n-1}(b) \oplus \dots \oplus \Pi(b) \oplus b \\ &\succeq \Pi^{n-1}(b) \oplus \dots \oplus \Pi(b) \oplus b, \quad n \geq 1. \end{aligned}$$

On déduit que toute solution de l'équation (1.13) est supérieure ou égale à $\Pi^*(b)$.

Exemple 1.26. (Equation $(a \otimes x) \oplus b = x$)

Dans cet exemple, on prend $\Pi = L_a : \mathcal{D} \mapsto \mathcal{D}; x \mapsto a \otimes x$. La plus petite solution de l'équation

$$x = (a \otimes x) \oplus b$$

est

$$\Pi^*(b) = a^* \otimes b \text{ avec } a^* = \bigoplus_{n \in \mathbb{N}} a^n.$$

Notation 2. L'opérateur "*)" est appelé étoile de Kleene. On note "+" l'opérateur défini dans un dioïde complet $(\mathcal{D}, \oplus, \otimes)$ par :

$$a \in \mathcal{D}, \quad a^+ = \bigoplus_{n \geq 1} a^n.$$

On a :

$$a^* = e \oplus a^+ \quad \text{et} \quad a^+ = a \otimes a^*.$$

Théorème 1.27. *Soit \mathcal{D} un dioïde complet et $\Pi : \mathcal{D} \rightarrow \mathcal{D}$ une application semi-continue supérieurement. La plus grande solution de l'équation (1.14) est $\Pi_*(b)$, où*

$$\Pi_* = \bigwedge_{n \in \mathbb{N}} \Pi^n.$$

Preuve :

- Comme Π est semi-continue supérieurement, on a :

$$\Pi(\Pi_*(b)) \wedge b = \Pi\left(\bigwedge_{n \in \mathbb{N}} \Pi^n(b)\right) \wedge b = \bigwedge_{n \in \mathbb{N}} \Pi(\Pi^n(b)) \wedge b = \bigwedge_{n \geq 1} \Pi^n(b) \wedge Id_{\mathcal{D}}(b) = \Pi_*(b).$$

Ceci montre que $\Pi_*(b)$ est solution de l'équation (1.14).

- $\forall x$ solution de (1.14), on a :

$$x = \Pi(x) \wedge b = \Pi(\Pi(x) \wedge b) \wedge b = \dots = \Pi^n(x) \wedge \Pi^{n-1}(b) \wedge \dots \wedge \Pi(b) \wedge b \preceq \Pi^{n-1}(b) \wedge \dots \wedge \Pi(b) \wedge b,$$

$$\forall n \geq 1.$$

Ceci montre que toute solution de l'équation (1.14) est inférieure ou égale à $\Pi_*(b)$.

1.3 Éléments de la théorie des systèmes linéaires

Après cette présentation sur les outils algébriques, la suite de ce chapitre est consacrée à la présentation des éléments de la théorie des systèmes linéaires dans les dioïdes.

1.3.1 Signaux et systèmes

Définition 1.28 (Signal). On appelle *signal* une application x de $\mathbb{R} \rightarrow (\mathcal{D}, \oplus, \otimes)$, $t \mapsto x(t)$. Sachant que \mathcal{D} est un dioïde complet.

Cet ensemble $\mathcal{D}^{\mathbb{R}}$ est muni d'une structure de moduloïde (cf. définition 1.12). Pour cela, on définit deux opérations :

- Une loi additive interne, notée \oplus , définie par :

$$\forall x_1, x_2 \in \mathcal{D}^{\mathbb{R}}, (x_1 \oplus x_2)(t) = x_1(t) \oplus x_2(t).$$

- Une loi externe, qui joue le rôle du produit d'un signal par un scalaire, notée " \cdot ", définie par :

$$\forall a \in \mathcal{D}, \forall x \in \mathcal{D}^{\mathbb{R}}, (a \cdot x)(t) = a \otimes x(t).$$

L'ensemble $\mathcal{D}^{\mathbb{R}}$ est composé de signaux. En effet :

- Un signal est croissant s'il satisfait

$$\forall t \in \mathbb{R}, \forall \tau \in \mathbb{N}, \quad x(t + \tau) \succeq x(t).$$

- Un signal est décroissant s'il satisfait

$$\forall t \in \mathbb{R}, \forall \tau \in \mathbb{N}, \quad x(t + \tau) \preceq x(t).$$

Selon la nature du modèle à étudier, deux types de signaux peuvent être considérés : les signaux discrets, utilisés lors de l'étude d'un modèle discret, et les signaux continus, utilisés pour l'étude d'un modèle continu.

Exemple 1.29. Le moduloïde $\overline{\mathbb{Z}}_{min}^{\mathbb{R}}$ (resp., $\overline{\mathbb{R}}_{min}^{\mathbb{R}}$) constitue un ensemble de signaux discrets (resp., continus).

Définition 1.30 (Produit de convolution). Le produit de convolution de deux signaux est noté \otimes et est défini par :

$$\forall x_1, x_2 \in \mathcal{D}^{\mathbb{R}}, \quad (x_1 \otimes x_2)(t) = \oint_{\tau \in \mathbb{R}} x_1(\tau) x_2(t - \tau) = \oint_{\tau \in \mathbb{R}} x_1(t - \tau) x_2(\tau),$$

où le symbole \oint correspond à \oplus et est utilisé dans cette expression afin de renforcer l'analogie avec la théorie des systèmes continus.

Exemple 1.31. Le produit de convolution de deux signaux de $\overline{\mathbb{R}}_{min}^{\mathbb{R}}$ satisfait la relation suivante :

$$\forall x_1, x_2 \in \overline{\mathbb{R}}_{min}^{\mathbb{R}}, \forall t \in \mathbb{R}, \quad (x_1 \otimes x_2)(t) = \min_{\tau \in \mathbb{R}} \{x_1(\tau) + x_2(t - \tau)\},$$

on parle de produit d'*inf*-convolution.

De manière duale, le produit de convolution de deux signaux de $\overline{\mathbb{R}}_{max}^{\mathbb{R}}$ donne

$$\forall x_1, x_2 \in \overline{\mathbb{R}}_{max}^{\mathbb{R}}, \forall t \in \mathbb{R}, \quad (x_1 \otimes x_2)(t) = \max_{\tau \in \mathbb{R}} \{x_1(\tau) + x_2(t - \tau)\},$$

on parle de produit de *sup*-convolution.

Définition 1.32 (Système linéaire). On appelle *système*, un opérateur, noté Σ , défini d'un ensemble de signaux vers un ensemble de signaux. Le système Σ est linéaire si :

$$\forall x_1, x_2 \in \mathcal{D}^{\mathbb{R}}, \quad \Sigma(x_1 \oplus x_2) = \Sigma(x_1) \oplus \Sigma(x_2) \quad (\text{propriété d'additivité}), \quad (1.15)$$

$$\forall x_1 \in \mathcal{D}^{\mathbb{R}}, \forall a \in \mathcal{D}, \quad \Sigma(a \otimes x_1) = a \otimes \Sigma(x_1) \quad (\text{propriété d'homogénéité}). \quad (1.16)$$

Définition 1.33 (Système continu). Un système Σ est *continu* si pour tout sous-ensemble fini ou infini $(x_i)_{i \in I}$ d'éléments de $\mathcal{D}^{\mathbb{R}}$, on a :

$$\Sigma\left(\bigoplus_{i \in I} x_i\right) = \bigoplus_{i \in I} \Sigma(x_i),$$

sachant que $(\mathcal{D}, \oplus, \otimes)$ est un dioïde complet.

Définition 1.34 (Système causal). Un système Σ est *causal* si pour toutes entrées x_1 et x_2 , on a :

$$\forall \tau \in \mathbb{Z}, x_1(t) = x_2(t) \text{ pour } t \leq \tau \Rightarrow [\Sigma(x_1)](t) = [\Sigma(x_2)](t) \text{ pour } t \leq \tau.$$

1.3.2 Systèmes élémentaires

On présente dans cette partie trois systèmes élémentaires linéaires :

- *Système gain* Γ^ν défini de $\overline{\mathbb{Z}}_{min}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}_{min}^{\mathbb{R}}$ par :

$$\forall t \in \mathbb{R}, \forall u, y \in \overline{\mathbb{Z}}_{min}^{\mathbb{R}}, \nu \in \overline{\mathbb{Z}}, y(t) = [\Gamma^\nu(u)](t) = \nu \otimes u(t).$$



Figure 1.1 – Système Γ^ν .

- *Système retard* Δ^τ défini de $\overline{\mathbb{Z}}_{min}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}_{min}^{\mathbb{R}}$ par :

$$\forall t \in \mathbb{R}, \forall u, y \in \overline{\mathbb{Z}}_{min}^{\mathbb{R}}, \tau \in \overline{\mathbb{Z}}, y(t) = [\Delta^\tau(u)](t) = u(t - \tau).$$



Figure 1.2 – Système Δ^τ .

- *Limiteur de flux* Φ_v défini de $\overline{\mathbb{R}}_{min}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}_{min}^{\mathbb{R}}$ par :

$$\forall t \in \mathbb{R}, \forall u, y \in \overline{\mathbb{R}}_{min}^{\mathbb{R}}, v \in \mathbb{R}^+, y(t) = [\Phi_v(u)](t) = \inf_{h \geq 0} (u(t - h) + v \times h).$$

Figure 1.3 – Système Φ_v .

Définition 1.35 (Système stationnaire). Un système Σ est *stationnaire* (ou invariant dans le temps) si

$$\forall x \in \mathcal{D}^{\mathbb{Z}}, \forall \tau \in \mathbb{Z}, \Sigma(\Delta^\tau(x)) = \Delta^\tau \Sigma(x),$$

où Δ^τ est le système élémentaire linéaire retard défini précédemment.

1.3.3 Opérations sur les systèmes

Il existe trois opérations élémentaires permettant de mettre en relation des systèmes : la composition série, la composition parallèle et la mise en feedback.

- **Composition série** $\Sigma = \Sigma_1 \otimes \Sigma_2$ est définie par :

$$[\Sigma(x)](t) = [\Sigma_2(\Sigma_1(x))](t).$$



Figure 1.4 – Composition série.

- **Composition parallèle** $\Sigma = \Sigma_1 \oplus \Sigma_2$ est définie par :

$$[\Sigma(x)](t) = [\Sigma_1(x)](t) \oplus [\Sigma_2(x)](t).$$

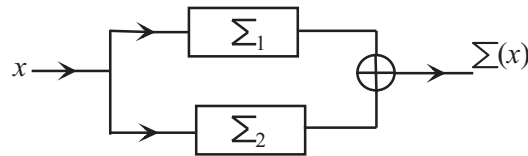


Figure 1.5 – Composition parallèle.

- **Mise en feedback** On considère le schéma suivant, où Σ_1 et Σ_2 sont deux opérateurs linéaires.

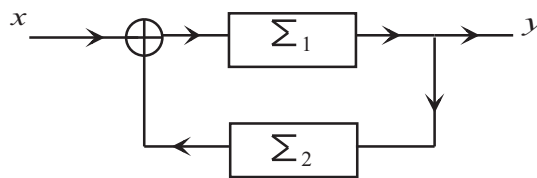


Figure 1.6 – Mise en feedback.

La relation entre la sortie y et l'entrée x satisfait :

$$y = \Sigma_1(\Sigma_2(y) \oplus x) = \Sigma_1(\Sigma_2(y)) \oplus \Sigma_1(x).$$

D'après le théorème 1.25, la plus petite solution existe et est donnée par :

$$(\Sigma_1 \Sigma_2)^* \Sigma_1(x) = \left(\bigoplus_{n \in \mathbb{N}} (\Sigma_1 \Sigma_2)^n \right) \Sigma_1(x).$$

1.4 Conclusion

Dans ce premier chapitre nous avons présenté les outils mathématiques utilisés par la suite. Après un bref rappel sur la structure algébrique des dioïdes, la théorie de la résiduation a été présentée. Cette théorie permet de donner une alternative à la notion d'inverse pour les applications définies sur des ensembles ordonnés, elle sera utilisée pour calculer la commande des systèmes n'admettant pas une représentation $(min, +)$ linéaire présentée au chapitre 3. Dans la dernière partie de ce chapitre, nous

avons donné un ensemble de définitions sur la théorie des systèmes utiles pour aborder, dans le chapitre suivant, la représentation de certains systèmes dynamiques, modélisables dans un dioïde d'opérateurs.

Modélisation des graphes d'événements temporisés discrets, continus, hybrides avec multiplieurs

Dans la littérature, de nombreuses classes de modèles sont proposées pour l'étude des systèmes dynamiques. Le choix d'un type de modèle est fonction des objectifs recherchés et de la nature du système considéré. Dans le cas des systèmes à événements discrets (SED), plusieurs concepts de modélisation ont été élaborés : par exemple, les chaînes de Markov [Baynat, 2000], les files d'attente [Baynat, 2000], les réseaux de Petri stochastiques pour les systèmes stochastiques [Ciardo et al., 1997], ou les réseaux de Petri déterministes pour les systèmes déterministes [Murata, 1989, Proth and Xie, 1995]. De façon analogue à cette classe de systèmes, plusieurs outils ont été conçus pour la modélisation des systèmes continus : les modèles sous forme d'équations différentielles et algébriques, les automates ou encore les réseaux de Petri continus. Si l'utilisation des modèles discrets et des modèles continus suffit pour modéliser certains systèmes, il s'avère que dans de nombreux cas il faut combiner les caractères continu et discret dans un même modèle, pour modéliser des systèmes, ceux-ci sont dits hybrides. Dans cette classe de systèmes, on retrouve les modèles de type réseaux de Petri hybrides et automates hybrides.

Les réseaux de Petri (RdP) sont des modèles de représentation souvent utilisés pour les SED, ils sont largement utilisés et permettent de modéliser, d'évaluer voire de piloter des systèmes dynamiques.

Certains systèmes mettant uniquement en jeu des phénomènes de synchronisation et de saturation peuvent être modélisés par une classe des réseaux de Petri, appelée classe des graphes d'événements tem-

porisés (GET). Ces derniers admettent une représentation linéaire sur une structure algébrique de dioïde. Cette représentation est bien adaptée pour aborder, par exemple, les problèmes de commande ou d'évaluation de performances. Néanmoins, lorsque la taille du système est importante (système complexe, nombre important d'entités), les techniques développées dans le cadre des systèmes discrets atteignent leur limite. L'utilisation des GET à arcs pondérés, [Teruel et al., 1992, Chao et al., 1993, Munier, 1993], encore appelés graphes d'événements temporisés avec multiplieurs, permet de réduire la taille du modèle.

L'objectif de ce chapitre est d'établir un modèle mathématique qui reflète le comportement dynamique des modèles considérés dans ce rapport, à savoir, les graphes d'événements temporisés **discrets** avec multiplieurs, notés GETDM, les graphes d'événements temporisés **continus** avec multiplieurs, notés GETCM, et les graphes d'événements temporisés **hybrides** avec multiplieurs, notés GETHM.

Ce chapitre est organisé en quatre parties :

- La section 2.1 fait un rappel sur la modélisation des GET discrets dans le dioïde Z_{min} .
- La classe des GETDM est introduite dans la section 2.2. Il est montré que la présence des poids sur les arcs fait qu'un GETDM n'admet pas une représentation $(min, +)$ linéaire. Comme alternative, en vue de modéliser ces graphes, nous proposons une approche opérationnelle basée sur un dioïde d'opérateurs.
- Afin d'aborder la classe des GETHM, les graphes d'événements temporisés avec multiplieurs continus, notés GETCM, seront présentés dans la section 2.3. En utilisant l'approche opérationnelle, un modèle décrivant l'évolution dynamique de ces graphes est établi.
- La section 2.4 est consacrée à la modélisation des GETHM.

Ce chapitre rassemble les contributions de [Hamaci et al., 2004a, Hamaci et al., 2005a].

2.1 Graphes d'événements temporisés discrets

Les réseaux de Petri (RdP) sont un outil graphique et mathématique, introduit en 1962 par Carl Adam Petri [Petri, 1962]. Ils permettent la modélisation d'un grand nombre de systèmes à événements discrets. Ils sont particulièrement adaptés à l'étude des processus complexes mettant en jeu des propriétés de synchronisme et de partage de ressource.

L'étude du comportement au cours du temps des systèmes dynamiques, notamment l'évaluation de

leur performance (temps de cycle, ...), a conduit à introduire la notion de temps dans les modèles de type RdP. Plusieurs modèles RdP intégrant le temps ont été proposés. Ces modèles peuvent être regroupés en deux classes : les modèles déterministes et les modèles stochastiques. Les premiers considèrent des valeurs déterministes pour les durées d'activité, alors que les seconds considèrent des valeurs probabilistes. Parmi les RdP temporisés existants, on peut citer : les RdP temporels [Merlin, 1979] qui associent un intervalle temporel à chaque transition ou à chaque place, les RdP T-temporisés [David and Alla, 1992] qui associent une constante positive (appelée durée de franchissement de la transition) à chaque transition et les RdP P-temporisés [David and Alla, 1992, Hillion, 1989] qui associent une constante positive (appelée temps de séjour dans la place) à chaque place du graphe. Il a été montré que les RdP P-temporisés peuvent se ramener à des réseaux T-temporisés et inversement [Murata, 1989]. Le choix de l'un de ces modèles temporisés est donc arbitraire. Par souci d'homogénéité avec la littérature réalisée sur l'algèbre des dioïdes, nous considérons par la suite que des RdP P-temporisés.

2.1.1 Réseaux de Petri : définitions et notations

Un RdP est un graphe biparti¹ défini par un 5-tuplet $(\mathcal{P}, \mathcal{T}, M, m, \tau)$ où :

1. \mathcal{P} (resp., \mathcal{T}) est un ensemble non vide et fini de places (resp., transitions) ;
2. $M \in \mathbb{N}^{(\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})}$ définit le poids des arcs : pour $p \in \mathcal{P}$ et $q \in \mathcal{T}$, le multiplicateur M_{pq} (resp., M_{qp}) spécifie le poids de l'arc allant de la transition n_q vers la place p (resp., de la place p vers la transition n_q) ;
3. $m \in \mathbb{N}^{\mathcal{P}}$ définit le marquage initial : m_p désigne le nombre de jetons initialement contenus dans la place p ;
4. $\tau \in \mathbb{N}^{\mathcal{P}}$ définit les temporisations associées aux places : τ_p désigne la durée minimale de séjour d'un jeton dans la place p .

Une transition sans place en amont est dite *transition source*, une transition sans place en aval est dite *transition puit*.

La figure 2.1 représente un RdP ordinaire (tous les poids sont égaux à un), dont les places sont représentées par des cercles, et les transitions par des barres.

¹Un graphe est dit *biparti* s'il n'y a pas d'arc joignant une place à une place, ni d'arc joignant une transition à une transition.

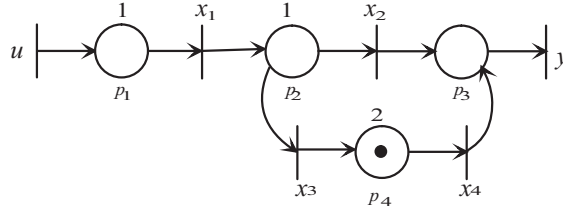


Figure 2.1 – Représentation graphique d'un RdP.

L'évolution au cours du temps des jetons dans un RdP se fait selon le processus de franchissement de transitions décrit ci-après. Etant donné un RdP et un marquage m , on dit que la transition $x_q \in \mathcal{T}$ est *franchissable* pour le marquage m si l'on a :

$$\forall p \in \mathcal{P}, m(p) \geq M_{qp}.$$

Lorsque cette condition est satisfaite, le *franchissement* de la transition x_q conduit à nouveau marquage m' défini par :

$$m'(p) = m(p) - M_{qp} + M_{pq'}.$$

Dans le cas du RdP de la figure 2.1, en partant du marquage $m_0^t = (0, 0, 0, 1)$ ($m(p_1) = 0$, $m(p_2) = 0$, $m(p_3) = 0$, $m(p_4) = 1$), on atteint, par franchissement de la transition x_4 , le marquage $m_1^t = (0, 0, 1, 0)$.

Plus généralement, pour un RdP, en posant $[W^-] = M_{qp}$ (élément de la matrice d'incidence avant), $[W^+] = M_{pq'}$ (élément de la matrice d'incidence arrière), $W = W^+ - W^-$ (la matrice d'incidence) et en considérant S une séquence de franchissement réalisable à partir d'un marquage m_i , on a l'équation fondamentale² suivante :

$$m_k = m_i + W \times \bar{S}, \tag{2.1}$$

où \bar{S} est le vecteur caractéristique de la séquence de franchissement S , c'est-à-dire un vecteur de dimension égale à $|\mathcal{T}|$; \bar{S}_q correspond au nombre de franchissement de la transition x_q dans la séquence S . Dans la figure 2.1 où la séquence de tir était $S = \{x_4\}$, le vecteur caractéristique est égal à $\bar{S}^t = (0, 0, 0, 1)$.

2.1.2 Invariants d'un réseau de Petri

Il existe deux types d'invariants (aussi appelés semiflots) dans un RdP [David and Alla, 1992]: les *invariants de marquage* (notés également P-invariant) et les *invariants de franchissement* (notés également

²L'équation fondamentale traduit la dynamique de fonctionnement du RdP.

T-invariant).

Définition 2.1 (P-invariant). Les invariants de marquage, appelés *P-invariant*, ou encore *P-semiflotts*, illustrent la conservation du nombre de jetons dans un sous ensemble de places du RdP. Un vecteur, noté Y , de dimension égale au nombre de places du RdP est un P-invariant si, et seulement s'il vérifie l'équation suivante :

$$Y^t \times W = \vec{0}, \quad Y \neq \vec{0}, \quad (2.2)$$

où W correspond à la matrice d'incidence du RdP.

D'après l'équation (2.1), on déduit que si Y est un P-invariant, alors pour un marquage donné, noté m_i , obtenu à partir d'un marquage initial m_0 , on a :

$$Y^t \times m_i = Y^t \times m_0 = k, \quad k \in \mathbb{N}^*.$$

Cette équation représente un invariant de marquage, elle signifie que si Y est un P-invariant du RdP alors la transposé du vecteur Y pondéré par le vecteur marquage m_i du RdP est une constante entière quel que soit le marquage m_i atteignable à partir du marquage initial m_0 . L'ensemble des places pour lesquelles la composante associée dans le P-invariant est non nulle, est appelée la *composante conservative* du RdP.

Définition 2.2 (T-invariant). Un vecteur non nul d'entiers θ de dimension $|\mathcal{T}| \times 1$ est un *T-invariant*, ou encore *T-semiflotts*, du RdP si, et seulement s'il vérifie l'équation suivante :

$$W \times \theta = \vec{0}. \quad (2.3)$$

D'après l'équation (2.1), le franchissement à partir d'un marquage m_i d'une séquence dont le vecteur caractéristique est θ ramène le graphe au même marquage $m_k = m_i$.

L'ensemble des transitions pour lesquelles la composante associée dans le T-invariant est non nulle est appelé le *support* du T-invariant. Un T-invariant correspondant à une séquence de franchissement réalisable est appelé *composante répétitive*.

Définition 2.3 (Consistance d'un réseau de Petri). Un RdP est dit *consistant* s'il possède un T-invariant θ couvrant toutes les transitions du réseau. Un RdP qui possède cette propriété est dit *répétitif* ou *réinitialisable*. Le réseau atteint un régime périodique dès lors qu'il existe une séquence de franchissement réalisable avec θ , comme vecteur caractéristique.

Définition 2.4 (Réseau de Petri conservatif). Un RdP est dit *conservatif* si toutes les places du graphe forment une composante conservative.

Définition 2.5.

- Un T-invariant θ est *élémentaire* s'il est *canonique*, c'est-à-dire si le plus grand commun diviseur de ses composantes est égal à un.
- Un T-invariant θ a un support *minimal* s'il n'existe pas un autre T-invariant θ' tel que :

$$\{q \in \mathcal{T} | \theta'(q) > 0\} \subset \{q \in \mathcal{T} | \theta(q) > 0\}.$$

- Un réseau de Petri consistant est dit fortement connexe si, et seulement s'il est conservatif.

Remarque 2.6.

- Toute combinaison de T-invariants (resp., P-invariants) est un T-invariant (resp., P-invariant).
- Un réseau de Petri consistant a un unique T-invariant élémentaire.
- Le produit des poids d'un circuit d'un réseau de Petri conservatif est égal à un.

Exemple 2.7. Considérons le RdP de la figure 2.2 qui suit, un P-invariant est le vecteur Y vérifiant :

$$Y^t \times W = (y_1, y_2) \times \begin{pmatrix} -3 & 2 \\ 3 & -2 \end{pmatrix} = (0 \ 0).$$

Il existe un P-invariant élémentaire : $Y^t = (1, 1)$. L'invariant de marquage est : $1 \times m(p_1) + 1 \times m(p_2) = 4$. Ceci signifie que les places p_1 et p_2 forment une composante conservative du graphe.

Le RdP de la figure 2.2 admet un T-invariant vérifiant :

$$W \times \theta = \begin{pmatrix} -3 & 2 \\ 3 & -2 \end{pmatrix} \times \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \vec{0}.$$

Il existe un T-invariant minimal : $\theta^t = (2, 3)$. Le support de θ est $\{n_1, n_2\}$.

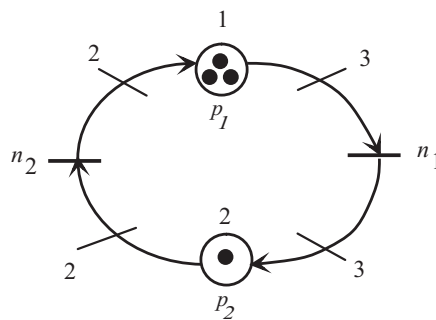


Figure 2.2 – Réseau de Petri à arcs pondérés.

Définition 2.8 (Nombre de franchissement dans un RdP). Un RdP est dit k -serveur si, pour chaque transition, il ne peut y avoir au maximum que k franchissements simultanés. Une boucle de ré-entrance implicite contenant k jetons est alors associée à chaque transition du graphe.

- Dans le cas où $k = 1$, le RdP est dit *mono-serveur*.
- Dans le cas où le nombre de franchissement de transitions n'est pas limité, le RdP est dit *infini-serveur*.

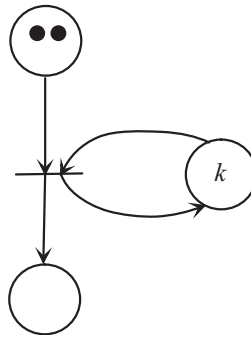


Figure 2.3 – Boucle de ré-entrance.

Dans la suite, on note $\bullet q$ (resp., $q\bullet$) l'ensemble des places situées immédiatement en amont (resp., en aval) de la transition n_q . Similairement, $\bullet p$ (resp., $p\bullet$) représente l'ensemble des transitions situées immédiatement en amont (resp., en aval) de la place p .

2.1.3 Graphes d'événements temporisés

Suite à cette brève présentation de RdP, notre intérêt va maintenant se porter sur une structure particulière des RdP qui permet la modélisation des systèmes où interviennent des phénomènes de synchronisation et de délai (les phénomènes de conflits étant supposés résolus au préalable).

Définition 2.9. Un graphe d'événements temporisé ordinaire (GET) est un RdP tel que toute place a exactement une transition en amont et une transition en aval.

Pour illustrer ce type de modèles temporisés, nous proposons l'exemple d'un graphe d'événements temporisé représentant le fonctionnement d'un atelier de coupe de bois.

Exemple 2.10. (Atelier de coupe de bois)

Le GET de la figure 2.4 modélise un atelier constitué d'une machine de coupe de bois. Quand une pièce arrive et que la machine de coupe est disponible, cette dernière est traitée (découpée).

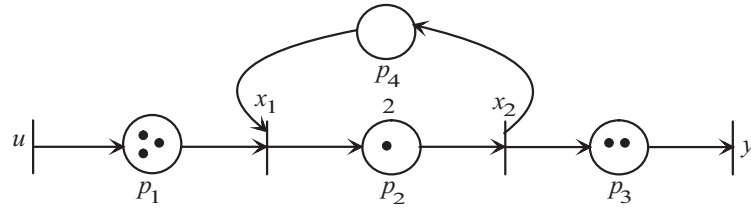


Figure 2.4 – Modèle GET d'un atelier de coupe de bois.

Le nombre de jetons dans une place s'interprète comme le nombre de *ressources* disponibles. Par exemple, le nombre de jetons dans la place p_1 correspond au nombre de pièces en attente d'être traitées par la machine de coupe, c'est à dire le nombre de "ressources" qui vont être consommées. Un jeton dans la place p_2 indique qu'une pièce est en train d'être traitée par la machine de coupe. La durée de ce traitement est de $2 u.t$ (unités de temps). La fin de la coupe par la machine se traduit par l'enlèvement d'un jeton dans la place p_2 et par l'ajout d'un jeton dans la place p_3 et dans la place p_4 . Le nombre de jetons dans la place p_3 correspond au nombre de pièces qui ont été traitées. La présence d'un jeton dans la place p_4 indique que la machine est disponible.

2.1.4 Propriétés des graphes d'événements temporisés

Nous rappelons brièvement quelques caractéristiques des graphes d'événements temporisés au travers des propositions suivantes.

Proposition 2.11. *Le nombre de jetons d'un circuit est constant dans un graphe d'événements temporisé*

Démonstration. Rappelons qu'un circuit est un chemin qui commence et se termine au même sommet. Alors, si une transition franchissable appartenant à un circuit est franchie, son franchissement prend un jeton dans la place en amont de la transition pour le remettre dans la place en aval de la transition. L'opération de franchissement d'un circuit laisse donc invariant le nombre de jetons du circuit.

Proposition 2.12. *Un graphe d'événements temporisé est vivant si, et seulement si, tout circuit élémentaire contient au moins une place initialement marquée.*

Démonstration. Supposons qu'aucune place d'un circuit élémentaire (c'est-à-dire que chaque sommet n'est rencontré qu'une seule fois dans le circuit) d'un GET ne soit initialement marquée. En référence à la proposition 2.11, ce circuit ne contiendra jamais de jeton et donc aucune de ses transitions est franchissable : le GET n'est donc pas vivant.

Inversement, dans un GET non vivant, une transition qui n'est jamais franchie possède obligatoirement

en amont une transition également jamais franchie. En remontant ainsi d'une transition jamais franchie à une autre située en amont, on aboutit inéluctablement à une transition appartenant à un circuit, circuit qui est donc nécessairement sans jeton.

2.1.5 Représentation d'état des graphes d'événements temporisés

A la différence du modèle d'état classiquement associé à un RdP (cf. équation 2.1), l'état considéré est associé non plus aux places d'un GET mais à ses transitions. Les variables d'état considérées sont de type compteurs d'événements. De telles variables permettent d'aboutir à une représentation linéaire de ces modèles dans les *dioides*. On a privilégié la représentation *compteur* à celle de *dateur* (cf. remarque 2.18) du fait des dioides inspirés de l'algèbre $(\min, +)$ qui seront proposés par la suite pour représenter des graphes d'événements temporisés avec multiplieurs.

Définition 2.13 (Compteur). Un *compteur* associé à une transition x_q est une application croissante, notée $x_q(t)$, de $\mathbb{R} \rightarrow \mathbb{Z} \cup \{\pm\infty\}$, $t \mapsto x_q(t)$, où $x_q(t) \in \mathbb{Z} \cup \{\pm\infty\}$ correspond au nombre de tirs de la transition x_q ayant lieu jusqu'à la date t .

Remarque 2.14. Deux types de compteur sont considérés dans ce document, selon que l'on s'intéresse à un modèle discret ou continu.

Le vecteur $X(t) = (x_1(t) \dots x_{|\mathcal{T}|}(t))^t$ définit le vecteur d'état pour un GET. A la différence du marquage, ces variables d'état ont une propriété de monotonie :

$$\forall t \in \mathbb{R}, \forall \alpha \geq 0, \forall x_q \in \mathcal{T}, x_q(t + \alpha) \geq x_q(t).$$

Remarque 2.15. Comme le franchissement des transitions est ici supposé être instantané, il n'y a pas à formuler d'hypothèses sur les tirs des transitions, leur fonctionnement est évidemment de type FIFO³(First-In, First-Out).

Pour illustrer l'évolution d'un compteur associé à la transition x_q d'un GET, considérons le GET élémentaire suivant :

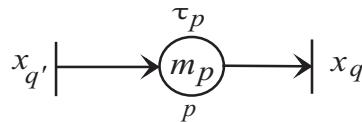


Figure 2.5 – GET élémentaire.

³Le k -ième jeton rentré dans une place sera également le k -ième jeton à contribuer au franchissement de la transition en aval à cette place.

Sous l'hypothèse de fonctionnement au plus tôt, qui est un comportement limite du GET, l'évolution du compteur $x_q(t)$, associé à la transition x_q du GET élémentaire de la figure 2.5, est exprimée par l'équation suivante :

$$x_q(t) = m_p + x_{q'}(t - \tau_p). \quad (2.4)$$

Dans un GET, le phénomène de synchronisation se produit lorsque plusieurs arcs convergent sur une même transition (cf. figure 2.6).

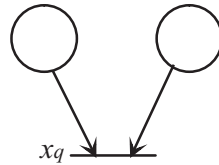


Figure 2.6 – Phénomène de synchronisation dans un GET.

En général, l'évolution d'un compteur associé à une transition de synchronisation d'un GET est donnée par l'équation suivante :

$$x_q(t) = \min_{p \in \bullet q, q' \in \bullet p} (m_p + x_{q'}(t - \tau_p)). \quad (2.5)$$

On remarque que cette équation est non linéaire dans l'algèbre usuelle de par la présence de l'opérateur min qui modélise le phénomène de synchronisation au niveau de la transition x_q . Par contre, elle se décrit de façon linéaire dans le dioïde $\overline{\mathbb{Z}}_{min}$. Le compteur x_q vérifie l'équation récurrente suivante :

$$x_q(t) = \bigoplus_{p \in \bullet q, q' \in \bullet p} (m_p \otimes x_{q'}(t - \tau_p)). \quad (2.6)$$

L'opération \bigoplus du dioïde $\overline{\mathbb{Z}}_{min}$ représente un phénomène de synchronisation au niveau de la transition x_q , et l'opération \otimes représente un décalage *événementiel* du compteur $x_q(t)$ par rapport à tout compteur associé à une transition en amont de la transition x_q .

D'une manière générale, les équations modélisant le fonctionnement au plus tôt des GET ont la forme suivante dans le dioïde $\overline{\mathbb{Z}}_{min}$:

$$\begin{cases} X(t) = \bigoplus_{i=0}^T A_i X(t-i) \oplus \bigoplus_{i=0}^T B_i U(t-i), \\ Y(t) = \bigoplus_{i=0}^T C_i X(t-i), \end{cases} \quad (2.7)$$

avec $T = \max_{p_i \in \mathcal{P}}(\tau(p_i))$. $X(t)$ est le vecteur d'état, composé des compteurs associés aux transitions internes du graphe, $U(t)$ et $Y(t)$ sont respectivement les vecteurs d'entrées (commandes) et de sorties, composés des transitions sources, respectivement puits, du graphe.

Exemple 2.16. Dans le dioïde $\overline{\mathbb{Z}}_{min}$, les variables d'état du GET de la figure 2.7 sont régies par les relations aux compteurs suivantes :

$$\begin{cases} x_1(t) &= 1 \otimes x_1(t-1) \oplus 2 \otimes x_2(t) \oplus u(t), \\ x_2(t) &= x_1(t-2), \\ y(t) &= x_2(t). \end{cases}$$

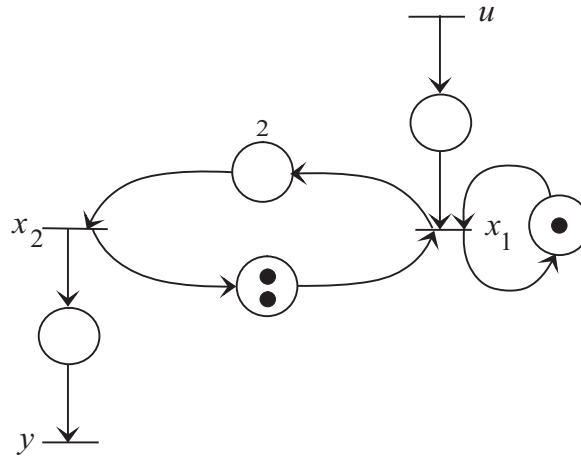


Figure 2.7 – Exemple d'un GET.

Il en résulte la représentation d'état suivante :

$$\begin{cases} X(t) = \begin{pmatrix} \varepsilon & 2 \\ \varepsilon & \varepsilon \end{pmatrix} X(t) \oplus \begin{pmatrix} 1 & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix} X(t-1) \oplus \begin{pmatrix} \varepsilon & \varepsilon \\ e & \varepsilon \end{pmatrix} X(t-2) \oplus \begin{pmatrix} e \\ \varepsilon \end{pmatrix} U(t), \\ Y(t) = \begin{pmatrix} \varepsilon & e \end{pmatrix} X(t), \end{cases}$$

avec $X = (x_1 \ x_2)^t$, $U = u$ et $Y = y$.

2.1.6 Forme récurrente explicite de la représentation d'état

La représentation d'état sous une forme ARMA (Auto Regressive Moving Average) donnée par la représentation (2.7) peut se ramener, moyennant une extension de la dimension du vecteur d'état, à une forme récurrente markovienne du type :

$$\begin{cases} X(t) = AX(t-1) \oplus BU(t), \\ Y(t) = CX(t). \end{cases} \quad (2.8)$$

Le GET équivalent auquel va correspondre une représentation de la forme (2.8) est tel que :

- la temporisation d'une place située entre deux transitions internes doit être égale à un ;
- la temporisation d'une place située entre une transition source et une transition interne, ou entre une transition interne et une transition puits, doit être nulle.

Exemple 2.17. La figure 2.8 représente un GET dont le comportement entrée-sortie est équivalent à celui de la figure 2.7. En introduisant un nouvel état x'_1 , on aboutit à la représentation d'état suivante sur le dioïde $\overline{\mathbb{Z}}_{\min}$:

$$\begin{cases} X(t) = \begin{pmatrix} \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} X(t) \oplus \begin{pmatrix} 1 & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} X(t-1) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} U(t), \\ Y(t) = (\varepsilon \ \varepsilon \ e) X(t), \end{cases}$$

avec $X = (x_1 \ x'_1 \ x_2)^t, U = u$ et $Y = y$.

On obtient, après utilisation du théorème 1.25 pour supprimer la partie implicite de l'équation d'état, la forme récurrente markovienne suivante :

$$\begin{cases} X(t) = \begin{pmatrix} 1 & 2 & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} X(t-1) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} U(t), \\ Y(t) = (\varepsilon \ \varepsilon \ e) X(t), \end{cases}$$

sachant que :

$$A_0^* = \begin{pmatrix} \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}^* = \begin{pmatrix} e & \varepsilon & 2 \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{pmatrix}, \quad A_0^* A_1 = \begin{pmatrix} 1 & 2 & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} \quad \text{et} \quad A_0^* B = \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix}.$$

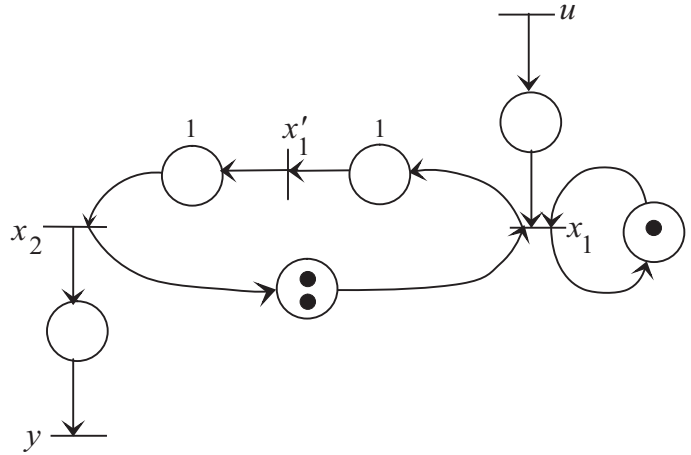


Figure 2.8 – GET équivalent à celui de la figure 2.7 en vue d'une représentation markovienne.

Remarque 2.18. Equations aux dateurs.

Au lieu de considérer les nombres cumulés de tirs d'une transition d'un GET à chaque instant, on s'intéresse aux différentes dates de tirs de cette transition. A chaque transition x_q , on associe l'application $x_q : \mathbb{Z} \rightarrow \overline{\mathbb{Z}}_{max}$, $k \mapsto x_q(k)$ où $x_q(k)$ désigne la date du k -ième tir de la transition x_q . Cette application est non décroissante et est appelée *dateur*.

Par un raisonnement analogue à celui effectué pour les compteurs, on montre que la description du comportement d'un GET peut se représenter *via* des équations $(max, +)$ linéaires.

Le graphe de la figure 2.7 peut être modélisé par les équations $(max, +)$ linéaires suivantes :

$$\begin{cases} x_1(k) &= 1 \otimes x_1(k-1) \oplus x_2(k-2) \oplus u(k), \\ x_2(k) &= 2 \otimes x_1(k), \\ y(k) &= x_2(k). \end{cases}$$

De manière analogue aux équations aux compteurs, on peut disposer d'une forme markovienne aux dateurs telle que :

- le marquage d'une place située entre deux transitions internes doit être égal à un ;
- le marquage d'une place située entre une transition source et une transition interne, ou entre une transition interne et une transition puits, doit être nulle.

2.1.7 Conditions initiales des graphes d'événements temporisés

Définition 2.19 (Conditions initiales canoniques). On dira que les conditions initiales d'un GET sont *canoniques* si tous les jetons du marquage initial sont considérés comme disponibles depuis l'instant $-\infty$.

Considérer des conditions initiales canoniques, implique de prendre pour hypothèse que le marquage initial du graphe n'est pas nécessairement égal au marquage à l'instant initial d'observation, (égal à 0 par convention). En effet, si les jetons initiaux sont supposés disponibles depuis $-\infty$, ils sont susceptibles d'avoir validé et tiré des transitions également à $-\infty$, aussi le marquage peut avoir évolué avant l'instant initial d'observation.

Exemple 2.20. La figure 2.9 représente un GET. Si on considère des conditions initiales canoniques, le marquage à l'instant initial d'observation est différent du marquage initial du graphe. En effet, le jeton contenu dans la place p_3 a validé et tiré la transition x_2 à $t = -\infty$. Un jeton a alors été ajouté dans les places p_2 et p_4 . Le jeton ajouté dans la place p_4 a de même validé et tiré la transition y à $t = -\infty$. En revanche, le jeton ajouté dans la place p_2 , ainsi que celui du marquage initial, ont validé la transition x_1 , sans pouvoir la tirer, car il n'y avait pas initialement, à $t = -\infty$, de jeton dans la place p_1 . Finalement, le marquage, après qu'il ait évolué librement depuis l'instant $t = -\infty$, à l'instant d'observation est celui représenté sur la figure 2.10. De plus, à $t = 0$, les transitions x_2 et y ont été déjà tirées une fois chacune.

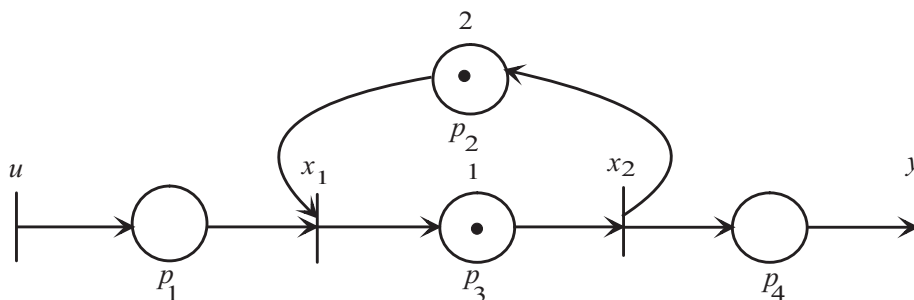


Figure 2.9 – Marquage initial ($t = -\infty$).

Définition 2.21 (Conditions initiales faiblement compatibles non canoniques). On dira que les conditions initiales d'un GET sont faiblement compatibles si tout jeton du marquage initial est disponible qu'à partir de $t = 0$. Pour $t < 0$, le graphe est considéré comme étant "gelé".

Les jetons visibles à l'instant initial d'observation $t = 0$ sont alors ceux du marquage initial. Ces

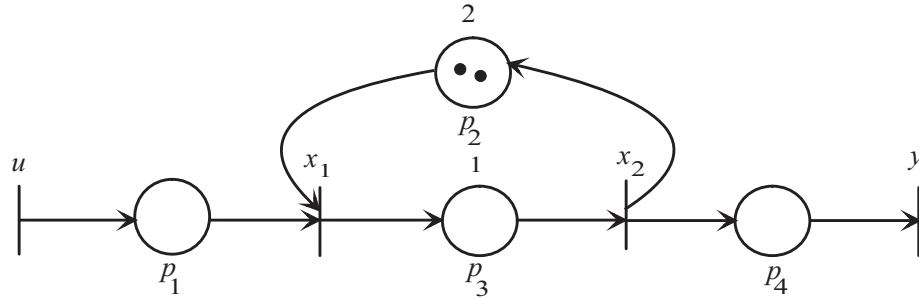


Figure 2.10 – Marquage à l’instant initial d’observation ($t=0$).

jetons sont supposés avoir été introduits dans le graphe avant l’instant 0, de telle sorte qu’à $t = 0$ un jeton du marquage initial est :

- soit en train de subir la temporisation de la place dans laquelle il a été introduit,
- soit tout juste disponible pour le tir de la transition aval.

2.2 Graphes d’événements temporisés discrets avec multiplieurs

2.2.1 Présentation des graphes d’événements temporisés discrets avec multiplieurs

De nombreux systèmes sont tels que les entités qui les traversent (par exemple, des pièces pour un système de production) subissent des phases d’assemblage ou de désassemblage. La classe des GET avec multiplieurs permet une modélisation simple et intuitive de tels systèmes.

Définition 2.22 (Graphe d’événements temporisé discret avec multiplieurs). Un graphe d’événements temporisé discret avec multiplieurs (GETDM) est un GET dans lequel des poids (nombres entiers strictement positifs) sont associés aux arcs. Ces graphes sont également appelés graphes d’événements généralisés ou *valués*.

Dans ce type de graphe, dont un exemple élémentaire est donné dans la figure 2.11, le poids $M_{pq'}$ rattaché à l’arc $n_{q'} \rightarrow p$ est tel que lors d’un franchissement de la transition $n_{q'}$, $M_{pq'}$ jetons sont ajoutés à la place p . Le poids M_{qp} associé à l’arc $p \rightarrow n_q$ implique que la transition n_q ne sera validée que si la place p contient au moins M_{qp} jetons. Lors du franchissement de cette transition, après un séjour de τ_p unités de temps dans la place p , M_{qp} jetons y seront retirés.

La figure 2.12 représente un GETDM tel que l’arc $p_1 \rightarrow n_1$ a un poids de 3, et l’arc $n_1 \rightarrow p_4$ a un poids de 2. Les arcs dont le poids n’est pas explicitement spécifié ont un poids égal à 1. La transition n_1 est validée puisque la place p_1 contient 3 jetons et la place p_2 contient plus d’un jeton. Après un séjour

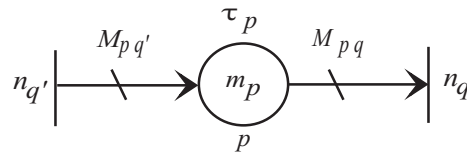


Figure 2.11 – GETDM élémentaire.

de deux unités de temps des jetons dans la place p_1 , le franchissement de n_1 consiste à retirer trois jetons de p_1 et un jeton de p_2 , et à ajouter un jeton à p_3 et deux jetons à p_4 . On obtient le marquage de la figure 2.12.b.

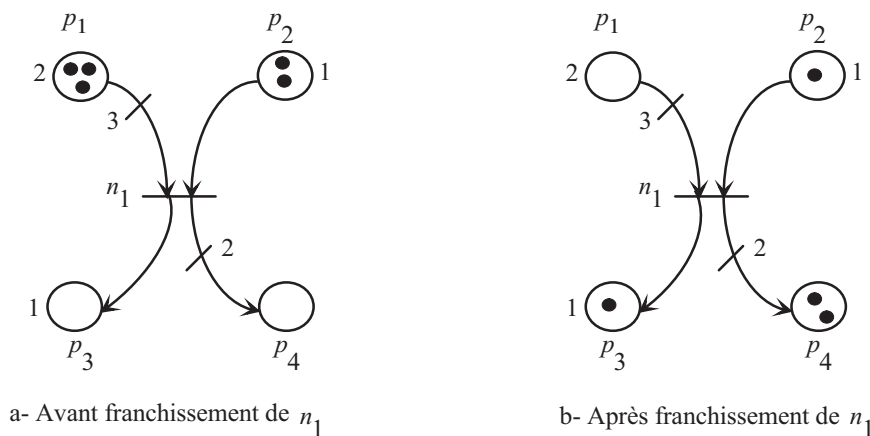


Figure 2.12 – évolution du marquage.

Définition 2.23 (Chemin élémentaire [Gondran and Minoux, 1985]). Un chemin est une succession de sommets (*cf.* figure 2.11) tel que le sommet de sortie d'un arc du chemin est le sommet d'entrée de l'arc qui le suit dans le chemin. Un chemin est dit *élémentaire* s'il contient au plus une fois chaque sommet du graphe.

Définition 2.24 (Circuit élémentaire [Gondran and Minoux, 1985]). Un circuit élémentaire, noté ρ , est un chemin élémentaire fermé tel que sa transition de fin, n_q , est aussi sa transition de début (*cf.* figure 2.13).

Définition 2.25 (Gain d'un circuit [Teruel et al., 1992]). Le gain d'un circuit ρ , noté $G(\rho)$, est le produit du quotient des poids des arcs entrants par le produit des poids des arcs sortants de toutes les places

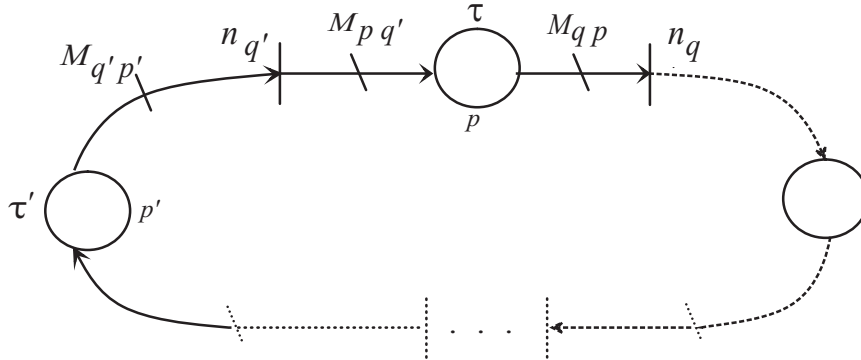


Figure 2.13 – Circuit élémentaire.

du circuit ϱ . Formellement, ce gain est défini comme suit :

$$G(\varrho) = \prod_{P_i \in \varrho} \frac{a_i}{b_i},$$

où :

- a_i est égal au poids de l'arc entrant dans la place P_i ,
- b_i est égal au poids de l'arc sortant de la place P_i .

Selon la valeur de son gain, un circuit ϱ est classé dans l'une des trois catégories suivantes :

- $G(\varrho) > 1$: circuit générateur,
- $G(\varrho) = 1$: circuit neutre,
- $G(\varrho) < 1$: circuit absorbant.

La valeur de $G(\varrho)$ informe sur la conservation des marques dans le circuit ϱ . Si un circuit est générateur, alors le nombre de marques dans le circuit augmente avec l'évolution du marquage. Si le circuit est neutre, ce gain reste constant au cours de l'évolution du marquage. Enfin, si le circuit est absorbant, le nombre de marque diminue.

Dans le cas des GET ordinaires, le nombre total de jetons dans un circuit est constant. De par la valuation des arcs dans \mathbb{N} , ceci n'est plus vrai dans le cas des GETDM. Chao et al. [Chao et al., 1993] ont utilisé un principe similaire en introduisant la notion de marques valuées des places d'un circuit. La *marque valuée*, notée \mathcal{N}_p , d'une place p d'un circuit peut se calculer comme suit :

$$\mathcal{N}_p = \frac{m_p}{M_{qp}\theta_q}, \tag{2.9}$$

où :

- m_p correspond au marquage initial de la place p ,

- M_{qp} correspond au poids de l'arc sortant de la place p ,
- θ_q correspond à la composante de T-invariant associée à la transition en aval de la place p .

La *marque évaluée totale* d'un circuit neutre, ϱ , est donnée par la somme des marques évaluées des places de ce circuit. Elle est donnée par :

$$\mathcal{N}(\varrho) = \sum_{p \in \varrho} \mathcal{N}_p. \tag{2.10}$$

L'étude de la vivacité des GETDM est difficile. Des résultats ont été obtenus dans ce domaine par Chao et al. [Chao et al., 1993] : un circuit d'un GETDM est vivant s'il contient initialement au moins une place marquée dont le marquage évalué est supérieur à un ($\mathcal{N}(\varrho) \geq 1$).

Contrairement aux GET ordinaires, il n'existe pas de condition nécessaire et suffisante pour vérifier la vivacité des GETDM.

2.2.2 Equations récurrentes des graphes d'événements temporisés discrets avec multi- plicateurs

Comme dans le cas des GET ordinaires, on se propose d'étudier les GETDM à travers leurs représentations d'état de type compteur.

Assertion 1. *L'évolution du compteur associé à la transition n_q du GETDM élémentaire de la figure 2.11 est donnée par l'équation suivante :*

$$n_q(t) = \lfloor M_{qp}^{-1}(m_p + M_{pq'}n_{q'}(t - \tau_p)) \rfloor. \tag{2.11}$$

La présence de la partie entière inférieure dans l'équation (2.11) assure la valeur discrète du compteur indépendamment de la valeur M_{qp} . En général, une transition n_q peut avoir plusieurs transitions en amont $\{n_{q'} \in \bullet\bullet q\}$, ce qui implique que l'équation compteur associée fasse intervenir le *min* entre les compteurs associés aux transitions en amont.

Exemple 2.26. La variable compteur associée à la transition n_q du GETDM représenté par la figure 2.14 satisfait l'équation suivante :

$$n_q(t) = \lfloor \frac{2 + 3n_{q'}(t - 1)}{2} \rfloor.$$

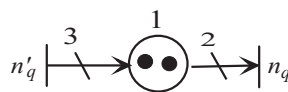


Figure 2.14 – Exemple de GETDM élémentaire .

Exemple 2.27. L'évolution dynamique du GETDM représenté par la figure 2.15 est décrite par les équations récurrentes suivantes :

$$\begin{cases} n_1(t) &= \min(2 + 3n_2(t-2), \lfloor \frac{u_1(t)}{2} \rfloor, u_2(t-1)), \\ n_2(t) &= \min(2n_1(t-1), 1 + n_2(t-1)), \\ y(t) &= n_2(t). \end{cases}$$

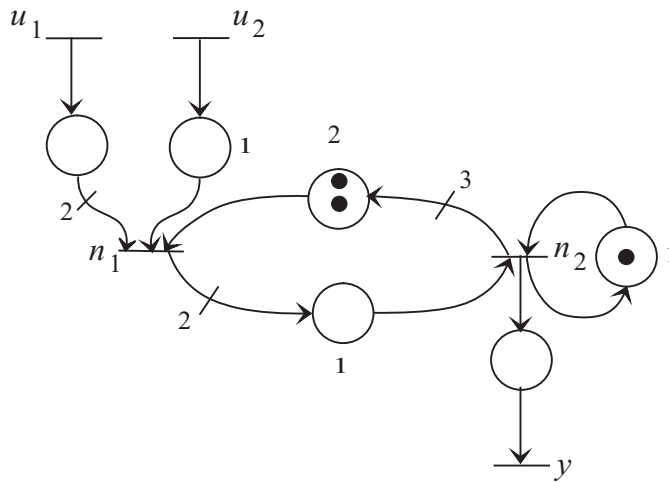


Figure 2.15 – Exemple de GETDM .

Le modèle mathématique représentant l'évolution de la dynamique des GETDM n'admet pas une représentation linéaire dans l'algèbre $(\min, +)$. Cette non linéarité est due à la présence des poids sur les arcs. De ce fait, il s'avère difficile d'utiliser l'algèbre $(\min, +)$ pour aborder, par exemple, les problèmes de commande et d'évaluation de performances des GETDM. Comme alternative, nous proposons un autre type de modélisation qui se base sur un dioïde d'opérateurs inspiré des travaux décrits par G.Cohen dans [Cohen et al., 1998b].

2.2.3 Approche opératorielle pour modéliser les graphes d'événements temporisés discrets avec multiplieurs

Trois types d'opérateurs définis de $\overline{\mathbb{Z}}^{\mathbb{R}}$ vers $\overline{\mathbb{Z}}^{\mathbb{R}}$ sont utilisés pour modéliser les GETDM.

1. **Opérateur de stock γ^ν** : Il représente un décalage événementiel de ν événements ($\nu \in \overline{\mathbb{Z}}$), et modélise un système gain élémentaire (cf. §1.3.2)

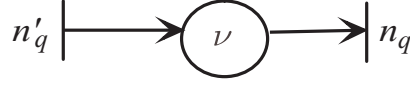


Figure 2.16 – Opérateur γ^ν .

Il est défini comme suit :

$$\forall t \in \mathbb{R}, \forall n_{q'} \in \overline{\mathbb{Z}}^{\mathbb{R}}, \quad n_q(t) = \gamma^\nu n_{q'}(t) = n_{q'}(t) + \nu.$$

Propriété 1. L'opérateur γ^ν satisfait les règles suivantes :

$$(\gamma^\nu \oplus \gamma^{\nu'}) n_q(t) = \gamma^{\min(\nu, \nu')} n_q(t).$$

$$(\gamma^\nu \otimes \gamma^{\nu'}) n_q(t) = \gamma^{\nu + \nu'} n_q(t).$$

En effet, on a

- $(\gamma^\nu \oplus \gamma^{\nu'}) n_q(t) = \min(n_q(t) + \nu, n_q(t) + \nu') = n_q(t) + \min(\nu, \nu') = \gamma^{\min(\nu, \nu')} n_q(t).$
- $(\gamma^\nu \otimes \gamma^{\nu'}) n_q(t) = \gamma^\nu(n_q(t) + \nu') = n_q(t) + \nu' + \nu = \gamma^{\nu + \nu'} n_q(t).$

2. **Opérateur de retard δ^τ** : Il représente un retard temporel de τ unités de temps ($\tau \in \overline{\mathbb{Z}}$), et modélise un système retard élémentaire (cf. 1.3.2).

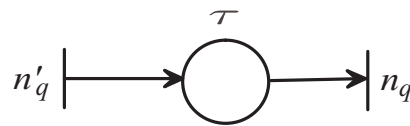


Figure 2.17 – Opérateur δ^τ .

Il est défini comme suit :

$$\forall t \in \mathbb{R}, \forall n_{q'} \in \overline{\mathbb{Z}}^{\mathbb{R}}, \quad n_q(t) = \delta^\tau n_{q'}(t) = n_{q'}(t - \tau).$$

Propriété 2. L'opérateur δ^τ satisfait les règles suivantes :

$$(\delta^\tau \oplus \delta^{\tau'}) n_q(t) = \delta^{\max(\tau, \tau')} n_q(t).$$

$$(\delta^\tau \otimes \delta^{\tau'}) n_q(t) = \delta^{\tau + \tau'} n_q(t).$$

En effet, on a :

- du fait que le signal $n_q(t)$ soit non décroissant, on a :

$$(\delta^\tau \oplus \delta^{\tau'})n_q(t) = \min(n_q(t - \tau), n_q(t - \tau')) = n_q(t - \max(\tau, \tau')) = \delta^{\max(\tau, \tau')}n_q(t).$$

- $(\delta^\tau \otimes \delta^{\tau'})n_q(t) = \delta^\tau n_q(t - \tau') = n_q(t - \tau' - \tau) = \delta^{\tau'+\tau}n_q(t).$

3. **Opérateur multiplieur** μ_m : Il représente un facteur de mise à l'échelle m .

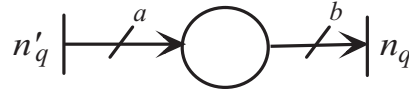


Figure 2.18 – Opérateur μ_m avec $m = \frac{a}{b}$.

Il est défini comme suit :

$$\forall t \in \mathbb{R}, \forall n_{q'} \in \overline{\mathbb{Z}}^{\mathbb{R}}, \quad n_q(t) = \mu_m n_{q'}(t) = \lfloor m \times n_{q'}(t) \rfloor,$$

avec $m \in \mathbb{Q}^+$ (m est égal au quotient de deux nombres naturels strictement positifs).

Remarque 2.28. Contrairement aux opérateurs γ^ν et δ^τ , l'opérateur μ_m n'admet pas de règles de simplification pour l'opération \otimes :

$$\forall m, m' \in \mathbb{Q}^+, \forall n_q \in \overline{\mathbb{Z}}^{\mathbb{R}}, \quad (\mu_m \otimes \mu_{m'})n_q(t) \text{ peut être différent de } (\mu_{m \times m'})n_q(t).$$

Propriété 3. L'opérateur μ_m satisfait les règles suivantes lorsqu'il est composé avec les opérateurs δ^τ et γ^ν :

$$(\mu_m \otimes \delta^\tau)n(t) = (\delta^\tau \otimes \mu_m)n(t),$$

$$(\mu_m \otimes \gamma^\nu)n(t) = (\gamma^{\nu \times m} \otimes \mu_m)n(t), \quad \text{pour } \nu \in m^{-1} \times \mathbb{N}.$$

En effet, on a :

- $(\mu_m \otimes \delta^\tau)n(t) = \lfloor m \times n(t - \tau) \rfloor = (\delta^\tau \otimes \mu_m)n(t).$
- $\forall \nu \in m^{-1} \times \mathbb{N}, \quad (\mu_m \otimes \gamma^\nu)n(t) = \lfloor m \times \nu + m \times n(t) \rfloor = m \times \nu + \lfloor m \times n(t) \rfloor = (\gamma^{\nu \times m} \otimes \mu_m)n(t),$ puisque $\nu \times m \in \mathbb{N}.$

Remarque 2.29. Comme il a été rappelé dans la section 1.3.1 du chapitre 1 (*cf.* définition 1.32), un opérateur linéaire est additif et homogène.

Si les opérateurs γ^ν et δ^τ , habituellement utilisés dans le cas des GET ordinaires, sont linéaires, l'opérateur μ_m est, par contre, seulement additif. En effet, on a :

$$\forall t \in \mathbb{R}, \forall n_1, n_2 \in \overline{\mathbb{Z}}^{\mathbb{R}},$$

$$\mu_m(n_1(t) \oplus n_2(t)) = \lfloor m \times (\min(n_1(t), n_2(t))) \rfloor = \min(\lfloor m \times n_1(t) \rfloor, \lfloor m \times n_2(t) \rfloor) = \mu_m n_1(t) \oplus \mu_m n_2(t).$$

La non linéarité de l'opérateur μ_m est due à sa non homogénéité, en effet :

$$\forall \lambda \in \mathbb{N}, \quad \lambda \otimes (\mu_m n_1(t)) = \lambda + \lfloor m \times n_1(t) \rfloor$$

est différent de

$$\mu_m(\lambda \otimes n_1(t)) = \lfloor m \times (\lambda + n_1(t)) \rfloor.$$

Présentons à présent le dioïde d'opérateurs permettant de modéliser les GETDM. Pour cela, on introduit le dioïde, noté \mathcal{D}_{\min} , non commutatif de sommes finies de composition d'opérateurs μ, γ , muni des deux lois suivantes : *min* pour l'opération (\oplus) et la composition pour l'opération (\otimes), avec pour éléments neutres $\varepsilon = \mu_{+\infty} \gamma^{+\infty}$ et $e = \mu_1 \gamma^0$.

Un élément de \mathcal{D}_{\min} est une application, notée $p = \bigoplus_{i=1}^k \mu_{r_i} \gamma^{\nu_i}$, définie de $\overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}$, telle que :

$$\forall t \in \mathbb{R}, p(x(t)) = \min_{1 \leq i \leq k} (\lfloor r_i(\nu_i + x(t)) \rfloor).$$

Utilisons l'opérateur δ pour définir un dioïde de séries formelles. Un élément de \mathcal{D}_{\min} est associé à chaque valeur de τ . Pour cela, on définit l'application g telle que :

$$g : \mathbb{R} \rightarrow \mathcal{D}_{\min}$$

$$\tau \mapsto g(\tau) = \bigoplus_{i=1}^{k_\tau} \mu_{r_i^\tau} \gamma^{\nu_i^\tau}.$$

Une telle application peut être représentée par une série formelle à coefficients dans \mathcal{D}_{\min} et dont l'indéterminée est δ . Elle est définie comme suit :

$$G(\delta) = \oint_{\tau \in \mathbb{R}} g(\tau) \delta^\tau,$$

où l'opérateur \oint correspond à la somme point à point pour $\tau \in \mathbb{R}$.

L'ensemble de ces séries formelles, muni des deux opérations suivantes :

$$F(\delta) \oplus G(\delta) : (f \oplus g)(\tau) = f(\tau) \oplus g(\tau) : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\min(f(\tau)(n(t)), g(\tau)(n(t)))\}_{t \in \mathbb{R}},$$

$$F(\delta) \otimes G(\delta) : (f \otimes g)(\tau) = \bigoplus_{i \in \mathbb{R}} f(i) \otimes g(\tau - i) : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\inf_{i \in \mathbb{R}} (f(i) + g(\tau - i))(n(t))\}_{t \in \mathbb{R}},$$

est un dioïde, noté $\mathcal{D}_{\min}[\delta]$. Ce dioïde a pour éléments neutres $\varepsilon = \mu_{+\infty} \gamma^{+\infty} \delta^{-\infty}$ et $e = \mu_1 \gamma^0 \delta^0$.

Un élément de $\mathcal{D}_{\min}[\delta]$ permet de coder un signal $n(t)$ sous forme d'une série formelle en δ , et il permet également de modéliser la relation liant deux transitions d'un GETDM.

Exemple 2.30. Dans le dioïde $\mathcal{D}_{\min}[\delta]$, le compteur $n_q(t)$, associé à la transition n_q du GETDM élémentaire de la figure 2.11, satisfait l'équation suivante :

$$N_q(\delta) = \mu_{M_{qp}^{-1}} \gamma^{m_p} \delta^{\tau_p} \mu_{M_{pq'}} N_{q'}(\delta). \quad (2.12)$$

Pour $N_{q'}$ correspondant à une impulsion, notée $E(\delta)$, et égale à $e = \mu_1 \gamma^0 \delta^0$, on obtient la relation liant la transition n_q et $n_{q'}$ sous forme d'une composition d'opérateurs donnée par $\mu_{M_{qp}^{-1}} \gamma^{m_p} \delta^{\tau_p} \mu_{M_{pq'}}$.

Assertion 2. L'évolution de la dynamique d'un GETDM est donnée par la représentation d'état suivante :

$$\begin{cases} N(\delta) = A N(\delta) \oplus B U(\delta), \\ Y(\delta) = C N(\delta) \oplus D U(\delta), \end{cases} \quad (2.13)$$

où $N(\delta)$ est le vecteur d'état composé des compteurs associés aux transitions internes du graphe, codés dans le dioïde $\mathcal{D}_{\min}[\delta]$. $U(\delta)$ et $Y(\delta)$ sont les vecteurs d'entrées et de sorties codés dans $\mathcal{D}_{\min}[\delta]$.

Les matrices A, B, C, D sont composées de monômes de séries formelles en δ , chaque monôme modélise une relation liant deux transitions du graphe dans $\mathcal{D}_{\min}[\delta]$.

2.2.4 Exemple

Dans le dioïde $\mathcal{D}_{\min}[\delta]$, le GETDM de la figure 2.15 admet la représentation d'état suivante :

$$\begin{cases} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \end{pmatrix} = \begin{pmatrix} \varepsilon & \gamma^2 \delta^2 \mu_3 \\ \delta \mu_2 & \gamma \delta \end{pmatrix} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \end{pmatrix} \oplus \begin{pmatrix} \mu_{1/2} & \delta \\ \varepsilon & \varepsilon \end{pmatrix} U(\delta), \\ Y(\delta) = (\varepsilon \ e) \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \end{pmatrix}. \end{cases} \quad (2.14)$$

Une telle représentation est décrite dans l'algèbre usuelle dans l'exemple 2.27

2.3 Graphes d'événements temporisés continus avec multiplieurs

Dans la section précédente, nous avons introduit les GETDM. Une méthode de modélisation de ces graphes dans un dioïde d'opérateurs a été proposée. Dans cette section sont abordés les RdP continus. Ces graphes sont utilisés dans deux cas : soit les systèmes à modéliser sont de nature continue (fluide), soit ils sont utilisés lorsque le marquage des modèles devient trop important, ce qui rend difficile l'utilisation des RdP discrets.

Définition 2.31 (Réseau de Petri continu). [David and Alla, 1992, Alla and David, 1998]

Un réseau de Petri continu (RdPC) est un graphe biparti défini par un 6-tuplet $(\mathcal{P}, \mathcal{T}, M, m, \tau, V)$, où :

- \mathcal{P} (resp., \mathcal{T}) est un ensemble non vide et fini de places (resp., transitions) ;
- $M \in \mathbb{R}^{(\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})}$ définit les poids des arcs : pour $p \in \mathcal{P}$ et $q \in \mathcal{T}$, le multiplieur M_{pq} (resp., M_{qp}) spécifie le poids dans \mathbb{R}^+ de l'arc allant de la transition n_q vers la place p (resp., de la place p vers la transition n_q) ;
- $m \in \mathbb{R}^{\mathcal{P}}$ définit le marquage initial : m_p désigne la quantité de fluide initialement contenue dans la place p ;
- $\tau \in \mathbb{R}^{\mathcal{P}}$ définit les temporisations associées aux places : τ_p correspond à la durée minimale de séjour du fluide dans la place p avant de contribuer au franchissement de ses transitions immédiatement en aval ;
- $V : \mathcal{T} \rightarrow \mathbb{R}^{+*}$ est l'application qui associe à chaque transition n_q un réel strictement positif appelé *vitesse maximale de franchissement*.

Selon la vitesse de franchissement des transitions, on distingue deux types de RdP continus :

1. Les *RdP continu à vitesses constantes* où les vitesses de franchissement des transitions sont constantes.
2. Les *RdP continu à vitesses variables* où les vitesses de franchissement des transitions dépendent du marquage [David and Alla, 1992, §3].

Remarque 2.32. Dans la suite de ce mémoire, notre intérêt va porter sur une classe des RdP continus à vitesses constantes, appelé graphes d'événements temporisés continus avec multiplieurs, notés GETCM.

Du point de vue graphique, les GETCM sont modélisés de la même façon que les GETDM. Cependant, pour éviter toute confusion, les transitions sont modélisées par des rectangles et les places par des doubles cercles (cf. figure 2.19).

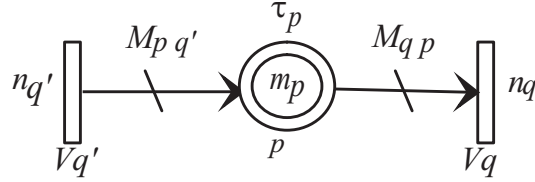


Figure 2.19 – Un GETCM élémentaire.

2.3.1 Propriété des graphes d'événements temporisés continus avec multiplieurs

Définition 2.33. [David and Alla, 1992] Une place p d'un RdP continu est *alimentée* à un instant t si et seulement s'il y a au moins une transition appartenant à $\bullet p$ qui est validée (cf. figure 2.20. a).

Définition 2.34. [David and Alla, 1992] Une transition n_q est *validée* à un instant t si toute place p appartenant à $\bullet q$ satisfait au moins une des deux conditions suivantes (cf. figure 2.20. b) :

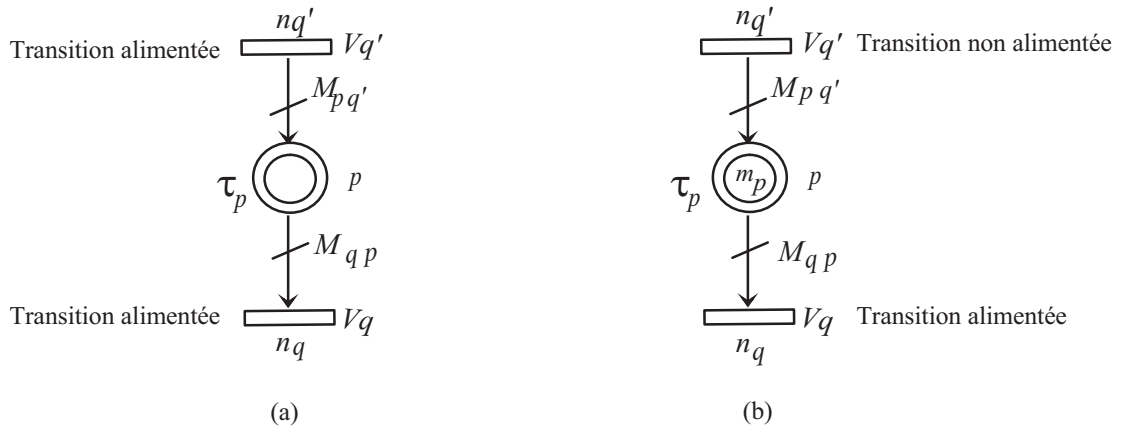


Figure 2.20 – Alimentation de transitions continues.

1. A l'instant t , $m(p) \geq 0$ ($m(p)$ correspond au marquage de la place p).
2. La place p est alimentée.

Une transition n_q est *fortement validée* si toutes les places appartenant à $\bullet q$ satisfont la première condition, elle est *faiblement validée* sinon.

• **Règle de Fonctionnement** Dans ce type de graphe (cf. figure 2.19), lors du franchissement de la transition $n_{q'}$, pendant dt unités de temps, une quantité $V_{q'} M_{p q'} dt$ de marques est ajoutée dans la place

p . Après un séjour de τ_p unités de temps dans cette place, une quantité $V_q M_{qp} dt$ de marques est retirée de la place p . Un tel comportement se produit tant que $m_p > 0$.

Exemple 2.35. Cet exemple est issu du livre [David and Alla, 1992]. Les places p_1 , p_2 et p_3 du GET représenté par la figure 2.21.a représentent les contenus des trois réservoirs décrits dans la figure 2.21.b. La transition n_1 , qui modélise la synchronisation entre le liquide du réservoir 1 et le liquide du réservoir 2 pour former un mélange dans le réservoir 3, exprime que durant *une unité de temps*, une quantité de fluide de 2.6×0.5 peut être retirée de la place p_1 , une quantité de fluide de 2×0.5 peut être retirée de la place p_2 et une quantité de fluide de 4.6×0.5 est ajoutée à la place p_3 .

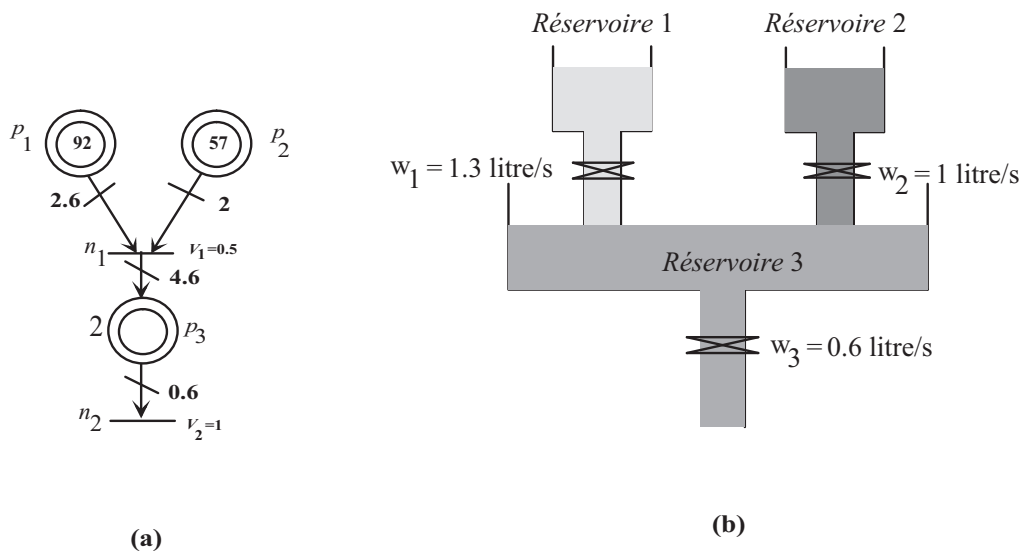


Figure 2.21 – Un GETCM et son système équivalent.

Comme dans le cas des GET discrets ordinaires, R. David et H. Alla ont établi qu'un GET continu ordinaire est vivant si chacun de ses circuits contient au moins une place marquée [David, 1997].

Contrairement au GETDM, une condition nécessaire et suffisante sur la vivacité des circuits des GETCM a été donnée par le théorème suivant.

Théorème 2.36. [Demongodin et al., 1999] *Un circuit, noté ρ , continu avec multiplieurs est vivant si, et seulement si :*

- *Il possède au moins une place de marquage non nul (i.e., $\exists p \in \rho$ tel que le marquage $m_p > 0$).*
- *Il est neutre ou générateur, c'est-à-dire $G(\rho) \geq 1$ (cf. définition 2.25).*

2.3.2 Equations récurrentes des graphes d'événements temporisés continus avec multiplieurs

Comme dans le cas discret, l'état regroupe les compteurs associés à chaque transition du graphe. Un compteur associé à une transition n_q d'un GETCM est une application, noté n_q , croissante de $\mathbb{R} \rightarrow \overline{\mathbb{R}}$, où pour tout $t \in \mathbb{R}$, $n_q(t)$ correspond à la quantité de fluide ayant franchie la transition n_q jusqu'à la date t .

Assertion 3. L'évolution d'un compteur associé à une transition d'un GETCM est donnée par l'équation récurrente suivante (cf. figure 2.19 pour les notations):

$$n_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \left(\inf_{h \geq 0} (M_{qp}^{-1} m_p + M_{qp}^{-1} M_{pq'} n_{q'}(t - \tau_p - h) + V_q h) \right), \quad (2.15)$$

Physiquement, la vitesse de franchissement associée à la transition n_q peut être vue comme un entonnoir ayant un débit maximum de V_q litres par seconde (cf. figure 2.22).

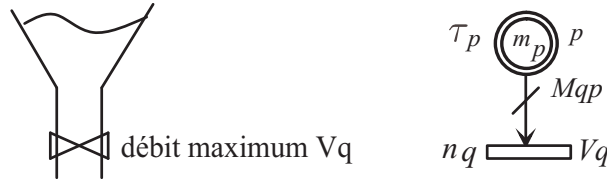


Figure 2.22 – Interprétation de la vitesse maximale de franchissement V_q avec la fonction d'un entonnoir.

Sachant que le débit maximum avec lequel la transition n_q est franchie est égale à V_q , on a :

$$\forall t, \forall h \geq 0, n_q(t) \leq n_q(t - h) + V_q h.$$

D'autre part, on a pour $n_{q'} \in \bullet p$:

$$\forall t, n_q(t) \leq M_{qp}^{-1} m_p + M_{qp}^{-1} M_{pq'} n_{q'}(t - \tau_p).$$

Il en résulte que :

$$\forall t, \forall h \geq 0, n_q(t) \leq M_{qp}^{-1} m_p + M_{qp}^{-1} M_{pq'} n_{q'}(t - \tau_p - h) + V_q h,$$

aussi, on a pour tout t :

$$n_q(t) \leq \inf_{h \geq 0} (M_{qp}^{-1} m_p + M_{qp}^{-1} M_{pq'} n_{q'}(t - \tau_p - h) + V_q h).$$

Puisque l'écoulement du fluide est supposé le plus élevé possible, on obtient :

$$n_q(t) = \inf_{h \geq 0} (M_{qp}^{-1}m_p + M_{qp}^{-1}M_{pq'}n_{q'}(t - \tau_p - h) + V_q h),$$

ceci montre que le comportement de la sortie correspond à un produit d'*inf-convolution* entre l'entrée et la relation de transfert $f(t) = V_q t$ (cf. définition 1.30). De par la présence des poids sur les arcs, ce type de système n'admet pas une représentation (*min, +*) linéaire. Comme alternative et de manière analogue à ce qui a été fait dans le cas des GETDM, on utilise une autre approche de modélisation qui va nous permettre d'exprimer la représentation d'état des GETCM.

2.3.3 Approche opératorielle pour modéliser les graphes d'événements temporisés continus avec multipliers

L'approche présentée ici est inspirée de celle introduite dans [Cohen et al., 1998a], [Cohen et al., 1998b], pour la modélisation des GETCM pour lesquels la vitesse maximale de franchissement est supposée infinie.

Pour la modélisation des GETCM, nous avons utilisé quatre types d'opérateurs, γ_c^ν , δ_c^τ , μ_{m_c} et ω_V , définis de $\overline{\mathbb{R}}^{\mathbb{R}}$ vers $\overline{\mathbb{R}}^{\mathbb{R}}$.

Remarque 2.37. L'indice "c" (comme continu) est utilisé pour différencier les opérateurs continus des opérateurs discrets introduits précédemment. Les opérateurs discrets seront d'ailleurs indicés par "d" lorsque nous utilisons conjointement les opérateurs discrets et continus pour représenter les graphes hybrides. Une description des opérateurs continus est faite ci-dessous.

- γ_c^ν représente un décalage d'une quantité de fluide égale à ν ($\nu \in \overline{\mathbb{R}}$) et est défini par $\gamma_c^\nu n(t) = n(t) + \nu$.
- δ_c^τ représente un retard de τ unités de temps ($\tau \in \overline{\mathbb{R}}$) de la progression du fluide et est défini par $\delta_c^\tau n(t) = n(t - \tau)$.
- μ_{m_c} représente un facteur de mise à l'échelle m ($m \in \mathbb{R}^+$) et est défini par $\mu_{m_c} n(t) = m \times n(t)$. A la différence de l'opérateur discret, lequel comporte une partie entière, l'opérateur μ_{m_c} vérifie la propriété suivante :

$$\forall t, m, m' \in \mathbb{R}^+, (\mu_{m_c} \otimes \mu_{m'_c})n(t) = \mu_{(m \times m')_c} n(t).$$

- ω_V représente un limiteur de flux ayant une vitesse maximale de franchissement V , avec $V \in \mathbb{R}^+ \cup \{+\infty\}$, il est défini par $\omega_V n(t) = \inf_{h \geq 0} (n(t - h) + V \times h)$.

L'opérateur ω_V satisfait les règles suivantes :

$$\omega_V \otimes \omega_{V'} = \omega_{\min(V, V')},$$

$$\omega_V \oplus \omega_{V'} = \omega_{\min(V, V')}.$$

Physiquement, deux limiteurs de débit placés en série ou en parallèle se comportent comme un seul, en fait, comme celui qui a la plus petite vitesse maximale de franchissement.

Comme dans le cas discret, les opérateurs $\gamma_c^\nu, \delta_c^\tau$, et ω_V sont linéaires, l'opérateur μ_{m_c} est seulement additif.

Proposition 2.38. *Les opérateurs utilisés pour la modélisation des GETCM vérifient les relations suivantes :*

1. $\gamma_c^\nu \omega_V = \omega_V \gamma_c^\nu.$
2. $\delta_c^\tau \omega_V = \omega_V \delta_c^\tau.$
3. $\mu_{m_c} \omega_V = \omega_{(V \times m)} \mu_{m_c}.$
4. $\mu_{m_c} \gamma_c^\nu = \gamma_c^{m \times \nu} \mu_{m_c},$ (contrairement au cas discret, cette propriété est toujours vérifiée).
5. $\mu_{m_c} \delta_c^\tau = \delta_c^\tau \mu_{m_c}.$

Preuve :

$$\begin{aligned} \text{Point 1 : } \forall t \in \mathbb{R}, \forall V \in \mathbb{R}^+, n \in \overline{\mathbb{R}}, \quad & (\gamma_c^\nu \omega_V)n(t) = \nu + \inf_{h \geq 0} (n(t-h) + V \times h) \\ & = \inf_{h \geq 0} (\nu + n(t-h) + V \times h) \\ & = (\omega_V \gamma_c^\nu)n(t). \end{aligned}$$

$$\begin{aligned} \text{Point 2 : } \forall t \in \mathbb{R}, \forall V \in \mathbb{R}^+, n \in \overline{\mathbb{R}}, \quad & (\delta_c^\tau \omega_V)n(t) = \inf_{h \geq 0} (n(t-\tau-h) + V \times h) \\ & = (\omega_V \delta_c^\tau)n(t). \end{aligned}$$

$$\begin{aligned} \text{Point 3 : } \forall t \in \mathbb{R}, \forall V \in \mathbb{R}^+, \forall m \in \mathbb{R}^+, n \in \overline{\mathbb{R}}, \quad & (\mu_{m_c} \omega_V)n(t) = m \times \inf_{h \geq 0} (n(t-h) + V \times h) \\ & = \inf_{h \geq 0} (m \times n(t-h) + m \times V \times h) \\ & = \omega_{(V \times m)} \mu_{m_c} n(t). \end{aligned}$$

$$\text{Point 4 : } \forall t \in \mathbb{R}, \forall m \in \mathbb{R}^+, n \in \overline{\mathbb{R}}, \quad (\mu_{m_c} \gamma_c^\nu)n(t) = m \times n(t) + m \times \nu = (\gamma_c^{m \times \nu} \mu_{m_c})n(t).$$

$$\text{Point 5 : } \forall t \in \mathbb{R}, \forall m \in \mathbb{R}^+, n \in \overline{\mathbb{R}}, \quad (\mu_{m_c} \delta_c^\tau)n(t) = m \times n(t - \tau) = (\delta_c^\tau \mu_{m_c})n(t).$$

Remarque 2.39. Les opérateurs γ_c^ν et δ_c^τ vérifient les mêmes règles de simplification que dans le cas discret.

Introduisons, à présent, le dioïde d'opérateurs permettant de modéliser les GETCM. Pour cela, on note \mathcal{C}_{\min} le dioïde non commutatif de sommes finies de composition d'opérateurs $\omega_V, \mu_{m_c}, \gamma_c^\nu$ muni des deux lois suivantes : l'opération *min* pour l'opération (\oplus) et la composition pour l'opération (\otimes) , avec pour éléments neutres $\varepsilon = \mu_{(+\infty)_c} \gamma_c^{+\infty}$ et $e = \omega_{+\infty} \mu_{(1)_c} \gamma_c^0$.

Un élément de \mathcal{C}_{\min} est une application, notée $p = \bigoplus_{i=1}^k \omega_{V_i} \mu_{m_{i_c}} \gamma_c^{\nu_i}$, définie de $\overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}$, telle que :

$$\forall t \in \mathbb{R}, p(x(t)) = \min_{1 \leq i \leq k} (\inf_{h \geq 0} (m_i \nu_i + m_i x(t-h) + V_i h)).$$

Utilisons l'opérateur δ_c pour définir un dioïde de séries formelles. Un élément de \mathcal{C}_{\min} est associé à chaque valeur de τ . Pour cela, on définit l'application g telle que :

$$g : \mathbb{R} \rightarrow \mathcal{C}_{\min},$$

$$\tau \mapsto g(\tau) = \bigoplus_{i=1}^{k_\tau} \omega_{V_i^\tau} \mu_{m_{(i)_c}^\tau} \gamma_c^{\nu_{(i)}^\tau}.$$

Une telle application peut être représentée par une série formelle à coefficients dans \mathcal{C}_{\min} et dont l'indéterminée est δ_c . Elle est définie comme suit :

$$G(\delta) = \bigoplus_{\tau \in \mathbb{R}} g(\tau) \delta_c^\tau.$$

L'ensemble de ces séries formelles muni des deux opérations suivantes :

$$F(\delta) \oplus G(\delta) : (f \oplus g)(\tau) = f(\tau) \oplus g(\tau) : \overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\min(f(\tau)(n(t)), g(\tau)(n(t)))\}_{t \in \mathbb{R}},$$

$$F(\delta) \otimes G(\delta) : (f \otimes g)(\tau) = \bigoplus_{i \in \mathbb{R}} f(i) \otimes g(\tau-i) : \overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\inf_{i \in \mathbb{R}} (f(i) + g(\tau-i))(n(t))\}_{t \in \mathbb{R}},$$

est un dioïde noté $\mathcal{C}_{\min}[[\delta]]$ avec pour éléments neutres $\varepsilon = \mu_{(+\infty)_c} \gamma_c^{+\infty} \delta_c^{-\infty}$ et $e = \omega_{+\infty} \mu_{(1)_c} \gamma_c^0 \delta_c^0$.

Assertion 4. La dynamique d'un GETCM peut être décrite par le système d'équations suivant :

$$\begin{cases} N(\delta) = AN(\delta) \oplus BU(\delta), & (i) \\ Y(\delta) = CN(\delta) \oplus DU(\delta), & (ii) \end{cases} \quad (2.16)$$

où $N(\delta)$ est le vecteur d'état composé des compteurs associés aux transitions internes du graphe, à valeurs dans le dioïde $\mathcal{C}_{\min}[[\delta]]$. $U(\delta)$ et $Y(\delta)$ sont les vecteurs d'entrées et de sorties à valeurs dans

$\mathcal{C}_{\min}[\delta]$.

Les éléments des matrices A, B, C, D sont des monomes de séries formelles en δ_c , elles modélisent les relations liant les transitions du graphe dans $\mathcal{C}_{\min}[\delta]$.

Tout chemin π d'un GETCM admet une relation liant son entrée à sa sortie, notée h_π . Elle est égale au produit (\otimes) des opérateurs de types $\omega_V, \mu_{m_c}, \gamma_c^\nu, \delta_c^\tau$, pris le long du chemin π . Cette relation est un monôme appartenant à $\mathcal{C}_{\min}[\delta]$ qui s'écrit de la façon suivante :

$$h_\pi = \omega_{V_\pi} \gamma_c^{\nu_\pi} \delta_c^{\tau_\pi} \mu_{(m)_\pi}, \quad (2.17)$$

où τ_π correspond à la somme des temporisations sur les places appartenant au chemin π et V_π, ν_π, m_π sont obtenues par l'application des différentes règles de commutation entre les opérateurs (cf. proposition 2.38).

Exemple 2.40. Le compteur $n_q(t)$ associé à la transition n_q du GETCM élémentaire de la figure 2.19 satisfait l'équation suivante :

$$\begin{aligned} N_q(\delta) &= \omega_{V_q} \mu_{(M_{qp}^{-1})_c} \gamma_c^{m_p} \delta_c^{\tau_p} \mu_{(M_{pq'})_c} N_{q'}(\delta) \\ &= \omega_{V_q} \gamma_c^{M_{qp}^{-1} m_p} \delta_c^{\tau_p} \mu_{(M_{pq'}, M_{qp}^{-1})_c} N_{q'}(\delta). \end{aligned}$$

Physiquement, le compteur $N_q(\delta)$ est égal au compteur $N_{q'}(\delta)$ composé avec les systèmes élémentaires décrits dans la figure 2.23.

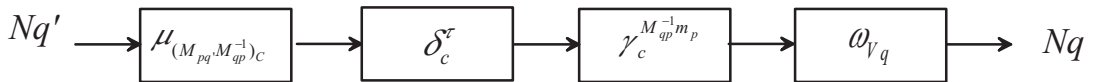


Figure 2.23 – Composition en série de systèmes élémentaires continus.

Dans le cas où $N_{q'}(\delta)$ est une entrée impulsionnelle, notée $E(\delta)$, égale à $e = \omega_{+\infty} \mu_{(1)_c} \gamma_c^0 \delta_c^0$, on obtient :

$$h_\pi = \omega_{V_q} \gamma_c^{M_{qp}^{-1} m_p} \delta_c^{\tau_p} \mu_{(M_{pq'}, M_{qp}^{-1})_c}.$$

Assertion 5. La relation entrée-sortie d'un GETCM, notée H , est définie comme suit :

$$Y = HU, \quad \text{où } H = D \oplus CA^*B \quad \text{avec } A^* = e \oplus A \oplus A^2 \oplus \dots \quad (2.18)$$

La matrice H , appelée matrice de transfert, est composée d'éléments appartenant au dioïde $\mathcal{C}_{\min}[\delta]$.

2.3.4 Exemple

En appliquant le système d'équations (2.16), le GETCM de la figure 2.24 admet la représentation d'état suivante :

$$\left\{ \begin{array}{l} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \end{pmatrix} = \begin{pmatrix} \omega_{V1}\gamma_c^2\delta_c^3 & \omega_{V1}\delta_c^1\mu_{(1/2)_c} \\ \omega_{V2}\gamma_c^2\delta_c^1\mu_{(2)_c} & \varepsilon \end{pmatrix} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \end{pmatrix} \oplus \begin{pmatrix} \omega_{V1}\gamma_c^{1/2}\mu_{(3/2)_c} \\ \varepsilon \end{pmatrix} U(\delta), \\ Y(\delta) = (\varepsilon \quad e) \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \end{pmatrix}. \end{array} \right. \quad (2.19)$$

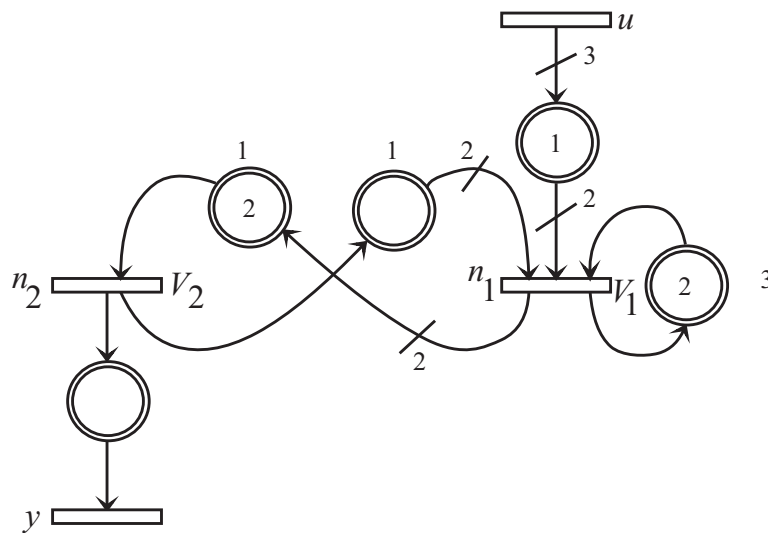


Figure 2.24 – Graphe d'événements temporisé continu avec multipliers.

2.4 Graphes d'événements temporisés hybrides avec multiplieurs

L'étude des systèmes en Automatique se fait habituellement en considérant des modèles de nature soit continue, soit à événements discrets. Toutefois, il n'est pas rare que l'étude de certains systèmes nécessite des modèles où interviennent à la fois des phénomènes de nature continue et à événements discrets. De tels modèles sont dits hybrides.

Parmi les différents outils de modélisation des systèmes hybrides, les réseaux de Petri apparaissent comme un outils performant et simple d'emploi, notamment à travers une représentation graphique uniforme des parties continues et à événements discrets. En effet, les réseaux de Petri continus sont des outils adaptés pour la modélisation et l'analyse des systèmes continus. Par contre, ils ne permettent pas la prise en compte d'événements discrets (une commande par exemple). Une alternative consiste à coupler des noeuds (places et transitions) continus avec des noeuds discrets [Giua and Usai, 1998]. Un tel couplage de comportements fluide et discret constitue l'idée de base des réseaux de Petri hybride introduits par René David dans [David, 1997] et Hassan Alla [Alla and David, 1998].

Dans notre cas d'étude, on s'est intéressé aux graphes d'événements temporisés hybrides avec multiplieurs, notés GETHM.

La figure 2.25 regroupe les liaisons possibles entre les sommets (place et transition) :

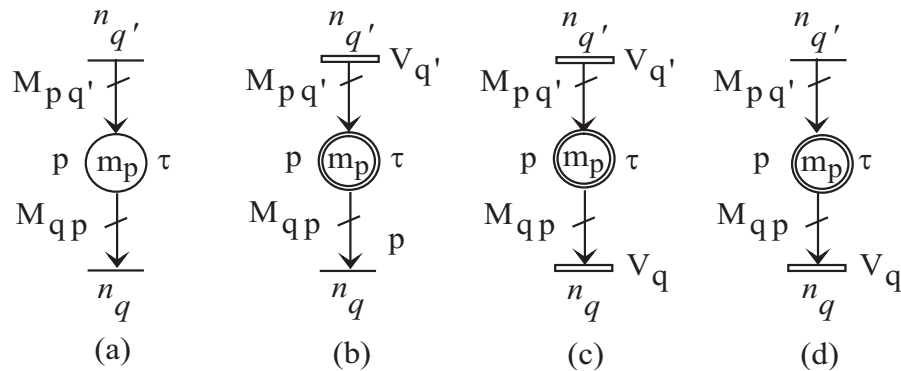


Figure 2.25 – Transferts élémentaires d'un GETHM .

Cas particulier : Une transition continue peut aussi avoir des places d'entrée discrètes. Cependant, ces dernières doivent aussi être des places de sortie de cette même transition, et *vice versa* (toutes les places de sortie discrètes doivent aussi être des places d'entrée), avec des arcs de même poids. Ceci est illustré par la place p discrète de la figure 2.26 ; une telle configuration permet de préserver un marquage en nombres entiers des places discrètes. Dans un système de production, ce graphe peut être utilisé pour

modéliser, par exemple, une panne ou une réparation d'une machine. Notons que dans le cas des GET, cette configuration présente peu d'intérêt.

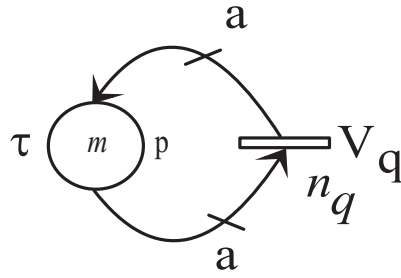


Figure 2.26 – Liaison d'une place discrète à une transition continue.

2.4.1 Equations récurrentes des graphes d'événements temporisés hybrides avec multipliers

Comme dans le cas des GETDM et des GETCM, l'état est constitué de compteurs associé à chaque transition du graphe. Leurs évolutions peut être illustrées par les exemples suivants.

Exemple 2.41. L'évolution dynamique du GETHM représenté sur la figure 2.27 est décrite par les équations récurrentes suivantes :

$$\begin{cases} n_1(t) &= \min(\inf_{h \geq 0} (\frac{u(t-h)}{2} + V_1 h), \inf_{h \geq 0} (m + n_3(t - t_1 - h) + V_1 h)), \\ n_2(t) &= \inf_{h \geq 0} (3n_4(t - h - 1) + V_2 h), \\ n_3(t) &= \lfloor n_1(t - 1) \rfloor, \\ n_4(t) &= \min(n_3(t), \lfloor 2 + n_2(t) \rfloor), \\ y(t) &= \lfloor n_2(t) \rfloor. \end{cases}$$

Exemple 2.42. L'évolution dynamique du GETHM représenté sur la figure 2.28 est décrite par les équations récurrentes suivantes :

$$\begin{cases} n_1(t) &= \min(\inf_{h \geq 0} (m + n_2(t - h) + V_1 h), \inf_{h \geq 0} (1 + 2u_1(t - h) + V_1 h), \inf_{h \geq 0} (\frac{u_2(t-t_1-h)}{2} + V_1 h)), \\ n_2(t) &= \min(\lfloor n_1(t - 1) \rfloor, 1 + n_2(t - t_2)), \\ y(t) &= n_2(t). \end{cases}$$

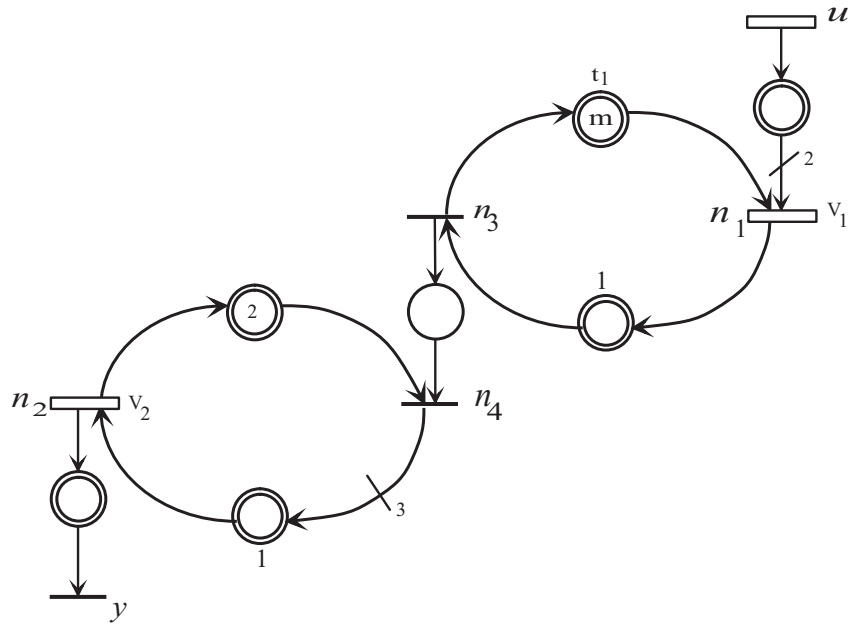


Figure 2.27 – Graphe d'événements temporisé hybride avec multipliers.

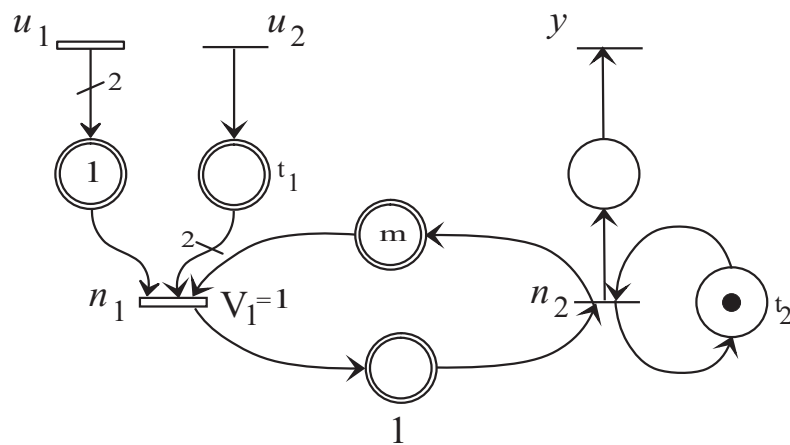


Figure 2.28 – Graphe d'événements temporisé hybride avec multipliers.

2.4.2 Approche opératorielle pour modéliser les graphes d'événements temporisés hybrides avec multiplieurs

En plus des opérateurs continus et discrets utilisés pour la modélisation de la partie continue et discrète d'un GETHM, deux autres types d'opérateurs sont introduits :

- L'opérateur I , correspondant à l'injection canonique, définie de $\overline{\mathbb{Z}}^R \rightarrow \overline{\mathbb{R}}^R$, telle que $I(n(t)) = n(t)$. Cet opérateur est utilisé pour modéliser l'interface Discrète-Continue (DC).
- L'opérateur P , correspondant à la projection canonique, définie de $\overline{\mathbb{R}}^R \rightarrow \overline{\mathbb{Z}}^R$, telle que $P(n(t)) = \lfloor n(t) \rfloor$. Cet opérateur est utilisé pour modéliser l'interface Continue-Discrète (CD).

Remarque 2.43. En utilisant les opérateurs I et l'opérateur P , l'opérateur γ_d^ν peut être défini par $\gamma_d^\nu = P\gamma_c^\nu I$ ($\nu \in \mathbb{Z} \cup \{+\infty\}$). En effet, on a :

$$I : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}, \{n(t) \in \overline{\mathbb{Z}}\}_{t \in \mathbb{R}} \mapsto \{n(t) \in \overline{\mathbb{R}}\}_{t \in \mathbb{R}},$$

$$\gamma_c^\nu : \overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}, \{n(t) \in \overline{\mathbb{R}}\}_{t \in \mathbb{R}} \mapsto \{n(t) + \nu \in \overline{\mathbb{R}}\}_{t \in \mathbb{R}},$$

$$P : \overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t) \in \overline{\mathbb{R}}\}_{t \in \mathbb{R}} \mapsto \{\lfloor n(t) \rfloor \in \overline{\mathbb{Z}}\}_{t \in \mathbb{R}}.$$

Par conséquent, on obtient :

$$P\gamma_c^\nu I : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t) \in \overline{\mathbb{Z}}\}_{t \in \mathbb{R}} \mapsto \{\lfloor n(t) + \nu \rfloor \in \overline{\mathbb{Z}}\}_{t \in \mathbb{R}} \text{ et } \lfloor n(t) + \nu \rfloor = n(t) + \nu \text{ puisque } n(t) + \nu \in \overline{\mathbb{Z}}.$$

D'une façon équivalente, nous avons $\delta_d^\tau = P\delta_c^\tau I$ avec $\tau \in \mathbb{R}$ et $\mu_{m_d} = P\mu_{(m)_c} I$ avec $m \in \mathbb{Q}^+$.

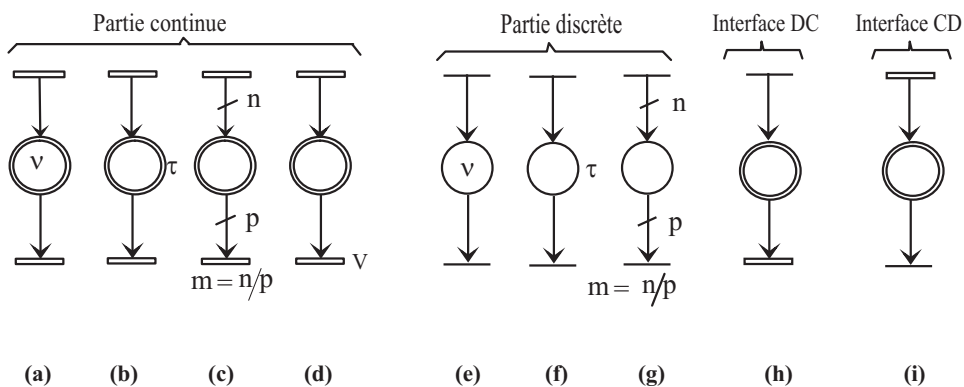


Figure 2.29 – Graphes élémentaires correspondant aux opérateurs utilisés pour modéliser les GETHM.

Assertion 6. L'évolution dynamique d'un GETHM peut être représentée par les équations d'état sui-

vantes :

$$\left\{ \begin{array}{l} \begin{pmatrix} N_c(\delta) \\ N_d(\delta) \end{pmatrix} = \begin{pmatrix} A_{CC} & A_{CD}I \\ PA_{DC} & A_{DD} \end{pmatrix} \begin{pmatrix} N_c(\delta) \\ N_d(\delta) \end{pmatrix} \oplus \begin{pmatrix} B_{CC} & B_{CD}I \\ PB_{DC} & B_{DD} \end{pmatrix} \begin{pmatrix} U_c(\delta) \\ U_d(\delta) \end{pmatrix}, \\ \begin{pmatrix} Y_c(\delta) \\ Y_d(\delta) \end{pmatrix} = \begin{pmatrix} C_{CC} & C_{CD}I \\ PC_{DC} & C_{DD} \end{pmatrix} \begin{pmatrix} N_c(\delta) \\ N_d(\delta) \end{pmatrix} \oplus \begin{pmatrix} D_{CC} & D_{CD}I \\ PD_{DC} & D_{DD} \end{pmatrix} \begin{pmatrix} U_c(\delta) \\ U_d(\delta) \end{pmatrix}. \end{array} \right. \quad (2.20)$$

Les vecteurs d'état, d'entrées et de sorties sont composés de signaux (compteurs) continus codés dans le dioïde $\mathcal{C}_{\min}[\delta]$ et de signaux discrets codés dans le dioïde $\mathcal{D}_{\min}[\delta]$, soient $N^t = (N_c^t N_d^t)$, $U^t = (U_c^t U_d^t)$, $Y^t = (Y_c^t Y_d^t)$. Les éléments des matrices A_{DD} , B_{DD} , C_{DD} et D_{DD} appartiennent au dioïde $\mathcal{D}_{\min}[\delta]$, alors que les éléments des autres matrices appartiennent au dioïde $\mathcal{C}_{\min}[\delta]$.

Remarque 2.44. Le comportement du graphe, représenté sur la Fig.2.30.a, satisfait l'équation suivante :

$$N_q(\delta) = \omega_{V_q} I \mu_{a_d^{-1}} \gamma_d^m \mu_{a_d} \delta_d^\tau P N_q(\delta), \quad N_q \in \mathcal{C}_{\min}[\delta].$$

On remarque que l'expression du compteur $N_q(\delta)$ dans le dioïde $\mathcal{C}_{\min}[\delta]$ est égale au compteur $N_q(\delta)$ décalé par une composition d'opérateurs discrets. La description de cette relation selon l'équation (2.20), qui suppose que l'intégrité des matrices est préservée, nous amène à considérer les équations suivantes :

$$\begin{pmatrix} N_q(\delta) \\ N_{q'}(\delta) \\ N_{q''}(\delta) \end{pmatrix} = \begin{pmatrix} \varepsilon_c & \varepsilon_c I & \omega_{V_q} I \\ P & \varepsilon_d & \varepsilon_d \\ P\varepsilon_c & \mu_{a_d^{-1}} \gamma_d^m \mu_{a_d} \delta_d^\tau & \varepsilon_d \end{pmatrix} \begin{pmatrix} N_q(\delta) \\ N_{q'}(\delta) \\ N_{q''}(\delta) \end{pmatrix}, \quad (N_q \in \mathcal{C}_{\min}[\delta], N_{q'}, N_{q''} \in \mathcal{D}_{\min}[\delta]),$$

ce qui correspond au graphe représenté par la figure 2.30.b.

Exemple 2.45. Les GETHM élémentaires représentés par la figure 2.25 satisfont les équations suivantes :

- $N_{q'}, N_q \in \mathcal{D}_{\min}[\delta]$: $N_q = \mu_{(M_{qp}^{-1})_d} \gamma_d^{m_p} \mu_{(M_{p q'})_d} \delta_d^\tau N_{q'}$ (Fig. 2.25.a) ;
- $N_{q'} \in \mathcal{C}_{\min}[\delta_c]$, $N_q \in \mathcal{D}_{\min}[\delta]$: $N_q = P \mu_{(M_{qp}^{-1})_c} \gamma_c^{m_p} \mu_{(M_{p q'})_c} \delta_c^\tau N_{q'}$ (Fig. 2.25.b) ;

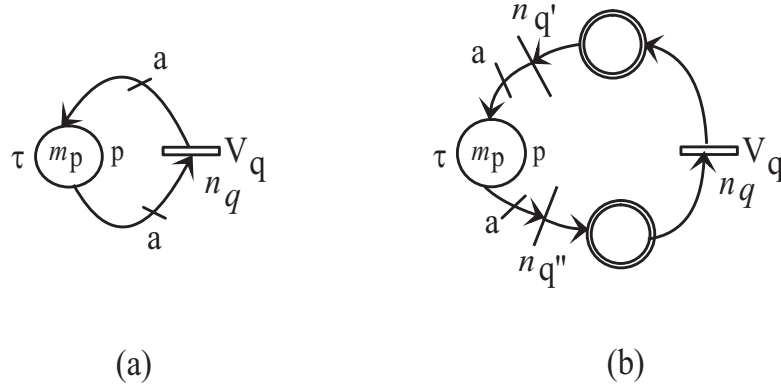


Figure 2.30 – Graphes équivalents.

- $N_{q'}, N_q \in \mathcal{C}_{\min}[\delta_c]$: $N_q = \omega_{V_q} \mu_{(M_{qp}^{-1})_c} \gamma_c^{m_p} \mu_{(M_{pq'})_c} \delta_c^\tau N_{q'}$ (Fig. 2.25.c);
- $N_{q'} \in \mathcal{D}_{\min}[\delta]$, $N_q \in \mathcal{C}_{\min}[\delta_c]$: $N_q = \omega_{V_q} \mu_{(M_{qp}^{-1})_c} \gamma_c^{m_p} \mu_{(M_{pq'})_c} \delta_c^\tau I N_{q'}$ (Fig. 2.25.d).

2.4.3 Exemple

Le GETHM représenté par la figure 2.27 admet la représentation d'état suivante :

$$\left\{ \begin{array}{l} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \\ N_3(\delta) \\ N_4(\delta) \end{pmatrix} = \begin{pmatrix} \varepsilon_c & \varepsilon_c & \omega_{V_1} \gamma_c^m \delta_c^{t_1} I & \varepsilon_c I \\ \varepsilon_c & \varepsilon_c & \varepsilon_c I & \omega_{V_2} \delta_c^1 \mu_{(3)_c} I \\ P \delta_c^1 & P \varepsilon_c & \varepsilon_d & \varepsilon_d \\ P \varepsilon_c & P \gamma_c^2 & e_d & \varepsilon_d \end{pmatrix} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \\ N_3(\delta) \\ N_4(\delta) \end{pmatrix} \oplus \begin{pmatrix} \omega_{V_1} \mu_{(1/2)_c} \\ \varepsilon_c \\ P \varepsilon_c \\ P \varepsilon_c \end{pmatrix} \\ \\ \begin{pmatrix} Y(\delta) \end{pmatrix} = \begin{pmatrix} P \varepsilon_c & P e_c & \varepsilon_d & \varepsilon_d \end{pmatrix} \begin{pmatrix} N_1(\delta) \\ N_2(\delta) \\ N_3(\delta) \\ N_4(\delta) \end{pmatrix} \oplus \begin{pmatrix} P \varepsilon_c \\ P \varepsilon_c \\ P \varepsilon_c \\ P \varepsilon_c \end{pmatrix} \end{array} \right. U(\delta), \tag{2.21}$$

avec $U(\delta), N_1(\delta), N_2(\delta) \in \mathcal{C}_{\min}[\delta]$ et $N_3(\delta), N_4(\delta), Y(\delta) \in \mathcal{D}_{\min}[\delta]$.

2.5 Conclusion

Une approche de modélisation des réseaux de Petri et plus particulièrement les graphes d'événements temporisés avec multiplieurs a été présenté dans ce chapitre. Nous avons considéré les graphes de nature discrète, continue puis hybride. Le résultat principal de ce chapitre est la modélisation des graphes d'événements temporisés hybrides avec multiplieurs. Cette approche de modélisation, inspirée des travaux de Guy cohen [Cohen et al., 1998c] et de Jan Komenda [Komenda et al., 1998], est à base d'une composition d'opérateurs discrets et continus. Dans le cas des GETDM, on a exprimé les équations récurrentes associées aux transitions du graphe dans un dioïde d'opérateurs discrets $\mathcal{D}_{\min}[\delta]$. Dans le cas des GETHM, en plus de $\mathcal{D}_{\min}[\delta]$, on a utilisé un autre dioïde, noté $\mathcal{C}_{\min}[\delta]$, basé sur des opérateurs continus pour pouvoir exprimer les équation récurrentes régissant l'évolution de la dynamique continue de ces graphes. Pour modéliser les interfaces possibles entre la partie continue et la partie discrète, on a utilisé deux opérateurs : l'opérateur I pour modéliser l'interface discrète-continue, et l'opérateur P pour modéliser l'interface continue-discrète.

Les modèles d'état obtenus sont utilisés pour aborder, dans le chapitre suivant, le problème de la synthèse de commande en juste-à-temps des graphes d'événements temporisés discrets et hybrides avec multiplieurs, ainsi que l'évaluation de performances de ces graphes dans le chapitre quatre.

CHAPITRE 3

Sur la commande optimale des graphes d'événements temporisés discrets et hybrides avec multiplieurs

Classiquement, commander un système revient à piloter ses entrées dans le but d'obtenir des performances spécifiées par un cahier des charges. Dans le domaine des systèmes à événements discrets, deux notions de commande ont été développées *via* l'approche algébrique des dioides : la commande par *poursuite de modèle* et la commande par *poursuite de trajectoire*.

La commande par *poursuite de modèle* a pour objectif le calcul d'un correcteur, noté F , tel que le comportement entrée-sortie du système corrigé, noté G_F , composé d'un système nominal, noté H , et du correcteur F , soit aussi proche que possible de celui du modèle de référence, noté G_{ref} . Plusieurs études ont été menées. Une étude consiste à placer le correcteur en amont du système nominal, autrement dit, à précompenser le système [Libeaut and Loiseau, 1996] [Cottenceau, 1999, §3.2]. Une autre étude consiste à placer le correcteur au niveau de la boucle de retour [Cottenceau et al., 2000, Cottenceau et al., 2001, Lüders and Santos-Mendes, 2002]. La figure 3.1 présente les schémas blocs du système nominal, du système en boucle fermée avec un correcteur situé au niveau de la boucle de retour et du modèle de référence.

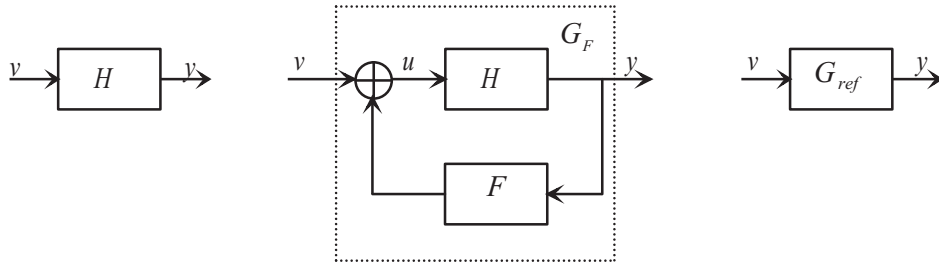


Figure 3.1 – Commande par *poursuite de modèle*.

Une structure de commande généralisant ces approches a été proposée dans [Maia et al., 2003]. Elle est basée sur l'utilisation simultanée d'un précompensateur et d'un bloc situé dans la boucle de retour de sortie.

La deuxième technique de commande est un problème de commande par *poursuite de trajectoire* connue *a priori*. Autrement dit, il s'agit d'établir une trajectoire d'entrée (commande) du système telle que la réponse en sortie "suive au mieux" une trajectoire de consigne correspondant au comportement désiré en sortie du système, ce qui correspond à une inversion du modèle. Dans ce domaine, l'approche proposée dans [Cohen et al., 1989] permet d'établir *la plus grande* trajectoire d'entrée, notée u , telle que la réponse, notée y , soit supérieure ou égale à une trajectoire de consigne, notée z . Plus précisément, pour un système Σ , on cherche le plus grand élément de l'ensemble suivant :

$$\{u \in \mathcal{D}^{\mathbb{Z}} \mid y = \Sigma(u) \preceq z\}, \quad (3.1)$$

où \preceq est l'ordre défini dans le dioïde \mathcal{D} .

Une telle commande est optimale vis-à-vis du critère de juste-à-temps¹. Elle est notée u_{opt} , et se définit par :

$$u_{opt} = \bigoplus_{\Sigma(u) \preceq z} u.$$

Différents travaux ont permis d'étendre la commande en juste-à-temps des systèmes linéaires aux cas où :

- le modèle ne reflète pas le comportement réel du système (l'existence d'une désadaptation entre le système et son modèle) [Boimond and Ferrier, 1996, Menguy et al., 2000a] [Menguy, 1997, §6] ;
- des entrées sont non maîtrisables [Menguy et al., 2000b][Menguy, 1997, §5.4.2] ;

¹"Juste-à-temps" s'interprète comme la quantité juste à l'instant voulu.

- le système est linéaire mais non stationnaire [Lahaye, 2000].

Dans la suite, on s'intéresse uniquement au problème de commande par *poursuite de trajectoire*. Nous montrons que la synthèse de cette commande, établie pour des systèmes $(\min, +)$ linéaires, se généralise pour des systèmes non linéaires n'admettant pas une représentation linéaire dans le dioïde $(\min, +)$, cette non linéarité étant due à la présence éventuelle de poids sur les arcs des graphes considérés.

Ce chapitre est organisé comme suit : après une brève description dans la section 3.1 sur le calcul de la relation entrée-sortie, illustrée à travers un exemple, la section 3.2 est consacrée à la commande en juste-à-temps. Une extension de cette commande aux GETHM sera présentée dans la section 3.3. Ce chapitre rassemble les contributions présentées dans [Hamaci et al., 2004a, Hamaci et al., 2005a].

3.1 Relation entrée-sortie dans le dioïde $\mathcal{D}_{\min}[[\delta]]$

Comme il a été vu dans le §2.2.3, l'évolution dynamique d'un graphe d'événements temporisé discret avec mutliplieurs peut être représentée par les équations d'états (1.25) dans le dioïde $\mathcal{D}_{\min}[[\delta]]$. Aussi, en appliquant le théorème 1.13, on peut déduire que la sortie Y s'exprime comme :

$$Y = HU \quad \text{où} \quad H = D \oplus CA^*B \quad \text{avec} \quad A^* = e \oplus A \oplus A^2 \oplus \dots \quad (3.2)$$

Les éléments de la matrice H appartiennent au dioïde $\mathcal{D}_{\min}[[\delta]]$, chaque élément étant une composition d'opérateurs utilisés pour la modélisation des GETDM.

Dans la majorité des cas, le calcul de la matrice A^* n'est pas simple, du fait de la non commutativité entre l'opérateur μ_m et l'opérateur γ^ν . Aussi le calcul de la sortie Y se fait de façon manuelle par simplification .

Exemple 3.1. L'évolution dynamique du GETDM de la figure 3.2 est donnée par les équations récurrentes suivantes dans le dioïde $\mathcal{D}_{\min}[[\delta]]$:

$$\begin{cases} N_1 &= \gamma^3 \delta^2 N_3 \oplus U, \\ N_2 &= \mu_{1/3} \delta^2 \mu_2 N_1 \oplus \mu_{1/3} \gamma^6 \delta^2 \mu_2 N_3, \\ N_3 &= \delta \mu_3 N_2, \\ Y &= N_2. \end{cases}$$

En remplaçant N_1 par son expression dans l'équation de N_2 , on obtient :

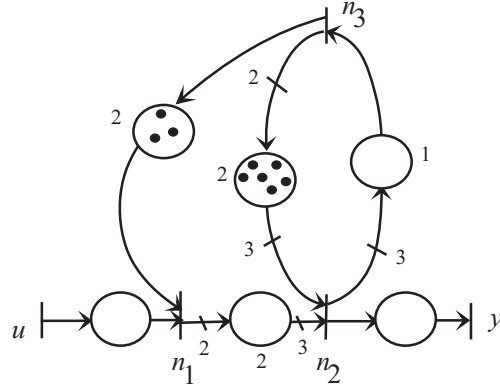


Figure 3.2 – GETDM.

$$N_2 = \mu_{1/3} \delta^2 \mu_2 \gamma^3 \delta^2 N_3 \oplus \mu_{1/3} \delta^2 \mu_2 U \oplus \mu_{1/3} \gamma^6 \delta^2 \mu_2 N_3,$$

de même, en remplaçant N_3 par son expression, on obtient :

$$N_2 = \mu_{1/3} \delta^2 \mu_2 \gamma^3 \delta^2 \delta \mu_3 N_2 \oplus \mu_{1/3} \delta^2 \mu_2 U \oplus \mu_{1/3} \gamma^6 \delta^2 \mu_2 \delta \mu_3 N_2.$$

La propriété (cf. §2.2.3) concernant la commutation de l'opérateur μ_m avec les opérateurs γ^ν et δ^τ , permet d'écrire :

$$N_2 = (\mu_2 \gamma \delta^5 \oplus \mu_2 \gamma \delta^3) N_2 \oplus \mu_{2/3} \delta^2 U,$$

en effet on a : $\gamma^3 \mu_3 = \mu_3 \gamma$ et $\gamma^6 \mu_6 = \mu_6 \gamma$.

Sachant que $\mu_2 \gamma \delta^5 \succeq \mu_2 \gamma \delta^3$ et par application du théorème 1.25, on obtient :

$$Y = (\mu_2 \gamma \delta^5)^* \mu_{(2/3)} \delta^2 U,$$

ce qui donne l'expression de la relation entrée-sortie : $H = (\mu_2 \gamma \delta^5)^* \mu_{(2/3)} \delta^2$,

soit dans l'algèbre classique, l'expression suivante :

$$y(t) = \min(2 + 2y(t - 5), \lfloor \frac{2}{3}u(t - 2) \rfloor).$$

3.2 Commande en juste-à-temps des graphes d'événements temporisés discrets avec multiplieurs

Dans la section précédente, on a calculé la sortie produite pour une entrée donnée *via* la relation entrée-sortie du modèle. Ne disposant pas de moyen systématique pour ce calcul, la commande en juste-à-temps, basée sur le calcul itératif du co-état, présente une alternative pour piloter un GETDM admettant un modèle d'état satisfaisant le système d'équations (2.13) dans le dioïde $\mathcal{D}_{\min}[\delta]$. Dans ce dioïde l'ordre est défini comme l'inverse de l'ordre usuel (par exemple, $2 \succeq 3$), par conséquent la commande optimale en juste-à-temps correspond, dans l'algèbre classique, à la plus petite commande U , notée U_{opt} , telle que la sortie Y soit supérieure ou égale au comportement désiré Z , ce qui correspond à :

$$U_{opt}(t) = \text{Inf}\{U \mid \forall t \in \mathbb{Z}, Y(t) \geq Z(t)\}.$$

La synthèse de cette commande est un problème de pseudo-inverse que la théorie de la résiduation [Blyth and Janowitz, 1972] permet de résoudre directement. En effet, la commande optimale, au sens du juste-à-temps, est solution d'un système d'équations, qui correspond au modèle d'état inversé, appelées *équations d'état backward*.

3.2.1 Equations backward dans le dioïde $\mathcal{D}_{\min}[\delta]$

Considérons un système décrit dans le dioïde $\mathcal{D}_{\min}[\delta]$ par (2.13) ou (3.2). Soit Z une trajectoire désirée en sortie. La plus grande commande U_{opt} (au sens de \preceq) telle que

$$Y = HU_{opt} \stackrel{def}{=} (CA^*B \oplus D)U_{opt} \preceq Z, \quad (3.3)$$

est donnée par la relation suivante :

$$U_{opt} = H^\sharp(Z) = (CA^*B \oplus D)\backslash Z. \quad (3.4)$$

Pour calculer cette commande, on utilise la proposition suivante qui donne le système d'équations *backward* correspondant à la représentation de H^\sharp définie par (3.4).

Proposition 3.2. [Baccelli et al., 1992] *La commande U_{opt} est la plus grande solution (au sens de \preceq) des équations suivantes :*

$$\begin{cases} \xi & = A\backslash\xi \wedge C\backslash Z, & (i) \\ U_{opt} & = B\backslash\xi \wedge D\backslash Z, & (ii) \end{cases} \quad (3.5)$$

où les matrices A, B, C, D correspondent aux matrices d'états du GETDM données par la représentation

(2.13). ξ est appelé le *co-état*, il regroupe les compteurs associés aux transitions du modèle *backward*. Il correspond au franchissement le plus tardif possible des transitions du GETDM tel que la sortie $Y \succeq Z$.

Preuve 1. Nous avons

$$U_{opt} = (CA^*B \oplus D) \backslash Z.$$

Les équations (1.9) et (1.10) permettent d'écrire :

$$\begin{aligned} U_{opt} &= (A^*B) \backslash (C \backslash Z) \wedge D \backslash Z, \\ &= B \backslash (A^* \backslash (C \backslash Z)) \wedge D \backslash Z. \end{aligned}$$

D'après le théorème 1.27, $\xi = A^* \backslash (C \backslash Z)$ est la plus grande solution de (3.5.i), ce qui correspond au système d'équations *backward* (3.5).

Si le vecteur d'état N du GETDM de dimension n , et U (resp. Y) est de dimension m (resp., p), alors par l'utilisation du théorème 1.24, le modèle *backward* (3.5) satisfait les équations suivantes :

$$\begin{cases} \forall i = 1, \dots, n. & \xi_i &= \left(\bigwedge_{j=1}^n A_{ji} \backslash \xi_j \right) \wedge \left(\bigwedge_{r=1}^p C_{ri} \backslash Z_r \right), \\ \forall l = 1, \dots, m. & U_{opt_l} &= \left(\bigwedge_{j=1}^n B_{jl} \backslash \xi_j \right) \wedge \left(\bigwedge_{r=1}^p D_{rl} \backslash Z_r \right). \end{cases} \quad (3.6)$$

Aux équations *backward* (3.5), on associe la condition "initiale" suivante :

$$\exists t_f \text{ telle que } z(t) = z(t_f) \text{ et } \xi(t) = \xi(t_f), \forall t > t_f.$$

Cette condition "initiale" signifie qu'on ne désire contrôler la production que jusqu'à l'instant t_f , ceci peut se justifier, par exemple, si on ne dispose pas de l'information sur la production désirée, au delà de l'instant t_f . Cette information manquante est alors fixée à la valeur $z(t_f)$.

La proposition suivante donne les résiduées des opérateurs discrets γ, δ, μ , on donne également les expressions des résiduées des opérateurs I et P utilisées pour les démonstrations.

Proposition 3.3. *Les résiduées des opérateurs γ, δ, μ sont données par :*

1. $\gamma^{\nu\#} : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{n(t) - \nu\}_{t \in \mathbb{R}}, \nu \in \mathbb{Z} \cup \{+\infty\},$
2. $\delta^{\tau\#} : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{n(t + \tau)\}_{t \in \mathbb{R}}, \tau \in \mathbb{Z},$

Les résiduées des opérateurs I (injection canonique) et P (projection canonique) sont données par :

3. $I^\# : \overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\lceil n(t) \rceil\}_{t \in \mathbb{R}},$
4. $P^\# : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{n(t)\}_{t \in \mathbb{R}}, \text{ soit } P^\# = I.$

5. $\mu_m^\sharp : \overline{\mathbb{Z}}^{\mathbb{R}} \rightarrow \overline{\mathbb{Z}}^{\mathbb{R}}, \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\lceil \frac{1}{m} \times n(t) \rceil\}_{t \in \mathbb{R}}, \quad m \in \mathbb{Q}^+ (\lceil \alpha \rceil \text{ correspond à la partie entière supérieure de } \alpha).$

Preuve 2.

• **Point 1** : Pour vérifier que γ^{ν^\sharp} est bien la résiduée de l'opérateur γ^ν , on utilise le théorème 1.18, en effet :

$$\forall n \in (\mathbb{Z} \cup \{\pm\infty\})^{\mathbb{R}}, \gamma^\nu \gamma^{\nu^\sharp}(n(t)) = \gamma^\nu(n(t) - \nu) = \nu + n(t) - \nu = n(t),$$

$$\text{et } \gamma^{\nu^\sharp} \gamma^\nu(n(t)) = n(t) - \nu + \nu = n(t).$$

• **Point 2** : Comme pour le point 1, en utilisant le théorème 1.18, on a :

$$\forall n \in (\mathbb{Z} \cup \{\pm\infty\})^{\mathbb{R}}, \delta^\tau \delta^{\tau^\sharp}(n(t)) = \delta^\tau(n(t + \tau)) = n(t + \tau - \tau) = n(t),$$

$$\text{et } \delta^{\tau^\sharp} \delta^\tau(n(t)) = n(t + \tau - \tau) = n(t).$$

• **Point 3** : $\forall n \in (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}}, I I^\sharp(n(t)) = I(\lceil n(t) \rceil) \preceq n(t)$

$$\text{et } \forall n \in (\mathbb{Z} \cup \{\pm\infty\})^{\mathbb{R}}, I^\sharp I(n(t)) = \lceil n(t) \rceil = n(t).$$

• **Point 4** : $\forall n \in (\mathbb{Z} \cup \{\pm\infty\})^{\mathbb{R}}, P P^\sharp(n(t)) = P I(n(t)) = \lfloor n(t) \rfloor = n(t)$

$$\text{et } \forall n \in (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}}, P^\sharp P(n(t)) = I P(n(t)) = \lfloor n(t) \rfloor \succeq n(t).$$

• **Point 5** : On pose $\mu_m = P \mu'_m I$, le théorème 1.19 permet d'écrire : $\mu_m^\sharp = (P \mu'_m I)^\sharp = I^\sharp \mu'_m{}^\sharp P^\sharp$, sachant que μ'_m est défini comme suit :

$$\begin{aligned} \mu'_m : (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}} &\rightarrow (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}}, \\ \{n(t)\}_{t \in \mathbb{R}} &\mapsto \{m \times n(t)\}_{t \in \mathbb{R}}. \end{aligned}$$

La résiduée de l'opérateur μ'_m est donnée par :

$$\begin{aligned} \mu'_m{}^\sharp : (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}} &\rightarrow (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}}, \\ \{n(t)\}_{t \in \mathbb{R}} &\mapsto \{\frac{1}{m} \times n(t)\}_{t \in \mathbb{R}}. \end{aligned}$$

On a $\forall n \in (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}}$, $\mu'_m(\mu_m^\sharp(n(t))) = \mu'_m(\frac{1}{m}n(t)) = n(t) \preceq n(t)$

et $\forall n \in (\mathbb{R} \cup \{\pm\infty\})^{\mathbb{R}}$, $\mu_m^\sharp \mu'_m(n(t)) = n(t) = n(t)$.

Remarque 3.4. L'opérateur μ'_m correspond à l'opérateur μ_{m_c} utilisé pour modéliser les GETCM. La résiduée d'opérateur I et P se prouve par l'utilisation du théorème 1.18.

Donc, on peut déduire l'application résiduée de l'opérateur μ_m , on a :

$$\mu_m^\sharp = (P\mu'_m I)^\sharp = I^\sharp \mu'_m^\sharp P^\sharp, \text{ c'est-à-dire } \forall n \in (\mathbb{Z} \cup \{\pm\infty\})^{\mathbb{Z}},$$

$$\mu_m^\sharp n(t) = \lceil \mu'_m^\sharp I(n(t)) \rceil = \lceil \frac{1}{m} \times n(t) \rceil.$$

3.2.2 Exemple

Considérons le GETDM donné par la figure 3.2. Dans le dioïde $\mathcal{D}_{\min}[\delta]$, ce graphe admet la représentation d'état suivante :

$$\left\{ \begin{array}{l} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} = \begin{pmatrix} \varepsilon & \varepsilon & \gamma^3 \delta^2 \\ \mu_{1/3} \delta^2 \mu_2 & \varepsilon & \mu_{1/3} \gamma^6 \delta^2 \mu_2 \\ \varepsilon & \delta \mu_3 & \varepsilon \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} U, \\ \\ Y = \begin{pmatrix} \varepsilon & e & \varepsilon \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix}. \end{array} \right. \quad (3.7)$$

Utilisant le modèle *backward* (3.6), l'expression de la commande optimale u_{opt} est donnée par la plus grande solution (au sens de \preceq) des équations *backward* suivantes :

$$\left\{ \begin{array}{l} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \begin{pmatrix} (\mu_{(1/3)} \delta^2 \mu_2) \xi_2 \\ (\delta \mu_3) \xi_3 \wedge Z \\ (\gamma^3 \delta^2) \xi_1 \wedge (\mu_{(1/3)} \gamma^6 \delta^2 \mu_2) \xi_2 \end{pmatrix}, \\ \\ U_{opt} = \xi_1, \end{array} \right.$$

soit dans l'algèbre classique :

$$\left\{ \begin{array}{l} \left(\begin{array}{l} \xi_1(t) \\ \xi_2(t) \\ \xi_3(t) \end{array} \right) = \left(\begin{array}{l} \lceil 3/2 \times \xi_2(t+2) \rceil \\ \max(\lceil 1/3 \times \xi_3(t+1) \rceil, z(t)) \\ \max(\xi_1(t+2) - 3, \lceil 3/2 \times \xi_2(t+2) - 6/2 \rceil) \end{array} \right), \\ u_{opt}(t) = \xi_1(t). \end{array} \right.$$

Pour les conditions initiales suivantes, on considère que :

$$\forall t \leq 0, z(t) = z(0) = 0, \text{ et } \forall t > t_f = 15, z(t) = z(t_f) = 9 \text{ et } \xi(t) = \xi(t_f),$$

pour une trajectoire désirée z , la commande u_{opt} et la sortie y du modèle sont données par le tableau suivant :

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
z	0	0	0	0	1	1	1	2	2	3	4	5	6	7	8	9	9
u	0	2	2	2	3	3	5	5	6	8	9	11	10	14	14	14	14
y	0	0	0	1	1	1	2	2	3	3	4	5	6	7	8	9	9

On remarque que la sortie y est effectivement supérieure où égale à la trajectoire désirée z (cf. figure 3.4).

La trajectoire u_{opt} est représentée dans la figure 3.3.

3.3 Commande en juste-à-temps des graphes d'événements temporisés hybrides avec multiplieurs

3.3.1 Modèle d'état *backward* hybride

Comme il a été vu au chapitre 2, l'évolution dynamique d'un GETHM peut être représentée par les équations d'états (2.20). Comme dans le cas des GETDM, on cherche à calculer la commande (u_{opt}) telle que la réponse en sortie (y) *suive au mieux* une trajectoire de consigne correspondant au comportement désiré en sortie (z) du système.

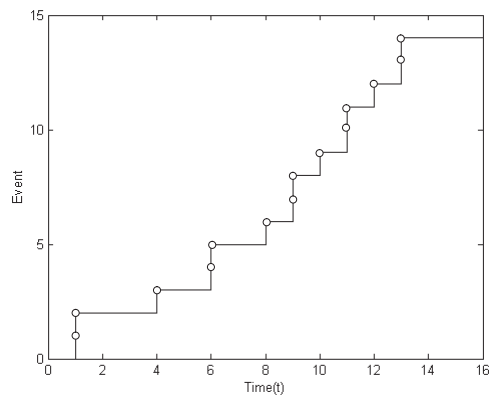


Figure 3.3 – Trajectoire u_{opt} .

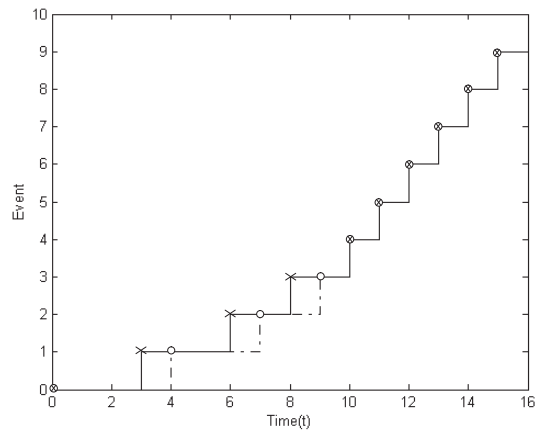


Figure 3.4 – Sortie y (trait épais) et sortie désirée z (pointillé).

Remarque 3.5. Dans cette section, les opérateurs indicés par "d" modélisent la partie discrète du GETHM, et les opérateurs indicés par "c" modélisent la partie continue. Comme mentionné précédemment dans le chapitre 2, l'opérateur P modélise l'interface CD (Continue-Discrète) et l'opérateur I modélise l'interface DC (Discrète-Continue).

Proposition 3.6. *La commande u_{opt} d'un GETHM est la plus grande solution (au sens de \preceq) des équations backward suivantes :*

$$\begin{cases} \begin{pmatrix} \xi_c \\ \xi_d \end{pmatrix} \\ \begin{pmatrix} U_c \\ U_d \end{pmatrix} \end{cases} = \begin{cases} \begin{pmatrix} A_{CC} \backslash \xi_c \wedge A_{DC} \backslash (I \xi_d) \wedge C_{CC} \backslash Z_c \wedge C_{DC} \backslash (I Z_d) \\ [A_{CD} \backslash \xi_c] \wedge A_{DD} \backslash \xi_d \wedge [C_{CD} \backslash Z_c] \wedge C_{DD} \backslash Z_d \end{pmatrix} \\ \begin{pmatrix} B_{CC} \backslash \xi_c \wedge B_{DC} \backslash (I \xi_d) \wedge D_{CC} \backslash Z_c \wedge D_{DC} \backslash (I Z_d) \\ [B_{CD} \backslash \xi_c] \wedge B_{DD} \backslash \xi_d \wedge [D_{CD} \backslash Z_c] \wedge D_{DD} \backslash Z_d \end{pmatrix} \end{cases}.$$

Preuve 3. U_{opt} est la plus grande solution des équations (3.5), sachant que les matrices A, B, C et D correspondent aux matrices d'états données par (2.20). On a :

$$\begin{aligned} A \backslash \xi &= \begin{pmatrix} A_{CC} & A_{CD} \circ I \\ P \circ A_{DC} & A_{DD} \end{pmatrix} \backslash \begin{pmatrix} \xi_c \\ \xi_d \end{pmatrix} = \begin{pmatrix} A_{CC}^\#(\xi_c) \wedge (P \circ A_{DC})^\#(\xi_d) \\ (A_{CD} \circ I)^\#(\xi_c) \wedge A_{DD}^\#(\xi_d) \end{pmatrix} \\ &= \begin{pmatrix} A_{CC}^\#(\xi_c) \wedge A_{DC}^\# \circ I(\xi_d) \\ I^\# A_{CD}^\#(\xi_c) \wedge A_{DD}^\#(\xi_d) \end{pmatrix}, \end{aligned}$$

ce qui donne

$$A \backslash \xi = \begin{pmatrix} A_{CC} \backslash \xi_c \wedge A_{DC} \backslash (I \xi_d) \\ [A_{CD} \backslash \xi_c] \wedge A_{DD} \backslash \xi_d \end{pmatrix}.$$

En suivant le même raisonnement pour le calcul de $C \backslash Z$, $B \backslash \xi$ et $D \backslash Z$, on trouve le modèle *backward* associé au GETHM donné par la proposition 3.6.

Proposition 3.7. Les opérateurs continus utilisés pour modéliser le GETHM sont résiduables, leurs résiduées sont définies de $\overline{\mathbb{R}}^{\mathbb{R}} \rightarrow \overline{\mathbb{R}}^{\mathbb{R}}$ par :

1. $\gamma_c^{\nu\sharp} : \{n(t)\}_{t \in \mathbb{R}} \mapsto \{n(t) - \nu\}_{t \in \mathbb{R}}$ tel que $\nu \in \mathbb{R} \cup \{+\infty\}$,
2. $\delta_c^{\tau\sharp} : \{n(t)\}_{t \in \mathbb{R}} \mapsto \{n(t + \tau)\}_{t \in \mathbb{R}}$ tel que $\tau \in \mathbb{R}$,
3. $\mu_{m_c}^{\sharp} : \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\frac{1}{m} \times n(t)\}_{t \in \mathbb{R}}$ tel que $m \in \mathbb{R}^+$,
4. $\omega_V^{\sharp} : \{n(t)\}_{t \in \mathbb{R}} \mapsto \{\sup_{h \geq t} (n(h) - V \times (h - t))\}_{t \in \mathbb{R}}$ tel que $V \in \mathbb{R}^+ \cup \{+\infty\}$.

Preuve 4. Les points 1, 2 et 3 se prouvent de la même façon que les résiduées des opérateurs discrets γ, δ, μ données par la proposition 3.3.

• **Point 4 :** Soit $n(t)$ un signal non décroissant, tel que $\forall t \geq t_f \ n(t) = n(t_f)$.

Pour tout $h \geq t_f, n(h) = n(t_f)$.

ω_V est une application croissante, aussi pour un t donné, on a

$\omega_V(h - t) \geq \omega_V(t_f - t) \Rightarrow -\omega_V(h - t) \leq -\omega_V(t_f - t)$, ceci implique que :

$$n(t_f) - \omega_V(t_f - t) \geq n(t_f) - \omega_V(h - t) \Rightarrow \sup_{h \in \mathbb{R}} [n(h) - \omega_V(h - t)] = \sup_{h \leq t_f} [n(h) - \omega_V(h - t)].$$

Pour $h \leq t$, on a : $h - t \leq 0 \Rightarrow \omega_V(h - t) = 0$, ceci implique que :

$$n(t) - \omega_V(t - t) \geq n(h) - \omega_V(h - t) \Rightarrow \sup_{h \in \mathbb{R}} [n(h) - \omega_V(h - t)] = \sup_{h \geq t} [n(h) - \omega_V(h - t)].$$

Au final, on obtient :

$$\omega_V^{\sharp}(n(t)) = \sup_{h \in \mathbb{R}} [n(h) - \omega_V(h - t)] = \sup_{t \leq h \leq t_f} [n(h) - \omega_V(h - t)], \text{ sachant que } \omega_V(h - t) = V \times (h - t).$$

3.3.2 Exemple

Considérons le GETHM donné par la figure 3.5. L'évolution dynamique du graphe est donnée par :

$$\left\{ \begin{array}{l} \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} = \begin{pmatrix} \varepsilon_c & \omega_1 \gamma_c^2 I \\ P \delta_c^1 & \gamma_d^1 \delta_d^1 \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \oplus \begin{pmatrix} \omega_1 \gamma_c^1 \mu_{(2)_c} & \omega_1 \mu_{(1/2)_c} \delta_c^1 I \\ P \varepsilon_c & \varepsilon_d \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}, \\ (Y) = (P \varepsilon_c \quad \varepsilon_d) \begin{pmatrix} N_1 \\ N_2 \end{pmatrix} \oplus (P \varepsilon_c \quad \varepsilon_d) \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}. \end{array} \right. \quad (3.8)$$

Pour un comportement désiré en sortie, noté Z , l'expression de la commande optimale est donnée par la plus grande solution (au sens de \preceq) des équations *backward* suivantes :

$$\left\{ \begin{array}{l} \begin{pmatrix} \xi_{1c} \\ \xi_{2d} \end{pmatrix} = \begin{pmatrix} \varepsilon_c \backslash \xi_{1c} \wedge \delta_c^1 \backslash (I\xi_{2d}) \wedge \varepsilon_c \backslash (IZ_d) \\ \lceil (\omega_1 \gamma_c^2) \backslash \xi_{1c} \rceil \wedge (\gamma_d^1 \delta_d) \backslash \xi_{2d} \wedge e_d \backslash Z_d \end{pmatrix} = \begin{pmatrix} \delta_c^1 \backslash (I\xi_{2d}) \\ \lceil (\omega_1 \gamma_c^2) \backslash \xi_{1c} \rceil \wedge (\gamma_d^1 \delta_d) \backslash \xi_{2d} \wedge Z_d \end{pmatrix}, \\ \begin{pmatrix} U_{1c} \\ U_{2d} \end{pmatrix} = \begin{pmatrix} (\omega_1 \gamma_c^1 \mu_{2c}) \backslash \xi_{1c} \wedge \varepsilon_c \backslash (I\xi_{2d}) \wedge \varepsilon_c \backslash (IZ_d) \\ \lceil (\omega_1 \mu_{(1/2c)} \delta_c) \backslash \xi_{1c} \rceil \wedge \varepsilon_d \backslash \xi_{2d} \wedge \varepsilon_d \backslash Z_d \end{pmatrix} = \begin{pmatrix} (\omega_1 \gamma_c^1 \mu_{2c}) \backslash \xi_{1c} \\ \lceil (\omega_1 \mu_{(1/2c)} \delta_c) \backslash \xi_{1c} \rceil \end{pmatrix}, \end{array} \right.$$

soit dans l'algèbre usuelle :

$$\left\{ \begin{array}{l} \begin{pmatrix} \xi_{1c}(t) \\ \xi_{2d}(t) \end{pmatrix} = \begin{pmatrix} \xi_{2d}(t+1) \\ \max(\lceil \omega_1^\# \xi_{1c}(t) - 2 \rceil, \xi_{2d}(t+1) - 1, z_d(t)) \end{pmatrix}, \\ \begin{pmatrix} u_{1c}(t) \\ u_{2d}(t) \end{pmatrix} = \begin{pmatrix} (\omega_1^\# \xi_{1c}(t) - 1) \times (1/2) \\ \lceil (\omega_1^\# \xi_{1c}(t+1)) \times 2 \rceil \end{pmatrix}. \end{array} \right.$$

Pour calculer $u_{1c}(t)$ et $u_{2d}(t)$, on pose les conditions initiales suivantes :

$$\forall t < 10, z(t) = 0, \text{ et } \forall t > t_f = 25, z_d(t) = z_d(t_f) = 14 \text{ et } \xi(t) = \xi(t_f),$$

sachant que la trajectoire désirée z_d est donnée par le tableau suivant :

t	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
z_d	0	0	1	1	2	2	3	3	4	4	5	5	6	6	8	10	12	14	14	14

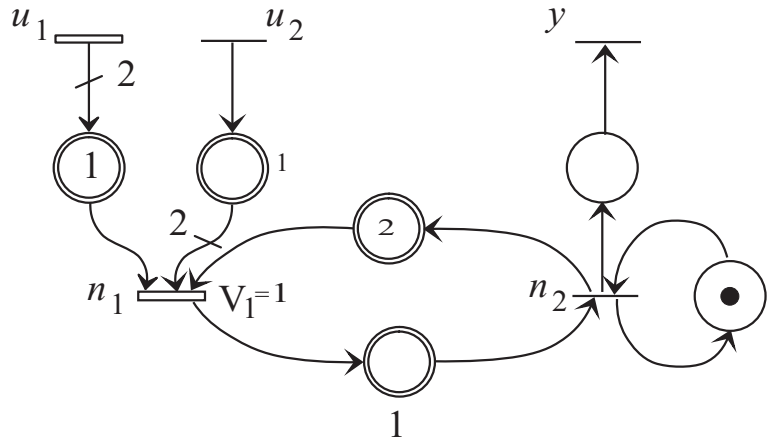


Figure 3.5 – GETHM à commander.

Par simulation, on obtient les résultats suivants :

Les commandes u_{1c} et u_{2d} obtenues par l'application de la consigne z_d à l'entrée du modèle *backward*, sont représentées sur la figure 3.6. La différence entre l'évolution de ces deux commandes est due principalement à la différence des poids sur l'arcs reliant u_{1c} à n_1 et u_{2d} à n_1 . L'influence de la partie entière supérieure sur la commande discrète, u_{2d} , explique l'évolution rapide de cette dernière par rapport à la commande continue, u_{1c} .

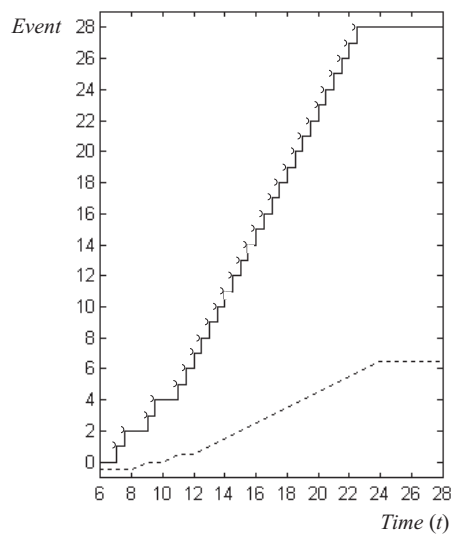


Figure 3.6 – Trajectoires de commande u_{1c} (pointillé) et u_{2d} (trait épais).

L'application des commandes u_{1c} et u_{2d} sur le GETHM décrit par le système d'équations (3.8), donne une trajectoire de sortie y supérieure ou égale (au sens de l'ordre \preceq qui est l'inverse de l'ordre usuel) à la trajectoire du comportement désirée en sortie, z , décrite par la figure 3.7.

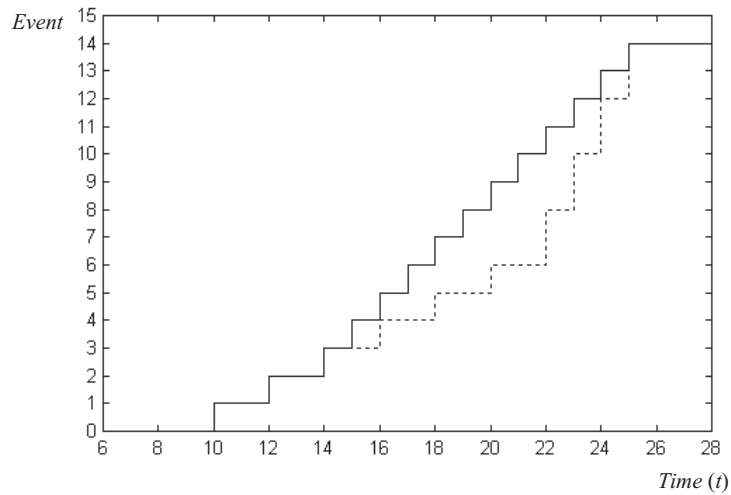


Figure 3.7 – La sortie désirée z (pointillé) et la sortie y (trait épais).

La figure 3.8 représente une comparaison entre l'évolution de la variable d'état n_1 et le co-état ξ_{1c} . La variable ξ_{1c} se comporte comme une variable discrète, ceci s'explique du fait de sa dépendance à la variable discrète ξ_{2d} .

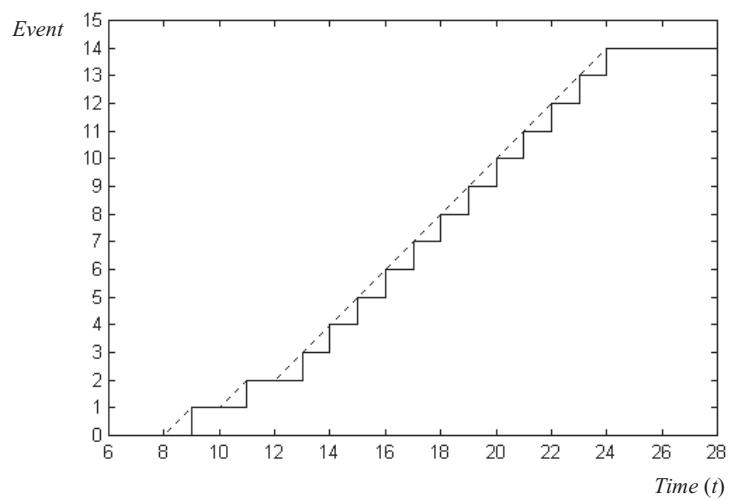


Figure 3.8 – Le co-état ξ_{1c} (trait épais) et l'état n_1 (pointillé).

3.4 Conclusion

Dans ce chapitre, nous avons abordé le problème de commande des GETDM admettant une représentation d'état dans le dioïde d'opérateur $\mathcal{D}_{\min}[[\delta]]$. Plus précisément, nous nous sommes intéressés à la synthèse d'une commande dans l'objectif de poursuite, en juste-à-temps, de trajectoire. La commande proposée est la plus grande commande (au sens \preceq) telle que la réponse soit inférieure ou égale à une trajectoire de consigne, correspondant au comportement désiré en sortie du système. Une extension au GETHM a été proposée.

CHAPITRE 4

Analyse de performances des graphes d'événements temporisés discrets avec multiplieurs

De manière analogue aux GET, l'analyse de performances des GETDM revient à calculer le temps de cycle du graphe et la fréquence de franchissement associée à chaque transition du graphe. Par exemple, pour un système de production, ces performances correspondent aux calculs : du temps de cycle (temps séparant deux pièces produites) et de la fréquence de franchissement (au nombre de pièces produites par unité de temps, par une machine). Si la théorie spectrale permet de calculer ces performances pour les GET, le problème est toujours ouvert pour les GETDM. L'étude bibliographique effectuée dans ce domaine, a montré qu'il n'existait pas de méthode analytique générale permettant d'évaluer ces performances directement à partir des GETDM. Les méthodes existantes se basent, soit sur la linéarisation de ces graphes [Munier, 1993], soit sur des conditions restrictives sur le marquage initial [Chao et al., 1993].

L'objectif de ce chapitre est d'analyser les performances des GETDM. Il est organisé comme suit : après un rappel sur les éléments de la théorie des graphes, une application sur l'évaluation de performances des GET est présentée. Avant d'exposer notre contribution dans ce domaine, un état de l'art sur les travaux existants est proposé. Afin d'appliquer certains résultats de base de la théorie spectrale pour l'analyse de performances des GETDM, deux nouvelles méthodes de linéarisation seront présentées. Ces méthodes ont pour but de linéariser le modèle mathématique associé au comportement d'un GETDM, qui n'admet pas une représentation linéaire dans le dioïde $(min, +)$, ceci afin d'obtenir un GET ordinaire équivalent.

La première méthode, inspirée des travaux de Guy Cohen [Cohen et al., 1998c], a pour principe de linéariser tout GET discret avec multiplieurs vérifiant une condition de linéarisation sur le marquage initial (*cf.* proposition 4.25). Dans le cas où la condition est satisfaite, les performances du GETDM sont déterminées de façon exacte. Dans le cas contraire, une méthode de linéarisation, dite approchée par ajout/retrait de jetons dans le graphe, est introduite. Cette alternative permet d'encadrer les performances du GETDM entre deux bornes : une valeur inférieure obtenue *via* un retrait de jetons et une valeur supérieure obtenue *via* un ajout de jetons dans le graphe .

La deuxième méthode est inspirée des travaux de Benoît Trouillet [Trouillet et al., 2002], elle concerne les GETDM admettant un T-invariant dont au moins une de ses composantes est égale à 1. Cette composante est associée à une transition appelée *transition de normalisation*. Elle permet, *via* un changement de variable, d'aboutir à un graphe dont les composantes du T-invariant sont toutes à valeurs unitaires (correspondant à un GET équivalent).

Les résultats de ces travaux ont été publiés dans [Hamaci et al., 2004c, Hamaci et al., 2004a, Hamaci et al., 2004b]. Ce chapitre rassemble les contributions de [Hamaci et al., 2005b, Hamaci et al., 2005a].les contributions présentées .

4.1 Quelques rappels

Dans cette section, on rappelle brièvement un ensemble de définitions de la théorie des graphes utiles pour la compréhension de ce chapitre.

Définition 4.1 (Graphe orienté). Un *graphe orienté* est un graphe défini par deux ensembles : un ensemble de noeuds (ou sommets), noté \mathbb{N} , et un ensemble d'arcs orientés, noté Λ .

Définition 4.2 (Chemin, circuit). Un *chemin*, noté Υ , est une suite de noeuds que l'on peut parcourir successivement en empruntant les arcs de Λ . On note $\xrightarrow{i \ j}$ un chemin allant d'un noeud d'origine, noté i , vers un noeud d'extrémité, noté j (*cf.* figure 4.1).

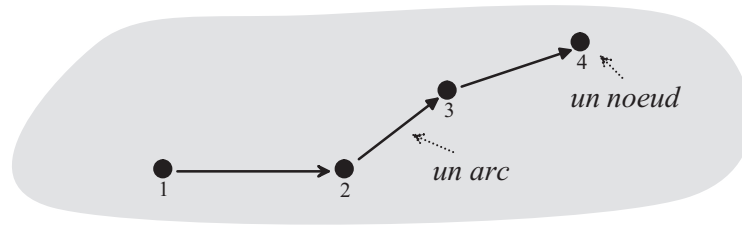


Figure 4.1 – Un chemin.

Un *circuit*, noté ζ , est un chemin dont le noeud d'origine est confondu avec le noeud d'extrémité (cf. figure 4.2).

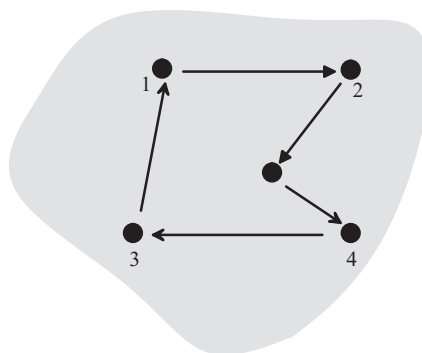


Figure 4.2 – Un circuit.

Définition 4.3 (Composante fortement connexe). Une *composante fortement connexe* est une classe d'équivalence induite par une relation d'équivalence, notée \mathfrak{R} , définie sur \mathfrak{N} comme suit :

$$i \mathfrak{R} j \Leftrightarrow i \xrightarrow{j} \text{ et } j \xrightarrow{i} .$$

Un graphe est *fortement connexe* s'il est composé d'une seule composante fortement connexe (cf. figure 4.3).

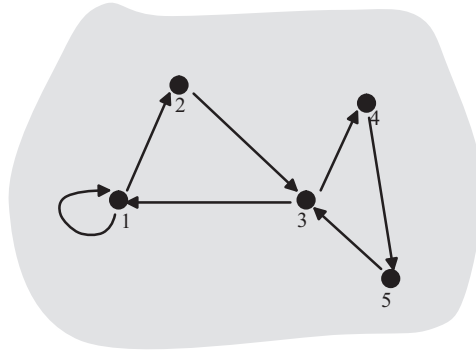


Figure 4.3 – Composante fortement connexe.

Définition 4.4 (Graphe valué). Un *graphe valué* est un graphe pour lequel tout arc $(i, j) \in \Lambda$ est muni d'un poids w_{ji} (notons que l'indice d'extrémité j précède l'indice d'origine i).

Soit $\Upsilon = (i_1, i_2, \dots, i_k)$ un chemin d'un graphe valué. On appelle :

- longueur de Υ , notée $|\Upsilon|_l$, le nombre d'arcs qui le composent ;
- poids de Υ , noté $|\Upsilon|_w$, la somme des poids des arcs qui le composent, soit :

$$|\Upsilon|_w = w_{i_k i_{k-1}} + \dots + w_{i_2 i_1}.$$

Définition 4.5 (Matrice et graphe de précédence). Soit A une matrice de dimension $n \times n$ à coefficients dans un dioïde \mathcal{D} . La matrice A d'un graphe valué à n noeuds, ayant pour élément A_{ij} le poids w_{ij} de l'arc $\overset{j}{\rightarrow} \overset{i}{\leftarrow}$, est appelée *matrice de précédence*.

A toute matrice carrée, par exemple, notée A , on peut associer un *graphe de précédence*, noté $\mathcal{G}(A)$ (cf. figure 4.4).

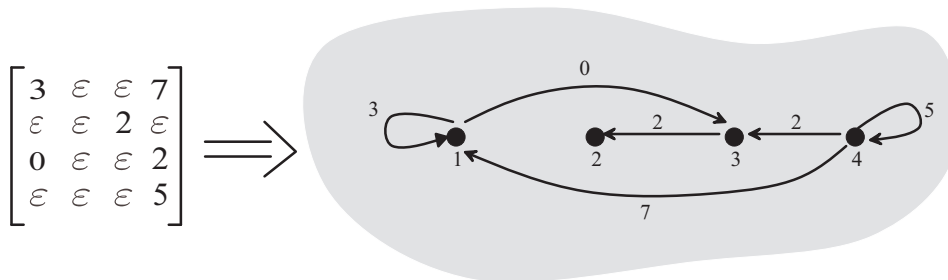


Figure 4.4 – Une matrice et son graphe de précédence associé.

On remarque que le poids d'un arc absent du graphe, est égal à ε .

• **Poids minimal des chemins.** Pour une matrice carrée A de dimension $n \times n$ à élément dans \mathbb{R}_{min} , l'éléments $(A^k)_{ij}$ représente le poids minimal des chemins de longueur k allant du noeud j au noeud i dans son graphe de précédence. Par exemple, pour $k = 2$, on aura :

$$(A^2)_{ij} = \bigoplus_{l=1}^n A_{il} \otimes A_{lj},$$

ce qui est équivalent dans l'algèbre usuelle à :

$$(A^2)_{ij} = \min_{l=1, \dots, n} \{A_{il} + A_{lj}\}.$$

Définition 4.6 (Matrice irréductible). Une matrice *irréductible* $A \in \mathcal{D}^{n \times n}$ est une matrice pour laquelle, pour toute paire (i, j) , il existe un entier k tel que $(A^k)_{ij} \neq \varepsilon$.

Vis-à-vis de son graphe de précédence, une matrice A est *irréductible* si pour toute paire de sommets (i, j) , il existe au moins un chemin allant de j à i , ce qui équivalent à dire que le graphe associé soit fortement connexe.

4.2 Propriétés spectrales et asymptotiques des matrices de $\mathbb{R}_{min}^{n \times n}$

Cette section rappelle les principales propriétés spectrales des matrices à coefficients dans le dioïde $\mathbb{R}_{min}^{n \times n}$, dont le théorème de cyclicité des matrices irréductibles. Enfin, on rappelle comment ces résultats peuvent être appliqués à l'évaluation de performances des GET.

Définition 4.7 (Valeurs propres et vecteurs propres). On appelle valeur propre et vecteur propre d'une matrice $A \in \mathcal{D}^{n \times n}$, le scalaire $\lambda \in \mathbb{R}_{min}$ et le vecteur $v \in \mathbb{R}_{min}^n \setminus \{\varepsilon\}$ tels que :

$$A \otimes v = \lambda \otimes v. \quad (4.1)$$

Le théorème suivant donne une méthode de calcul de la valeur propre d'une matrice irréductible.

Théorème 4.8. [Gondran and Minoux, 1985]

Une matrice $A \in \mathbb{R}_{min}^{n \times n}$ irréductible admet une unique valeur propre, notée λ , égale au rayon spectral de la matrice A , soit :

$$\rho(A) = \bigoplus_{k=1}^n (tr A^k)^{\otimes 1/k},$$

où $tr A^k = \bigoplus_{i=1}^n (A^k)_{ii}$ correspond au poids minimal¹ de tous les circuits de longueur k du graphe $\mathcal{G}(A)$. En définissant le *poids moyen* d'un circuit ζ comme la valeur de son poids, noté $|\zeta|_w$, divisée par sa longueur, noté $|\zeta|_l$; le poids moyen minimal des circuits de longueur k s'obtient alors en divisant $tr A^k$ par k , ce qui est équivalent à l'expression $(tr A^k)^{\otimes 1/k}$ dans le dioïde $\mathbb{R}_{min}^{n \times n}$.

Cette valeur propre peut se déduire directement à partir du graphe $\mathcal{G}(A)$ à travers le résultat donné dans le théorème suivant.

Théorème 4.9. [Gondran and Minoux, 1985] *La valeur propre λ d'une matrice $A \in \mathbb{R}_{min}^{n \times n}$ irréductible est égale au poids moyen minimal des circuits de $\mathcal{G}(A)$, c'est-à-dire :*

$$\lambda = \min_{\zeta \in \mathcal{C}} \frac{|\zeta|_w}{|\zeta|_l},$$

où \mathcal{C} est l'ensemble des circuits de $\mathcal{G}(A)$.

Une matrice irréductible peut admettre plusieurs vecteurs propres associés à la valeur propre λ . Afin de les calculer, on définit, au préalable, la matrice A_λ .

Définition 4.10. Soit $A \in \mathbb{R}_{min}^{n \times n}$ une matrice irréductible de valeur propre λ . On définit la matrice, notée A_λ , par :

$$A_\lambda = \lambda^{-1} \otimes A.$$

Le théorème suivant fournit une condition permettant de tester si une colonne de la matrice A_λ^+ est un vecteur propre de A .

Théorème 4.11. [Gondran and Minoux, 1985] *Soit $A \in \mathcal{D}^{n \times n}$ une matrice irréductible de valeur propre λ . La j -ième colonne de la matrice A_λ^+ , notée $(A_\lambda^+)_{.j}$, est un vecteur propre de A , si elle satisfait l'égalité :*

$$(A_\lambda^+)_{.j} = A_\lambda \otimes (A_\lambda^+)_{.j}.$$

Remarque 4.12. Dans le cas où A est irréductible, toutes les coordonnées d'un vecteur propre sont différentes de ε .

Définition 4.13 (Circuit critique, graphe critique). Soient $A \in \mathbb{R}_{min}^{n \times n}$ une matrice irréductible de valeur propre λ , et $\mathcal{G}(A)$ le graphe de précedence associé.

Un circuit ζ est dit *critique* si son poids moyen est minimum, c'est-à-dire si $\frac{|\zeta|_w}{|\zeta|_l} = \lambda$.

On appelle *graphe critique*, noté $\mathcal{G}^c(A)$, le sous graphe de $\mathcal{G}(A)$ formé des sommets et des arcs appartenant à un circuit critique.

¹ $tr A^k = \bigoplus_{i=1}^n (A^k)_{ii}$ correspond au poids maximal de tous les circuits de longueur k du graphe $\mathcal{G}(A)$ dans le cas du dioïde $\mathbb{R}_{max}^{n \times n}$.

Définition 4.14 (Cyclicité). On appelle :

- *cyclicité d'une composante fortement connexe* le plus grand commun diviseur (pgcd) des longueurs des circuits qui la composent ;
- *cyclicité d'un graphe* le plus petit commun multiple (ppcm) des cyclicités de ses composantes fortement connexes.

Exemple 4.15. Le graphe de précédence représenté par la figure 4.5 admet la matrice irréductible suivante dans le dioïde $\mathbb{R}_{min}^{3 \times 3}$:

$$A = \begin{pmatrix} \varepsilon & \varepsilon & 1 \\ 0 & 2 & 0 \\ \varepsilon & 2 & \varepsilon \end{pmatrix}.$$

Dans cet exemple, le graphe est composé de trois circuits élémentaires :

- le premier circuit : $noeud_1 \rightarrow noeud_2 \rightarrow noeud_3 \rightarrow noeud_1$ a un poids moyen égal à 1,
- le deuxième circuit : $noeud_2 \rightarrow noeud_3 \rightarrow noeud_2$ a un poids moyen égal à 1,
- le troisième circuit : $noeud_2 \rightarrow noeud_2$ a un poids moyen égal à 2.

Aussi, on a

$$\lambda = \min(1, 1, 2) = 1, \text{ ce qui correspond au poids moyen minimal du graphe.}$$

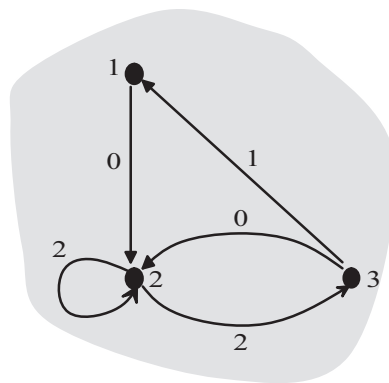


Figure 4.5 – Exemple d'un graphe de précédence fortement connexe.

Comme on a vu précédemment, cette valeur propre peut également être calculée en utilisant le théorème 4.8. On a :

$$A^2 = \begin{pmatrix} \varepsilon & 3 & \varepsilon \\ 2 & 2 & 1 \\ 2 & 4 & 2 \end{pmatrix}, \quad A^3 = \begin{pmatrix} 3 & 5 & 3 \\ 2 & 3 & 2 \\ 4 & 4 & 3 \end{pmatrix}.$$

D'où :

- $tr(A) = \bigoplus_{i=1}^3 A_{ii} = 2,$
- $tr(A^2) = \bigoplus_{i=1}^3 A_{ii}^2 = 2,$
- $tr(A^3) = \bigoplus_{i=1}^3 A_{ii}^3 = 3,$

ainsi, on a :

$$\lambda = \bigoplus_{k=1}^3 (tr A^k)^{\otimes 1/k} = tr(A) \oplus tr(A^2)^{\otimes 1/2} \oplus tr(A^3)^{\otimes 1/3} = 1.$$

Calcul des vecteurs propres associés à la matrice A

On a d'après la définition 4.10:

$$A_\lambda = \lambda^{-1} \otimes A = \begin{pmatrix} \varepsilon & \varepsilon & 0 \\ -1 & 1 & -1 \\ \varepsilon & 1 & \varepsilon \end{pmatrix} \quad (\text{rappel : } \lambda^{-1} \text{ correspond à } -\lambda \text{ dans l'algèbre usuelle),$$

$$\text{d'où } A_\lambda^+ = A_\lambda \oplus A_\lambda^2 \oplus \dots = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}.$$

En appliquant le théorème 4.11, on vérifie que les trois colonnes de la matrice A_λ^+ correspondent bien aux vecteurs propres associés à la valeur propre $\lambda (=1)$ au sens où ils vérifient la relation (4.1).

4.3 Evaluation de performances des Graphes d'événements temporisés dans l'algèbre $(min, +)$

Soit un graphe d'événements temporisé fortement connexe fonctionnant en régime autonome. Un tel graphe se représente par l'équation récurrente $X(t) = A \otimes X(t-1)$ dans l'algèbre $(min, +)$. La valeur propre λ associée à la matrice irréductible A à coefficients dans le dioïde $\mathbb{R}_{min}^{n \times n}$ correspond au *taux de production* asymptotique du GET, défini comme le nombre moyen de tirs par unité de temps. Ce taux de production, qui représente aussi la fréquence de franchissement de chaque transition, est identique pour toutes les transitions du graphe. Pour un système de production, il correspond aux nombres moyens de pièces produites par unité de temps.

Définition 4.16 (Taux de production asymptotique [Gaubert, 1995]).

Soit $x(t)$ le compteur associé à la transition x . Son taux de production asymptotique, noté λ , est défini par :

$$\lambda = \lim_{t \rightarrow +\infty} \frac{x(t)}{t}.$$

Ce taux de production peut être calculé de deux façons : soit en appliquant une approche algébrique à travers le théorème 4.8, soit en utilisant le théorème suivant qui permet de calculer λ à partir du graphe de précédence.

Théorème 4.17. [Gaubert, 1992] *Le taux de production λ d'un GET fortement connexe est caractérisé par :*

$$\lambda = \min_{c \text{ circuit élémentaire}} \frac{N(c)}{T(c)}, \tag{4.2}$$

où $N(c)$ correspond au nombre total de jetons du circuit c , équivalent au poids du circuit du graphe de précédence associé, et $T(c)$ correspond à la somme des temporisations des places du circuit c , équivalent à la longueur du circuit du graphe de précédence associé.

L'inverse de cette valeur propre est également utilisée pour analyser les performances d'un GET. Elle correspond au *temps de cycle moyen* du graphe, c'est-à-dire le temps moyen séparant deux tirs consécutifs d'une transition. Pour un système de production, il correspond au temps moyen séparant deux pièces produites. Ce temps de cycle, noté TC , peut être calculé de la façon suivante :

$$TC = \max_{c \text{ circuit élémentaire}} \frac{T(c)}{N(c)}. \tag{4.3}$$

Exemple 4.18. Les performances du GET de la figure 4.6 sont identiques à celles du graphe de précédence de la figure 4.5 calculées précédemment.

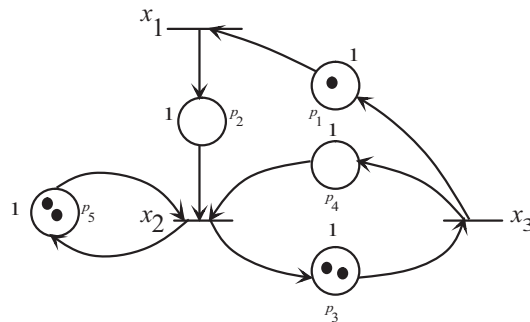


Figure 4.6 – GET équivalent au graphe de précédence de la figure 4.5.

Relativement à l'évolution de l'état du système en régime autonome, si λ est la valeur propre de A et si le vecteur d'état initial $X(0)$ est un vecteur propre de A , on a :

$$\begin{aligned} X(t) &= AX(t-1) \\ &= A^t X(0) \\ &= A^{t-1} \otimes A \otimes X(0) = A^{t-1} \otimes \lambda \otimes X(0) \\ &= \lambda^t \otimes X(0) = \lambda X(t-1), \end{aligned}$$

ce qui est équivalent dans l'algèbre classique à :

$$X(t) = \lambda + X(t-1),$$

autrement dit, l'état du système est incrémenté de λ lorsque t est incrémenté de 1, ce qui signifie que, dès l'état initial, le GET fonctionne, en régime autonome, de façon périodique selon une période égale à 1.

L'établissement d'une valeur propre de A permet d'évaluer le comportement périodique d'un GET en régime autonome, lorsque la condition $X(0)$ correspond à un vecteur propre de A . Le théorème suivant fournit une réponse sur le comportement du graphe pour une condition initiale quelconque.

Théorème 4.19. [Baccelli et al., 1992] Soit $A \in \mathbb{R}_{min}^{n \times n}$ une matrice irréductible. Il existe deux entiers N et c tels que :

$$n \geq N \Rightarrow A^{n+c} = \lambda^c A^n,$$

où c est appelé cyclicité de A .

La matrice A étant irréductible, une fois le régime périodique atteint, on a :

$$\begin{aligned} X(t) &= A^{n+c} X(t-n-c) \\ &= \lambda^c \otimes X(t-c), \end{aligned}$$

autrement dit, en régime autonome, pour t suffisamment grand, il existe $c \in \mathbb{N}$ tel que :

$$X(t) = c \times \lambda + X(t-c).$$

En résumé, on peut dire qu'un GET en régime autonome atteint, après un régime transitoire fini, un régime périodique de période égale à c .

Remarque 4.20 (Evaluation de performances d'un GET dans le dioïde \mathbb{Z}_{min}). Les résultats énoncés ci-dessus dans le dioïde \mathbb{R}_{min} s'appliquent également dans d'autres dioïdes, tels que \mathbb{R}_{max} , \mathbb{Z}_{max} , ou \mathbb{Z}_{min} . Dans la suite de ce chapitre, le dioïde utilisé est le dioïde \mathbb{Z}_{min} . Notons qu'une matrice irréductible à coefficients dans \mathbb{Z}_{min} , peut admettre une valeur propre et des vecteurs propres associés dans le dioïde \mathbb{Q}_{min} . Chaque vecteur propre fournit un régime stationnaire du GET associé, lorsque ce vecteur propre n'est pas entier, cela signifie que le régime stationnaire n'est pas réalisable avec un nombre entier de jetons. Cette difficulté apparaît dans le théorème spectral 4.8, sachant que la valeur propre λ d'une matrice irréductible A est égale au poids moyen minimal de ces circuits, obtenu en divisant $tr A^k$ par k , ce qui n'assure pas l'intégrité de la valeur propre dans le dioïde \mathbb{Z}_{min} . Notons que cette difficulté disparaît dans le formalisme dateurs, où, avoir des dates rationnelles, ou même réelles, reste interprétable avec le comportement du graphe.

Exemple 4.21. Soit le GET représenté par la figure 4.7. Pour le marquage initial : $M(0) = (0, 0, 1, 0, 1)$ et les temporisations sur les places : $(1, 0, 1, 1, 1)$, la valeur propre associée à la matrice du GET est égale à $\lambda = \frac{1}{2}$.

En suivant le même raisonnement que précédemment, nous obtenons le vecteur propre : $v = (-\frac{1}{2}, -\frac{1}{2}, 0)^t$. On remarque bien que certaines composantes du vecteur v sont dans \mathbb{Q}_{min} et non dans \mathbb{Z}_{min} .

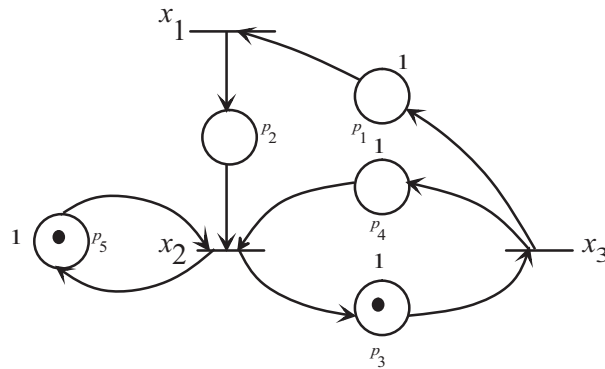


Figure 4.7 – Graphe d'événements temporisé.

Remarque 4.22 (Matrice réductible). Les propriétés spectrales et asymptotiques des matrices à éléments dans un dioïde ont également été étudiées dans le cas *réductible* (non irréductible) [Gaubert, 1992, §4]. Pour une matrice $A \in \mathbb{R}_{min}^{n \times n}$ réductible, l'unicité de la valeur propre n'est pas assurée. Les résultats énoncés dans ce cas s'appuient sur une décomposition du graphe de précedence de A en composantes fortement connexes.

4.4 Evaluation de performances des graphes d'événements temporisés discrets avec multiplieurs

Comme dans le cas des GET, les performances d'un GETDM fortement connexe, sont obtenues dans le cas d'un fonctionnement au plus tôt et lorsque le régime permanent est établi [Munier, 1993]. Par contre, les fréquences de franchissement ne sont pas identiques pour toutes les transitions, mais sont proportionnelles au T-invariant élémentaire (*cf.* définition 2.5). Pour chaque transition, la fréquence de franchissement, notée λ_{m_q} , peut être définie de la façon suivante :

$$\lambda_{m_q} = \frac{\theta_q}{TC_m}, \quad (4.4)$$

où :

- θ_q est la composante du T-invariant associée à la transition n_q ,
- TC_m est le temps de cycle du graphe.

Dans le cas d'un graphe neutre et fortement connexe, ce temps de cycle peut être défini comme suit.

Définition 4.23. [Sauer, 2003] Le *temps de cycle*, noté TC_m , d'un GETDM neutre et fortement connexe, obtenu à partir d'un marquage initial M_0 , est le temps moyen nécessaire pour franchir une fois le T-invariant minimal en mode de fonctionnement au plus tôt à partir de M_0 .

Si le calcul de performances des GET a pu être réalisé *via* grâce à la théorie des graphes, le problème reste, à notre connaissance, ouvert pour les GETDM. La difficulté de ces types de graphes est due à la présence des poids sur les arcs qui empêche l'utilisation du théorème 4.17. L'exemple suivant illustre cette difficulté.

Exemple 4.24. Le GETDM de la figure 4.8 admet le T-invariant suivant : $(1, 2, 2)^t$. Il est composé de deux circuits :

- Le premier circuit, noté ϱ_1 , est composé des places p_1 et p_2 ,
- Le deuxième circuit, noté ϱ_2 , est composé des places p_3 et p_4 .

L'évolution du marquage de ce GETDM est donnée par la figure 4.9. D'après la définition 4.23, le temps de cycle de ce graphe est égal à 4 unités de temps.

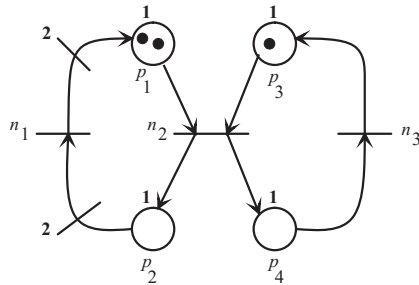


Figure 4.8 – Exemple de GETDM.

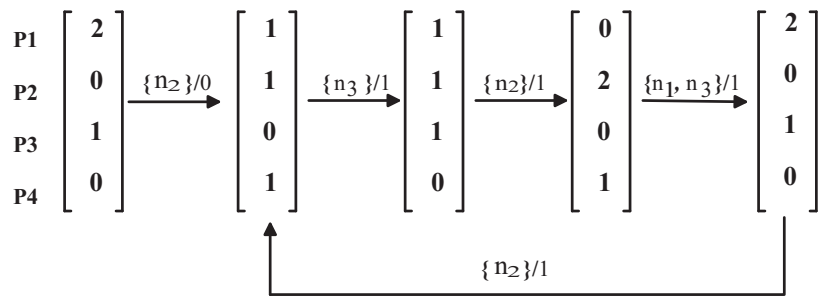


Figure 4.9 – Graphe d'évolution du marquage du GETDM de la figure 4.8.

Notation 3.

- $\{kn_i\}/t$: après t unités de temps, la transition n_i est franchie k fois.

On note que lorsque les circuits ϱ_1 et ϱ_2 sont considérés séparément, le temps de cycle de ϱ_1 est égal à 2 unités de temps, de même pour le temps de cycle de ϱ_2 . Contrairement au GET ordinaire, le maximum des temps de cycle ne donne pas le temps de cycle du GETDM (égal à 4 unités de temps).

Avant de présenter notre contribution dans ce domaine, présentons les différents travaux que nous avons pu trouver sur l'analyse de performances des GETDM.

1. Méthode proposés par D.T. Chao

D.T. Chao [Chao et al., 1993] propose une méthode analytique pour calculer le temps de cycle des GETDM avec des conditions restrictives sur le marquage initial. Cette méthode s'applique lorsque chaque circuit élémentaire du graphe a une unique place p marquée d'un marquage valué $\mathcal{N}_p \geq 1$ (cf. équation (2.9)). Ce temps de cycle se calcule de la façon suivante :

$$TC_m = \max_{\varrho \text{ circuit élémentaire}} \frac{\mathcal{T}(\varrho)}{\mathcal{N}(\varrho)},$$

où $\mathcal{N}(\varrho)$ est le marquage valué du circuit ϱ défini par la relation (2.10). $\mathcal{T}(\varrho)$ correspond à la somme des temporisations dans le circuit ϱ . Cette méthode permet le calcul du temps de cycle d'un GETDM dans le cas multi-serveur (cf. définition 2.8). Dans le cas mono-serveur, les boucles de ré-entrance implicites contiennent un seul jeton et doivent être considérées comme des circuits. La condition $\mathcal{N}_p \geq 1$ est très rarement vérifiée pour ces places implicites. Par conséquent, cette méthode est peu applicable dans le cas des GETDM mono-serveurs.

2. Méthode proposée par Alix Munier

Cette technique est une extension des travaux de P. Chrétienne [Chretienne, 1983] et C. Ramchandani [Ramchandani, 1974], elle a pour objectif le calcul du temps de cycle moyen d'un graphe d'événements avec des poids sur les arcs. Dans le cas mono-serveur, A. Munier montre que le calcul du temps de cycle d'un graphe avec des poids sur les arcs peut se ramener à celui d'un graphe d'événements ordinaire. Pour cela, une méthode de linéarisation a été proposée. Cette méthode nécessite l'expansion du réseau initial. Cette expansion se fait en deux étapes : la première est la décomposition du graphe initial en un ensemble de composantes fortement connexes ; la seconde est la duplication des noeuds du graphe (places et transitions). Le lecteur intéressé par cette méthode peut trouver plus de détails dans [Munier, 1993].

Dans le cas où les modèles sont multi-serveurs, une extension de cette méthode a été proposée par Nakamura et Silva [Nakamura and Silva, 1999].

3. Approche de fluidification proposée par Guy Cohen

Cette méthode prend en compte le problème de pondération sur les arcs en transformant le modèle en un GETC. Elle utilise un dioïde d'opérateurs pour la modélisation du GETCM. Son principe consiste à linéariser le modèle mathématique qui reflète l'évolution dynamique du modèle. Le modèle obtenu est $(\min, +)$ linéaire et permet d'utiliser certains éléments de la théorie spectrale pour l'analyse de performances des GET ordinaires. Cette méthode présente des avantages : elle préserve la structure et la taille du graphe initial. Il a été montré que le comportement d'un GETDM

coïncide avec sa version continue si :

$$\forall q \in T, \forall p \in \bullet q, \frac{m_p}{M_{qp}} \in \theta_q \mathbb{N}$$

où m_p est le marquage de la place p , M_{qp} est le poids de l'arc sortant de la place p , θ_q est la composante de T-invariant associée à la transition n_q . Ainsi, sous cette condition, il est possible de linéariser un GETDM en considérant sa version continue. Pour une présentation détaillée de cette méthode, nous renvoyons le lecteur à [Cohen et al., 1998c].

4. Méthode proposée par Benoît Trouillet [Trouillet, 2003]

A l'aide d'une algèbre spécifique, appelée (*min, div*), une méthode est proposée afin de transformer un GETDM en un GET ordinaire, indépendamment du marquage initial. Cette méthode considère que les GETDM sont fortement connexes, et que chaque circuit du graphe admet au moins une composante du T-invariant égale à 1.

5. D'autres méthodes existent pour évaluer les performances des GETDM. Des bornes inférieures et supérieures du temps de cycle moyen dans les cas déterministe et stochastique ont été proposées par [Compos et al., 1991] (dans [Balbo and Silva, 1998, §17]) en utilisant des techniques de programmation linéaire.

Sont proposées également des méthodes de simulation, en faisant évoluer le graphe, pour déterminer le temps de cycle moyen d'un GETDM [Sauer, 2004].

Il résulte de cette étude bibliographique qu'une possibilité, pour calculer analytiquement le temps de cycle moyen d'un GETDM, sans hypothèse restrictive sur le marquage initial, consiste à construire un GET discret ordinaire équivalent. Par contre, les méthodes décrites précédemment ont pour conséquence d'étendre le nombre de transitions du graphe. Une telle expansion peut alors rendre l'étude du GET ordinaire, obtenu après linéarisation, laborieuse dans le cas où la taille du GETDM est élevée.

Dans ce contexte, pour aborder le problème d'évaluation de performances des GETDM, nous proposons deux méthodes de linéarisation des modèles mathématiques reflétant l'évolution dynamique de ces graphes.

4.5 Méthode de linéarisation des graphes d'événements temporisés discrets avec multipliers avec des conditions sur le marquage initial

Cette méthode s'applique sur des GETDM consistants (cf. définition 2.3) et conservatifs (cf. définition 2.4). On note N le vecteur d'état du GETDM. On définit un vecteur X de même dimension que N . Chaque composante de X de type compteur satisfait une équation récurrente $(min, +)$ linéaire.

Cette méthode a pour principe d'exprimer chaque composante de N en fonction d'une composante de X . Le fait que le GETDM considéré soit consistant implique l'existence d'un T-invariant sur les transitions. Comme on a vu au chapitre 2, un T-invariant est un vecteur dont les composantes ne sont pas forcément unitaires, contrairement à un GET où toutes les composantes du T-invariant sont unitaires. En effet, chaque compteur $n_q(t)$ associé à la transition n_q du GETDM s'exprime comme suit :

$$\forall q \in \mathcal{T}, \forall t \in \mathbb{Z}, \quad n_q(t) = \theta_q x_q(t), \quad (4.5)$$

où :

- $n_q(t)$ est le compteur associé à la transition n_q du GETDM,
- θ_q est la composante ($\in \mathbb{N}^*$) du T-invariant associée à la transition n_q du GETDM.

Contrairement à la méthode proposée par Alix Munier, où à chaque transition n_q du GETDM correspond θ_q représentants dans le GET obtenu après linéarisation (ce qui provoque une expansion du graphe), le changement de variables (4.5) signifie que durant un cycle, chaque tir d'une transition x_q du GET est équivalent à θ_q tirs de la transition n_q dans le GETDM, ce qui donne l'équivalence entre le comportement du GETDM et le GET obtenu après linéarisation.

4.5.1 Linéarisation exacte des graphes d'événements temporisés discrets avec multipliers

Pour que le changement de variable (4.5) soit possible, c'est-à-dire pour que le GETDM soit linéarisable, chaque compteur $x_q(t)$ doit appartenir au dioïde \mathbb{Z}_{min} . La proposition suivante donne la condition pour que le GETDM soit linéarisable.

Proposition 4.25. *Un GETDM est linéarisable si :*

$$\forall q \in \mathcal{T}, \forall p \in \bullet q, \quad \lfloor \frac{m_p}{M_{qp}} \rfloor \in \theta_q \mathbb{N}. \quad (4.6)$$

Preuve 5. D'après l'assertion 1 (chapitre 2), chaque compteur $n_q(t)$ d'un GETDM satisfait la relation suivante :

$$n_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \lfloor M_{qp}^{-1} (m_p + M_{pq'} n_{q'}(t - \tau_p)) \rfloor.$$

En utilisant le changement de variable (4.5) et la distributivité de la multiplication par rapport à l'opérateur min , on obtient l'expression $x_q(t)$ suivante :

$$x_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \frac{1}{\theta_q} \lfloor (\frac{m_p}{M_{qp}} + \frac{M_{pq'}}{M_{qp}} n_{q'}(t - \tau_p)) \rfloor.$$

Notons $\theta = (\theta_1 \ \theta_2 \dots \theta_n)^t$ le vecteur T-invariant. Ce vecteur satisfait l'expression $C \times \theta = 0$, où C est la matrice d'incidence associée au GETDM. Il en résulte, sachant que les graphes sont consistants et conservatifs [Munier, 1993], que :

$$\forall q', q \in \mathcal{T} \quad p \in P, M_{qp} \theta_q - \theta_{q'} M_{pq'} = 0.$$

De cette relation, on déduit que :

$$\frac{\theta_q}{M_{pq'}} = \frac{\theta_{q'}}{M_{qp}},$$

par conséquent,

$$x_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \frac{1}{\theta_q} \lfloor (\frac{m_p}{M_{qp}} + \frac{\theta_q}{\theta_{q'}} n_{q'}(t - \tau_p)) \rfloor,$$

c'est-à-dire,

$$x_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \frac{1}{\theta_q} \lfloor (\frac{m_p}{M_{qp}} + \theta_q x_{q'}(t - \tau_p)) \rfloor.$$

Puisque $\theta_q x_{q'}(t - \tau_p)$ est un entier, nous obtenons :

$$x_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \left(\frac{1}{\theta_q} \lfloor \frac{m_p}{M_{qp}} \rfloor + x_{q'}(t - \tau_p) \right), \quad (4.7)$$

ce qui correspond à une équation récurrente ($min, +$) linéaire dès lors que $\frac{1}{\theta_q} \lfloor \frac{m_p}{M_{qp}} \rfloor \in \mathbb{N}$.

Aussi, si la condition (4.25) est vérifiée pour toute transition du graphe, l'équation (2.11) s'exprime comme une équation récurrente ($min, +$) linéaire.

Remarque 4.26. Définissons une classe d'équivalence au niveau des marquages initiaux induite par la relation d'équivalence suivante :

$$m' \equiv m'' \Leftrightarrow \forall q \in \mathcal{T}, \forall p \in \bullet q, \quad \lfloor \frac{m'_p}{M_{qp}} \rfloor = \lfloor \frac{m''_p}{M_{qp}} \rfloor.$$

Tous les marquages initiaux appartenant à une même classe d'équivalence engendrent un même comportement au niveau des transitions du graphe. Aussi, pour ces mêmes marquages, le GETDM admet, après linéarisation, un même modèle $(min, +)$ linéaire.

Cette linéarisation nous permet de déduire les performances du GETDM. En effet, la fréquence de franchissement associée à la transition n_q du GETDM, notée λ_{m_q} , peut se déduire du modèle $(min, +)$ linéaire obtenu, selon la relation suivante :

$$\lambda_{m_q} = \theta_q \lambda, \quad (4.8)$$

où λ est la valeur propre du modèle $(min, +)$ linéaire.

4.5.2 Linéarisation approchée par ajout/retrait de jetons

Dans le cas où la condition de linéarisation (4.25) n'est pas satisfaite, une alternative consiste à introduire la notion de la linéarisation *approchée* correspondant à un *ajout/retrait* de jetons dans le GETDM.

• **Principe :** Cette méthode a pour principe d'approcher autant que possible, en utilisant la condition de linéarisation (4.25), le comportement du GETDM initial. Il en résulte une approximation par valeur inférieure (resp., supérieure) obtenue par un retrait (resp., ajout) minimal de jetons dans les places ne satisfaisant pas la condition de linéarisation. Autrement dit, dans chaque place p pour laquelle $\lfloor \frac{m_p}{M_{qp}} \rfloor \notin \theta_q \mathbb{N}$, nous retirons autant de jetons que nécessaire du marquage initial, afin que la condition de linéarisation soit vérifiée.

On note \underline{X} (resp., \overline{X}) le vecteur d'état du GET ordinaire obtenu à l'issue de la linéarisation approchée par retrait (resp., ajout) de jetons dans le GETDM.

En reprenant le même raisonnement que celui permettant d'établir la proposition 4.25, nous obtenons :

$$\underline{x}_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \left(\frac{1}{\theta_q} \lfloor \frac{m_p - \underline{m}_p}{M_{qp}} \rfloor + \underline{x}_{q'}(t - \tau_p) \right), \quad (4.9)$$

où \underline{m}_p est le nombre minimum de jetons retirés de la place p tel que $\lfloor \frac{m_p - \underline{m}_p}{M_{qp}} \rfloor \in \theta_q \mathbb{N}$.

De même, on a :

$$\overline{x}_q(t) = \min_{p \in \bullet q, q' \in \bullet p} \left(\frac{1}{\theta_q} \lfloor \frac{m_p + \overline{m}_p}{M_{qp}} \rfloor + \overline{x}_{q'}(t - \tau_p) \right), \quad (4.10)$$

où \bar{m}_p est le nombre minimum de jetons ajoutés dans la place p tel que $\lfloor \frac{m_p + \bar{m}_p}{M_{qp}} \rfloor \in \theta_q \mathbb{N}$.

Sachant que le retrait (resp., l'ajout) de jetons dans un GETDM ralentit (resp., accélère) l'évolution du graphe, on a par conséquent :

$$\forall q, \theta_q \underline{x}_q(t) = \underline{n}_q(t) \leq n_q(t) \leq \bar{n}_q(t) = \theta_q \bar{x}_q(t).$$

Contrairement au cas d'une linéarisation exacte, la linéarisation approchée nous permet seulement d'encadrer la fréquence de franchissement de la transition n_q , soit λ_{m_q} , on a :

$$\underline{\lambda}_{m_q} \leq \lambda_{m_q} \leq \bar{\lambda}_{m_q}, \quad (4.11)$$

où $\underline{\lambda}_{m_q}$ (resp., $\bar{\lambda}_{m_q}$) correspond à la fréquence de franchissement de la transition n_q obtenu par un retrait (resp., ajout) minimal de jetons dans les places du GETDM pour lesquelles la condition de linéarisation (4.25) n'est pas satisfaite.

4.5.3 Exemple

Cet exemple est inspiré de la thèse de Benoît Trouillet [Trouillet, 2003].

Le GETDM de la figure 4.10 modélise une chaîne de production produisant deux types de produit, A et B .

Le produit A est composé de deux pièces A_1 et d'une pièce A_2 . Le premier composant est fabriqué par une machine de type p_{13} . La fabrication du deuxième composant nécessite les deux pièces de la machine p_{13} . La machine p_5 effectue l'assemblage de deux pièces A_1 et d'une pièce A_2 .

Le deuxième produit B est fabriqué par la machine p_{12} . La première opération est effectuée sur un lot de trois pièces sur la machine p_6 . La pièce produite sera ensuite traitée successivement sur les machines p_8 et p_{10} . Parallèlement, un lot de deux pièces est traité sur la machine p_7 , puis ensuite sur p_9 et p_{11} . Enfin, un lot de cinq pièces est constitué pour être traité sur la machine p_{12} .

Afin d'obtenir un GETDM fortement connexe caractérisant le comportement cyclique du système, le modèle est rebouclé (arcs joignant la transition n_4 aux places p_1 et p_3 , et la place p_{12} à la transition n_5).

Equations récurrentes En considérant le GETDM représenté par la figure 4.10, nous déterminons les équation récurrentes régissant l'évolution dynamique de ce graphe, soit le système d'équations suivant :

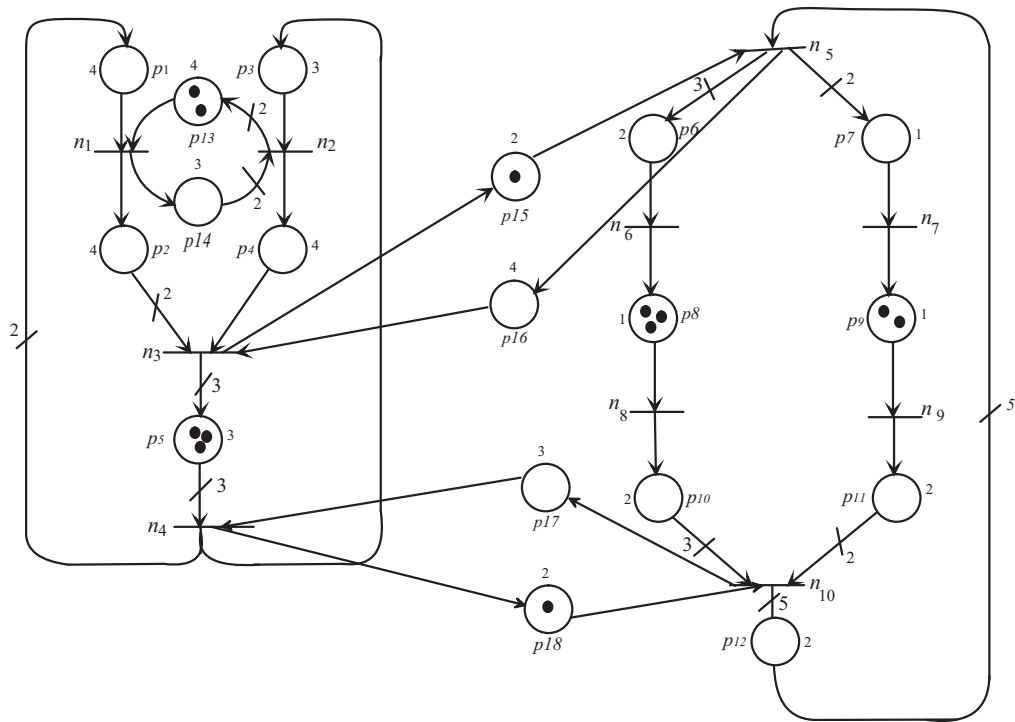


Figure 4.10 – Un GETDM .

$$\left\{ \begin{array}{l} n_1(t) = \min(2 + 2n_2(t - 4), 2n_4(t - 4)), \\ n_2(t) = \min(\lfloor \frac{n_1(t-3)}{2} \rfloor, n_4(t - 3)), \\ n_3(t) = \min(\lfloor \frac{n_1(t-4)}{2} \rfloor, n_2(t - 4), n_5(t - 4)), \\ n_4(t) = \min(\lfloor \frac{3+3n_3(t-3)}{3} \rfloor, n_{10}(t - 3)) \\ n_5(t) = \min(1 + n_3(t - 2), n_{10}(t - 2)), \\ n_6(t) = 3n_5(t - 2), \\ n_7(t) = 2n_5(t - 1), \\ n_8(t) = 3 + n_6(t - 1), \\ n_9(t) = 2 + n_7(t - 1), \\ n_{10}(t) = \min(\lfloor \frac{n_8(t-2)}{3} \rfloor, \lfloor \frac{n_9(t-2)}{2} \rfloor). \end{array} \right.$$

Linéarisation exacte Comme nous l'avons constaté précédemment, le système ainsi obtenu n'est pas $(\min, +)$ linéaire en raison de la présence de parties entières rendant les équations non linéaires. Pour linéariser ces équations, on commence par rechercher l'invariant total de transitions. Cet invariant correspond à $\theta^t = (2 \ 1 \ 1 \ 1 \ 1 \ 3 \ 2 \ 3 \ 2 \ 1)$.

On vérifie que chaque marquage d'une place satisfait la condition (4.25), ce qui signifie que le GETDM est linéarisable. En utilisant le changement de variable (4.5), on obtient le modèle $(\min, +)$ linéaire suivant, issu de l'équation (4.7) :

$$\left\{ \begin{array}{l} x_1(t) = \min(1 + x_2(t - 4), x_4(t - 4)), \\ x_2(t) = \min(x_1(t - 3), x_4(t - 3)), \\ x_3(t) = \min(x_1(t - 4), x_2(t - 4), x_5(t - 4)), \\ x_4(t) = \min(1 + x_3(t - 3), x_{10}(t - 3)), \\ x_5(t) = \min(1 + x_3(t - 2), x_{10}(t - 2)), \\ x_6(t) = x_5(t - 2), \\ x_7(t) = x_5(t - 1), \\ x_8(t) = 1 + x_6(t - 1), \\ x_9(t) = 1 + x_7(t - 1), \\ x_{10}(t) = \min(x_8(t - 2), x_9(t - 2)). \end{array} \right.$$

Ces équations modélisent le comportement du GET représenté par la figure 4.11.

Pour évaluer les performances du GETDM représenté par la figure 4.10, on utilise le modèle $(\min, +)$ linéaire associé pour déterminer les fréquences de franchissement de chacune de ses transitions et son

temps de cycle.

En utilisant le théorème 4.17, on trouve un taux de production du GET de la figure 4.11, obtenu après linéarisation, égal à $\frac{1}{11}$. De l'équation (4.8), on déduit la fréquence de franchissement de chaque transition du GETDM, soient

$$\lambda_{m_1} = \frac{2}{11}, \lambda_{m_2} = \lambda_{m_3} = \lambda_{m_4} = \lambda_{m_5} = \frac{1}{11}, \lambda_{m_6} = \frac{3}{11}, \lambda_{m_7} = \frac{2}{11}, \lambda_{m_8} = \frac{3}{11}, \lambda_{m_9} = \frac{2}{11}, \lambda_{m_{10}} = \frac{1}{11}.$$

Par conséquent, l'équation (4.4) permet de déduire que le temps de cycle TC_m du GETDM de la figure 4.10 est égal à 11 unités de temps.

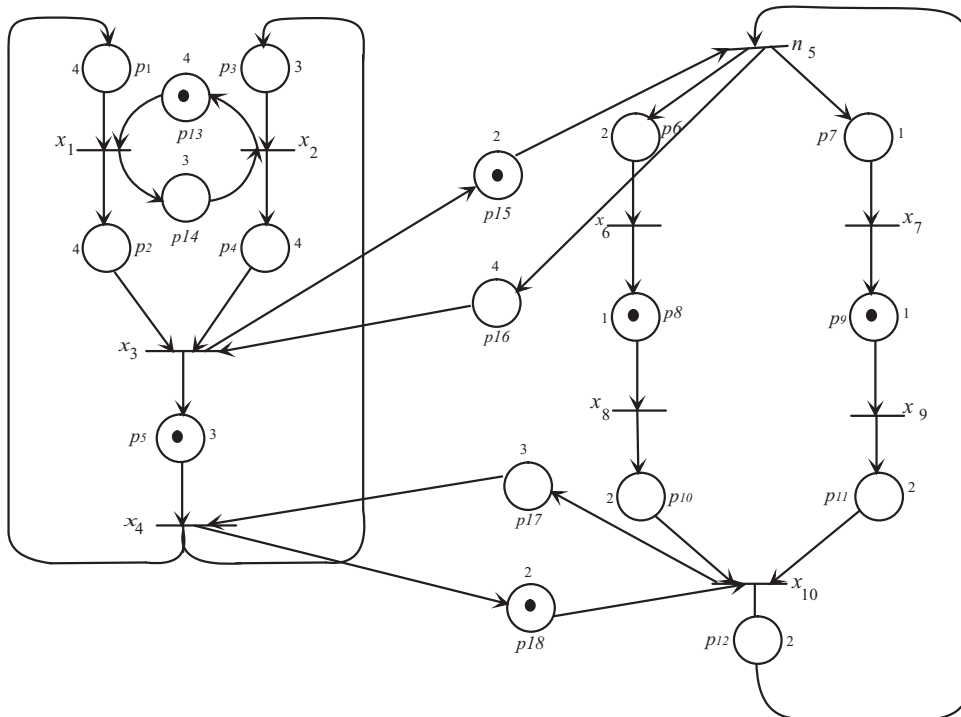


Figure 4.11 – GETDM linéarisé.

Linéarisation approchée par ajout/retrait de jetons

Pour le marquage initial suivant : $M(0) = (1 \ 0 \ 0 \ 0 \ 3 \ 0 \ 0 \ 3 \ 2 \ 0 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1)$, on

note que la place p_1 , marqué avec un jeton, ne satisfait pas la condition de linéarisation (4.25), ce qui motive l'utilisation de la méthode de *linéarisation approchée par ajout/retrait de jetons* dans le graphe. Le retrait d'un jeton dans la place p_1 permet de vérifier la condition (4.25), de même que l'ajout d'un jeton dans cette place.

L'utilisation des équations (4.9) et (4.10) permet d'obtenir les modèles linéaires approchés suivants :

$$\left\{ \begin{array}{l} \underline{x}_1(t) = \min(1 + \underline{x}_2(t-4), \underline{x}_4(t-4)), \\ \underline{x}_2(t) = \min(\underline{x}_1(t-3), \underline{x}_4(t-3)), \\ \underline{x}_3(t) = \min(\underline{x}_1(t-4), \underline{x}_2(t-4), \underline{x}_5(t-4)), \\ \underline{x}_4(t) = \min(1 + \underline{x}_3(t-3), \underline{x}_{10}(t-3)), \\ \underline{x}_5(t) = \min(1 + \underline{x}_3(t-2), \underline{x}_{10}(t-2)), \\ \underline{x}_6(t) = \underline{x}_5(t-2), \\ \underline{x}_7(t) = \underline{x}_5(t-1), \\ \underline{x}_8(t) = 1 + \underline{x}_6(t-1), \\ \underline{x}_9(t) = 1 + \underline{x}_7(t-1), \\ \underline{x}_{10}(t) = \min(\underline{x}_8(t-2), \underline{x}_9(t-2)). \end{array} \right.$$

et

$$\left\{ \begin{array}{l} \bar{x}_1(t) = \min(1 + \bar{x}_2(t-4), 1 + \bar{x}_4(t-4)), \\ \bar{x}_2(t) = \min(\bar{x}_1(t-3), \bar{x}_4(t-3)), \\ \bar{x}_3(t) = \min(\bar{x}_1(t-4), \bar{x}_2(t-4), \bar{x}_5(t-4)), \\ \bar{x}_4(t) = \min(1 + \bar{x}_3(t-3), \bar{x}_{10}(t-3)), \\ \bar{x}_5(t) = \min(1 + \bar{x}_3(t-2), \bar{x}_{10}(t-2)), \\ \bar{x}_6(t) = \bar{x}_5(t-2), \\ \bar{x}_7(t) = \bar{x}_5(t-1), \\ \bar{x}_8(t) = 1 + \bar{x}_6(t-1), \\ \bar{x}_9(t) = 1 + \bar{x}_7(t-1), \\ \bar{x}_{10}(t) = \min(\bar{x}_8(t-2), \bar{x}_9(t-2)). \end{array} \right.$$

Dans le cas où un jeton est retiré de la place p_1 dans le GETDM, on obtient un GET avec une valeur propre λ égale à $\frac{1}{11}$. L'équation (4.8) permet de déduire la fréquence de franchissement de chaque transition du GETDM : $\lambda_{m_1} = \frac{2}{11}$, $\lambda_{m_2} = \lambda_{m_3} = \lambda_{m_4} = \lambda_{m_5} = \frac{1}{11}$, $\lambda_{m_6} = \frac{3}{11}$, $\lambda_{m_7} = \frac{2}{11}$, $\lambda_{m_8} = \frac{3}{11}$,

$\lambda_{m_9} = \frac{2}{11}$, $\lambda_{m_{10}} = \frac{1}{11}$. Par conséquent, on déduit de l'équation (4.4) que le temps de cycle TC_m est égal à 11 unités de temps.

Dans le cas où on a ajouté un jeton dans la place p_1 dans le GETDM, le taux de production du GET obtenu λ est égale à $\frac{1}{10}$. De l'équation (4.8), on déduit la fréquence de franchissement de chaque transition du GETDM : $\bar{\lambda}_{m_1} = \frac{2}{10}$, $\bar{\lambda}_{m_2} = \bar{\lambda}_{m_3} = \bar{\lambda}_{m_4} = \bar{\lambda}_{m_5} = \frac{1}{10}$, $\bar{\lambda}_{m_6} = \frac{3}{10}$, $\bar{\lambda}_{m_7} = \frac{2}{10}$, $\bar{\lambda}_{m_8} = \frac{3}{10}$, $\bar{\lambda}_{m_9} = \frac{2}{10}$, $\bar{\lambda}_{m_{10}} = \frac{1}{10}$. Par conséquent, l'équation (4.4) permet de déduire que le temps de cycle \overline{TC}_m est égal à 10 unités de temps.

Finalement, on obtient pour $M(0) = (1 \ 0 \ 0 \ 0 \ 3 \ 0 \ 0 \ 3 \ 2 \ 0 \ 0 \ 0 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1)$ les résultat suivant :

$$10 \leq TC_m \leq 11,$$

$$\frac{2}{11} \leq \lambda_{m_1} \leq \frac{2}{10}, \quad \frac{1}{11} \leq \lambda_{m_2} \leq \frac{1}{10}, \quad \frac{1}{11} \leq \lambda_{m_3} \leq \frac{1}{10}, \quad \frac{1}{11} \leq \lambda_{m_4} \leq \frac{1}{10}, \quad \frac{1}{11} \leq \lambda_{m_5} \leq \frac{1}{10},$$

$$\frac{3}{11} \leq \lambda_{m_6} \leq \frac{3}{10}, \quad \frac{2}{11} \leq \lambda_{m_7} \leq \frac{2}{10}, \quad \frac{3}{11} \leq \lambda_{m_8} \leq \frac{3}{10}, \quad \frac{2}{11} \leq \lambda_{m_9} \leq \frac{2}{10}, \quad \frac{1}{11} \leq \lambda_{m_{10}} \leq \frac{1}{10}.$$

4.6 Méthode de linéarisation des graphes d'événements temporisés discrets avec multipliers avec une condition sur le T-invariant

Cette méthode concerne également les graphes consistants et conservatifs dont au moins une composante du T-invariant est égale à 1. Une telle composante est associée à une transition dite de *normalisation*. Cette méthode est basée sur une modélisation opératorielle dans le dioïde $\mathcal{D}_{\min}[\delta]$, définie dans le chapitre 2, pour coder les compteurs associés aux transitions du graphe.

Comme mentionné précédemment, chaque compteur d'un GETDM peut être codé par la composition des opérateurs : μ , γ , δ (cf. équation (2.12)).

En plus des règles de commutation entre ces opérateurs, cités précédemment dans le chapitre 2, la proposition suivante fournit d'autres règles utiles pour la méthode de linéarisation que nous allons proposer.

Proposition 4.27. Soient $a, b \in \mathbb{N}^*$, nous avons :

1. $\mu_{a-1}\mu_b = \mu_{(a-1)b}$.
2. Soit $N(\delta)$ tel que, $\forall t \in \mathbb{Z}$, $n(t)$ est un multiple de a , alors $\mu_{a-1}\gamma^b N(\delta) = \gamma^{[a^{-1}b]}\mu_{a-1}N(\delta)$.
3. $\gamma^b\mu_a = \mu_a\gamma^{a^{-1}b}$ est équivalent à $\mu_a\gamma^b = \gamma^{ab}\mu_a$.

Preuve 6.

1. $\mu_{a^{-1}}\mu_b N(\delta)$ correspond à $\lfloor a^{-1} \lfloor b n(t) \rfloor \rfloor = \lfloor a^{-1} b n(t) \rfloor$, ce qui équivaut à $\mu_{(a^{-1}b)} N(\delta)$.
2. $\mu_{a^{-1}}\gamma^b N(\delta)$ correspond à $\lfloor a^{-1}(b + n(t)) \rfloor = \lfloor a^{-1}b \rfloor + a^{-1}n(t)$ sachant que $n(t) \in \mathbb{Z} \cup \{\pm\infty\}$ est un multiple de a , ce qui est équivalent à $\gamma^{\lfloor a^{-1}b \rfloor} \mu_{a^{-1}} N(\delta)$.
3. $\gamma^b \mu_a N(\delta)$ correspond à $b + \lfloor a n(t) \rfloor = \lfloor b + a n(t) \rfloor = \lfloor a(a^{-1}b + n(t)) \rfloor$, ce qui est équivalent à $\mu_a \gamma^{a^{-1}b} N(\delta)$.

Pour illustrer le principe de linéarisation de cette méthode, considérons l'exemple suivant.

Exemple 4.28. Le GETDM de la figure 4.12 admet la représentation opératorielle suivante :

$$\begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} = \begin{pmatrix} \varepsilon & \mu_{1/3}\gamma^6\delta^1\mu_2 & \varepsilon \\ \mu_{1/2}\delta^2\mu_3 & \varepsilon & \gamma^3\delta^3\mu_3 \\ \varepsilon & \mu_{1/3}\delta^1 & \varepsilon \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} \oplus \begin{pmatrix} E \\ E \\ E \end{pmatrix},$$

où E correspond à une entrée impulsionnelle ajoutée à chaque transition du graphe dans le but d'avoir des conditions initiales telles que l'évolution des jetons ne puisse se faire qu'à l'instant $t \geq 0$ (sachant que les jetons sont supposés être présents depuis l'instant $-\infty$).

4.6.1 Principe de linéarisation

Un GETDM consistant possède un T-invariant minimal, dans lequel toutes ses composantes appartiennent à \mathbb{N}^* . Le principe de linéarisation consiste à normaliser le T-invariant du GETDM de façon que les transitions aient le même nombre de franchissements par cycle.

Soit $\theta = (\theta_1 \theta_2 \dots \theta_n)^t$ le vecteur T-invariant du GETDM avec $n = \text{card}(\mathcal{T})$.

On note $\text{ppcm}(\theta_i)$ le plus petit commun multiple des entiers θ_i . Par conséquent, pour tout $i \in \{1, \dots, n\}$, il existe un entier α_i tel que $\theta_i \alpha_i = \text{ppcm}(\theta_i)$. On introduit le vecteur d'état $X = (x_1 \ x_2 \dots x_n)^t$ défini par $x_i = \alpha_i n_i$ pour tout $i \in \{1, \dots, n\}$.

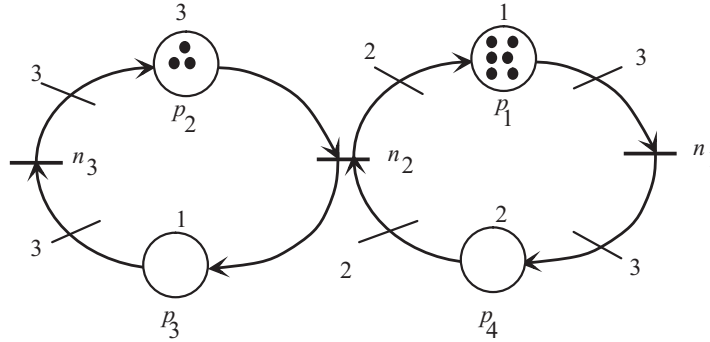


Figure 4.12 – GETDM à linéariser.

Définition 4.29. Si θ représente le T-invariant associé aux transitions, on appelle vecteur de franchissement normalisé, ou linéarisé, le vecteur $X = (x_i)_{1 \leq i \leq n}$ défini à partir du vecteur de franchissement $N = (n_i)_{1 \leq i \leq n}$, selon la relation suivante :

$$x_i(t) = \alpha_i \times n_i(t), \quad (4.12)$$

$$\text{où } \alpha_i = \frac{\text{ppcm}(\theta_i)}{\theta_i}.$$

Exemple 4.30. Le T-invariant du GETDM représenté par la figure 4.12 est : $(2 \ 3 \ 1)^t$. Cela signifie que, lors d'un cycle, c'est-à-dire après franchissement de $2n_1, 3n_2, 1n_3$, le marquage du graphe correspond au marquage initial. Dans ce cas, $\text{ppcm} = 6$, $\alpha_1 = 3$, $\alpha_2 = 2$ et $\alpha_3 = 6$. Nous obtenons ainsi le vecteur de franchissement normalisé suivant :

$$x_1(t) = 3n_1(t),$$

$$x_2(t) = 2n_2(t),$$

$$x_3(t) = 6n_3(t).$$

Rappelons que l'évolution d'un compteur d'un GETDM est régit par une équation non linéaire dans l'algèbre $(\min, +)$. Dans le but d'obtenir une représentation $(\min, +)$ linéaire, la proposition suivante donne l'expression du vecteur de franchissement normalisé.

Proposition 4.31. *Considérant le changement de variable suivant : $x_i(t) = \alpha_i n_i(t)$; $\forall i = 1, \dots, n$ tel que $\alpha_i \theta_i = \text{ppcm}(\theta_i)$. Les composantes x_i peuvent s'exprimer dans le dioïde $\mathcal{D}_{\min}[\delta]$ comme suit :*

$$X_q(\delta) = \bigoplus_{p \in \bullet q, q' \in \bullet p} \mu_{\alpha_q} \mu_{\alpha_q}^{-1} \gamma^{\alpha_q M_{qp}^{-1} m_p} \delta^{\tau_p} X_{q'}(\delta), \quad (4.13)$$

ce qui est équivalent à,

$$x_q(t) = \min_{p \in \bullet q, q' \in \bullet p} (\lfloor \alpha_q \lfloor \frac{\alpha_q M_{qp}^{-1} m_p + x_{q'}(t - \tau_p)}{\alpha_q} \rfloor \rfloor).$$

Preuve 7. On a $X_q(\delta) = \mu_{\alpha_q} N_q(\delta)$ avec $\alpha_q \in \mathbb{N}^*$, aussi on déduit de l'équation (2.12) que :

$$X_q(\delta) = \bigoplus_{p \in \bullet q, q' \in \bullet p} \mu_{\alpha_q} \mu_{M_{qp}^{-1}} \gamma^{m_p} \delta^{\tau_p} \mu_{M_{pq'}} N_{q'}(\delta).$$

Le point 3 de la proposition 4.27 permet d'écrire :

$$X_q(\delta) = \bigoplus_{p \in \bullet q, q' \in \bullet p} \mu_{\alpha_q} \mu_{M_{qp}^{-1}} \mu_{M_{pq'}} \gamma^{M_{pq'}^{-1} m_p} \delta^{\tau_p} N_{q'}(\delta).$$

Du point 1 de la proposition 4.27, il en résulte que :

$$X_q(\delta) = \bigoplus_{p \in \bullet q, q' \in \bullet p} \mu_{\alpha_q} \mu_{(M_{qp}^{-1} M_{pq'})} \gamma^{M_{pq'}^{-1} m_p} \delta^{\tau_p} N_{q'}(\delta).$$

De la définition de θ (c'est-à-dire $\theta \neq 0$ et $C \times \theta = 0$ sachant que C correspond à la matrice d'incidence), on déduit que $M_{pq'} \theta_{q'} = M_{qp} \theta_q$. Cependant, sachant que $\theta_q = \alpha_q^{-1} \text{ppcm}(\theta_i)_{1 \leq i \leq n}$ et $\theta_{q'} = \alpha_{q'}^{-1} \text{ppcm}(\theta_i)_{1 \leq i \leq n}$, on déduit que : $M_{qp}^{-1} M_{pq'} = \alpha_q^{-1} \alpha_{q'}$ et $M_{pq'}^{-1} = \alpha_{q'}^{-1} \alpha_q M_{qp}^{-1}$. Par conséquent, les points 1 et 3 de la proposition 4.27 permettent d'écrire :

$$\begin{aligned} X_q(\delta) &= \bigoplus_{p \in \bullet q, q' \in \bullet p} \mu_{\alpha_q} \mu_{(\alpha_q^{-1} \alpha_{q'})} \gamma^{\alpha_{q'}^{-1} \alpha_q M_{qp}^{-1} m_p} \delta^{\tau_p} N_{q'}(\delta), \\ &= \bigoplus_{p \in \bullet q, q' \in \bullet p} \mu_{\alpha_q} \mu_{\alpha_q^{-1}} \gamma^{\alpha_q M_{qp}^{-1} m_p} \delta^{\tau_p} \mu_{\alpha_{q'}} N_{q'}(\delta), \end{aligned}$$

ce qui donne l'expression de $X(\delta)$ puisque $X_{q'}(\delta) = \mu_{\alpha_{q'}} N_{q'}(\delta)$.

On note que la commutation de l'opérateur $\mu_{\alpha_q^{-1}}$ avec $\gamma^{\alpha_q M_{qp}^{-1} m_p}$ pour linéariser l'équation (4.13) n'est pas toujours possible (cf. point 2 de la proposition 4.27).

Pour pouvoir linéariser les équations aux compteurs associées aux transitions d'un GETDM, on exprime chaque compteur du graphe en fonction d'un compteur associé à une transition ayant sa composante dans le T-invariant égale à 1. Cette transition, dite *de normalisation*, va permettre, comme indiqué par la suite, de linéariser le modèle.

Définition 4.32. [Trouillet et al., 2002] Une transition de *normalisation* est une transition pour laquelle le coefficient qui lui est associé dans le T-invariant est égale à 1.

Pour l'exemple représenté par la figure 4.12, nous constatons que n_3 est une transition de normalisation.

Proposition 4.33. Soit $x_{q'}$ une transition de normalisation, on a : $\forall a \in \mathbb{N}, \forall q = 1, \dots, n,$

$$\mu_{\alpha_q} \mu_{\alpha_q^{-1}} \gamma^a X_{q'}(\delta) = \gamma^{\alpha_q \lfloor \alpha_q^{-1} a \rfloor} X_{q'}(\delta).$$

Preuve 8. Par définition de $x_{q'}$, on a $x_{q'}(t) = \text{ppcm}(\theta_i) \times n_{q'}(t)$. Sachant que $x_{q'}(t)$ est un multiple de α_q ($\alpha_q \in \mathbb{N}^*$) pour tout $q \in \{1, \dots, n\}$, on déduit, d'après le point 2 de la proposition 4.27, que : $\mu_{\alpha_q^{-1}} \gamma^a X_{q'}(\delta) = \gamma^{\lfloor \alpha_q^{-1} a \rfloor} \mu_{\alpha_q^{-1}} X_{q'}(\delta)$.

Aussi, en utilisant le point 3 de la proposition 4.27, on déduit que :

$\mu_{\alpha_q} \mu_{\alpha_q^{-1}} \gamma^a X_{q'}(\delta) = \gamma^{\alpha_q \lfloor \alpha_q^{-1} a \rfloor} \mu_{\alpha_q} \mu_{\alpha_q^{-1}} X_{q'}(\delta) = \gamma^{\alpha_q \lfloor \alpha_q^{-1} a \rfloor} X_{q'}(\delta)$, sachant que $\mu_{\alpha_q} \mu_{\alpha_q^{-1}} X_{q'}(\delta)$ correspond à $\lfloor \alpha_q \times \lfloor \alpha_q^{-1} x_{q'}(t) \rfloor \rfloor = x_{q'}(t)$ puisque $x_{q'}(t)$ est un multiple de α_q .

Dans la suite, on donne le modèle d'état obtenu après le changement de variables donné par la proposition 4.31. Considérons le vecteur X composé des compteurs décrits par la proposition 4.31. Soit $X^t = (X_{\bar{N}}^t \ X_N^t)$ où $X_{\bar{N}}$ regroupe les compteurs associés aux transitions de normalisation, et X_N regroupe les autres compteurs. Les compteurs E correspondent aux entrées impulsionnelles, elles ont aucune influence sur l'évolution du graphe, puisque $\mu_{\alpha_q} \mu_{\alpha_q^{-1}} E(\delta) = E(\delta)$. Finalement, on obtient :

$$\begin{pmatrix} X_{\bar{N}} \\ X_N \end{pmatrix} = \begin{pmatrix} F & G \\ H & L \end{pmatrix} \begin{pmatrix} X_{\bar{N}} \\ X_N \end{pmatrix} \oplus \begin{pmatrix} E \\ E \end{pmatrix}. \quad (4.14)$$

Comme il a été vu au chapitre 1, ce type d'équations admet la plus petite solution suivante :

$$\begin{cases} X_{\bar{N}} = F^* G X_N \oplus F^* E, \\ X_N = (HF^* G \oplus L) X_N \oplus HF^* E \oplus E, \end{cases} \quad (4.15)$$

avec $F^* = e \oplus F \oplus F^2 \oplus \dots$.

Proposition 4.34. Un GETDM consistant et conservatif est linéarisable sans augmentation du nombre de transitions, s'il contient au moins une transition de normalisation.

Preuve 9. De l'équation 4.13, on peut écrire $X_q(\delta) = \mu_{\alpha_q} \mu_{\alpha_q^{-1}} \gamma^{\alpha_q M_{q,p}^{-1} m_p} \delta^{\tau_p} X_{q'}(\delta) \oplus E(\delta)$. Supposons que la transition $X_{q'}$ soit une transition de normalisation. De par la proposition 4.33, on obtient :

$X_q(\delta) = \gamma^{\alpha_q \lfloor M_{q,p}^{-1} m_p \rfloor} \delta^{\tau_p} X_{q'}(\delta) \oplus E(\delta)$, c'est-à-dire une relation linéaire liant la transitions $X_{q'}$ et X_q . En général, cette technique de linéarisation peut être appliquée à toutes les transitions du graphe,

puisque celui-ci est fortement connexe et donc toutes les transitions peuvent être liées à une transition de normalisation du graphe (cf. équation 4.15).

Cette linéarisation nous permet de déduire les performances du GETDM. En effet, la fréquence de franchissement associée à chaque transition du GETDM, λ_{m_q} , peut être calculée à partir du modèle $(min, +)$ linéaire obtenu via la relation suivante :

$$\lambda_{m_q} = \frac{\lambda}{\alpha_q}, \quad (4.16)$$

où λ est la valeur propre du modèle $(min, +)$ linéaire.

4.6.2 Exemple

Le GETDM représenté sur la figure 4.12 admet, comme vu précédemment, le T-invariant $\theta^t = (231)$ où la composante θ_3 est associée à une transition de normalisation.

La proposition 4.31 permet de définir un vecteur d'état normalisé $X = (X_1 \ X_2 \ X_3)^t$ sachant que $X_i(\delta) = \mu_{\alpha_i} N_i(\delta)$, $i = 1, 2, 3$ avec $\alpha_1 = 3, \alpha_2 = 2, \alpha_3 = 6 (= ppcm(\theta_i))$. On obtient les matrices suivantes F, G, H, L avec $X_{\bar{N}} = (X_1 \ X_2)^t$, $X_N = X_3$:

$$F = \begin{pmatrix} \varepsilon & \mu_3 \mu_{1/3} \gamma^6 \delta^1 \\ \mu_2 \mu_{1/2} \delta^2 & \varepsilon \end{pmatrix}, \quad G = \begin{pmatrix} \varepsilon \\ \mu_2 \mu_{1/2} \gamma^6 \delta^3 \end{pmatrix}, \quad H = \left(\varepsilon \ \mu_6 \mu_{1/6} \delta^1 \right), \quad L = \varepsilon.$$

Le calcul de $F^* G X_N$ se base sur la propriété de la transition de normalisation X_N .

$$\begin{aligned} F^* G X_N &= (e \oplus F \oplus F^2 \oplus F^3 \oplus \dots) G X_N \\ &= (G X_N \oplus F G X_N \oplus \underbrace{F \otimes F G X_N}_{F^2 G X_N} \oplus \underbrace{F \otimes F^2 G X_N}_{F^3 G X_N} \oplus \dots). \end{aligned}$$

En suivant le même raisonnement pour le calcul de $F^* E$, on trouve :

$$\begin{aligned} F^* G X_N &= \begin{pmatrix} \gamma^{12} \delta^4 (\gamma^6 \delta^3)^* \\ \gamma^6 \delta^3 (\gamma^6 \delta^3)^* \end{pmatrix} X_N \text{ de par la proposition 4.33,} \\ F^* E &= \begin{pmatrix} (\gamma^6 \delta^3)^* \\ \delta^2 (\gamma^6 \delta^3)^* \end{pmatrix} \text{ (on a } \forall a, b \in \mathbb{N}, \mu_{\alpha_q} \mu_{\alpha_q^{-1}} \gamma^a \delta^b E(\delta) = \gamma^{\alpha_q \lfloor \alpha_q^{-1} a \rfloor} \delta^b E(\delta)). \end{aligned}$$

L'utilisation du système d'équation (4.15) permet d'obtenir :

$$\begin{cases} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \gamma^{12}\delta^4(\gamma^6\delta^3)^* \\ \gamma^6\delta^3(\gamma^6\delta^3)^* \end{pmatrix} X_3 \oplus \begin{pmatrix} (\gamma^6\delta^3)^* \\ \delta^2(\gamma^6\delta^3)^* \end{pmatrix} E, \\ X_3 = \gamma^6\delta^4(\gamma^6\delta^3)^* X_3 \oplus \delta^3(\gamma^6\delta^3)^* E, \end{cases}$$

laquelle est la plus petite solution du système d'équations implicite suivant :

$$\begin{cases} X_1 = \gamma^6\delta^3 X_1 \oplus \gamma^{12}\delta^4 X_3 \oplus E, \\ X_2 = \gamma^6\delta^3 X_2 \oplus \gamma^6\delta^3 X_3 \oplus \delta^2 E, \\ X_3 = \gamma^6\delta^4 X_3 \oplus \delta^3 E. \end{cases} \quad (4.17)$$

Ces équations admettent une représentation $(min, +)$ linéaire dans le dioïde \mathbb{Z}_{min} :

$$\begin{cases} x_1(t) = 6x_1(t-3) \oplus 12x_3(t-4) \oplus e(t), \\ x_2(t) = 6x_2(t-3) \oplus 6x_3(t-3) \oplus e(t-2), \\ x_3(t) = 6x_3(t-4) \oplus e(t-3), \end{cases}$$

lesquelles peut se représenter par le GET de la figure 4.13.

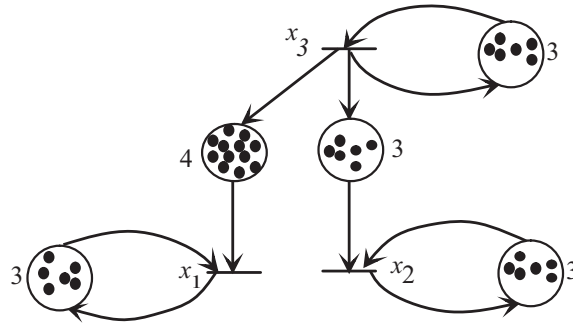


Figure 4.13 – GET linéarisé.

Notons qu'il n'y a pas unicité vis-à-vis du système implicite, le choix correspondant à (4.17) vise la simplicité au sens où a^*b est la plus petite solution de l'équation implicite $x = ax \oplus b$.

A titre d'exemple, une autre représentation extraite de (4.17) est telle que :

$$\begin{cases} X_1 = \gamma^6\delta^1 X_2 \oplus E, & (i) \\ X_2 = \delta^2 X_1 \oplus \gamma^6\delta^3 X_3 \oplus E, & (ii) \\ X_3 = \delta^1 X_2 \oplus E. & (iii) \end{cases} \quad (4.18)$$

En effet, en remplaçant, dans l'égalité (4.18.ii), la variable X_1 par son expression (4.18.i), on obtient :

$$X_2 = \gamma^6 \delta^3 X_2 \oplus \gamma^6 \delta^3 X_3 \oplus \delta^2 E, \quad \text{soit l'expression de } X_2 \text{ dans (4.17),}$$

en supprimant la partie implicite en X_2 par application du théorème 1.25, on obtient :

$$X_2 = (\gamma^6 \delta^3)^* \gamma^6 \delta^3 X_3 \oplus (\gamma^6 \delta^3)^* \delta^2 E,$$

en remplaçant, dans l'égalité (4.18.iii), la variable X_2 , on obtient :

$$X_3 = (\gamma^6 \delta^4) X_3 \oplus \delta^3 E, \quad \text{soit l'expression de } X_3 \text{ dans (4.17).}$$

En remplaçant, dans l'égalité (4.18.i), la variable X_2 par son expression (4.18.ii), on obtient directement l'expression de X_1 équivalente à celle donnée par (4.17).

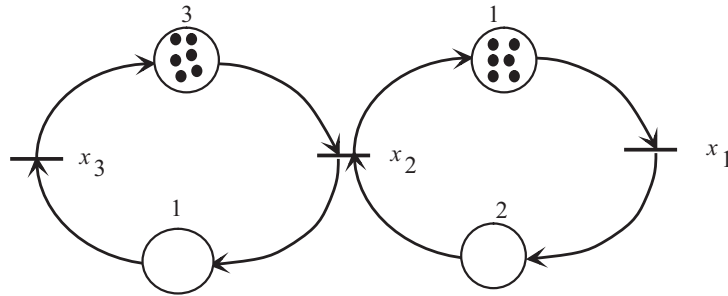


Figure 4.14 – GET linéarisé équivalent.

Il reste à établir le lien entre le GETDM et le GET résultant de la linéarisation. Ce lien est donné par la relation (4.12) qui lie le vecteur d'état N au vecteur d'état normalisé X . Pour cet exemple, nous avons $x_1(t) = 3n_1(t)$, autrement dit, à un tir de la transition n_1 correspond trois tirs de la transition x_1 . Ainsi, le comportement du GETDM se déduit facilement de celui du GET projeté sur les transitions x_1, x_2, x_3 .

Au vu du T-invariant de transition $(2 \ 3 \ 1)^t$, le franchissement d'un cycle dans le GETDM se traduit par le franchissement de six cycle dans le GET, ce qui signifie que l'évolution du comportement du GET est six fois plus rapide que celui du GETDM.

Pour évaluer les performances du GETDM de la figure 4.12, on utilise son modèle $(min, +)$ linéaire, ce qui permet de calculer les fréquences de franchissement de chacune de ses transitions, ainsi que son

temps de cycle.

En utilisant le théorème 4.17, on trouve un taux de production du GET de la figure 4.14, obtenu après linéarisation, égal à $\frac{3}{2}$. De l'équation (4.16), on déduit la fréquence de franchissement de chaque transition du GETDM, à savoir : $\lambda_{m_1} = \frac{1}{2}$, $\lambda_{m_2} = \frac{3}{4}$, $\lambda_{m_3} = \frac{1}{4}$.

Par conséquent, l'équation (4.4) permet de déduire que $TC_m = 4$ unités de temps.

Une confirmation de ces résultats peut être faite directement à travers le graphe d'évolution du marquage du GETDM, soit :

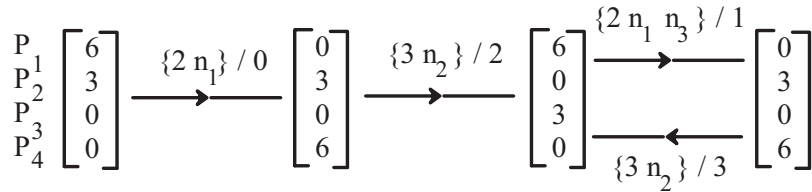


Figure 4.15 – Graphe d'évolution du marquage de la figure 4.12 .

4.7 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'évaluation de performances des GETDM. Pour pouvoir appliquer la théorie spectrale dans le but d'évaluer les performances de ces graphes, deux méthodes de linéarisation ont été proposées. Ces méthodes ont pour objectif d'obtenir un modèle admettant une représentation $(min, +)$ linéaire. Le principe de la première méthode s'applique sous réserve de satisfaire une condition sur le marquage initial du GETDM. Une alternative est proposée dans le cas où la condition n'est pas satisfaite, il en résulte une approximation par valeurs inférieure et supérieure. La deuxième méthode concerne les GETDM ayant une transition pour laquelle sa composante de T-invariant est égale à un. Il a été montré que ces transitions présentent un intérêt au sens où elles permettent de linéariser les équations non linéaires associées aux transitions du graphe.

possèdent un intérêt mathématique, elles permettent de linéariser toute équation récurrente dépendant d'elles.

CHAPITRE 5

Conclusion et Perspectives

Les travaux présentés dans ce mémoire portent sur l'étude des graphes d'événements temporisés avec multiplieurs, la nature des graphes pouvant être discrète, continue, voire hybride. Ces travaux ont pour objectif d'étendre certains résultats établis sur la modélisation, la commande et l'analyse de performances des graphes d'événements temporisés discrets, classiquement étudiés dans le cadre de l'algèbre des dioïdes, aux graphes d'événements temporisés avec multiplieurs.

Ce mémoire s'articule autour de quatre chapitres. Dans le premier chapitre figurent les rappels fondamentaux concernant les propriétés mathématiques des dioïdes, la théorie de la résiduation et un ensemble de définitions sur la théorie des systèmes. Les résultats décrits dans ce chapitre sont classiques, et sont présentés en détail dans [Baccelli et al., 1992].

Le deuxième chapitre traite la modélisation des graphes d'événements temporisés discrets et hybrides avec multiplieurs. Ces graphes n'admettent pas une représentation linéaire dans l'algèbre $(\min, +)$, du fait de la présence de poids sur les arcs. Pour pallier à ce problème de non linéarité, une nouvelle approche de modélisation à base d'opérateurs est proposée. Il est montré que tout graphe d'événements temporisé discret avec multiplieurs admet une représentation d'état dans un dioïde d'opérateurs discrets, noté $\mathcal{D}_{\min}[\delta]$. Le même principe de modélisation est appliqué aux graphes d'événements temporisés continus avec multiplieurs à travers l'utilisation d'une structure algébrique à base d'opérateurs continus, noté $\mathcal{C}_{\min}[\delta]$. Pour modéliser les graphes d'événements temporisés hybrides avec multiplieurs, nous introduisons deux opérateurs modélisant les interfaces continue-discrète et discrète-continue. Grâce au modèle d'état proposé (cf. équation (2.20) dans § 2.4.2), nous unifions la présentation des parties discrète et continue dans un seul formalisme, ce qui permet de proposer des commandes comportant des aspects discret et continu au sein d'un même modèle.

Une commande est calculée, dans le chapitre trois, pour les graphes d'événements temporisés discrets avec multiplieurs, dans le dioïde $\mathcal{D}_{\min}[\delta]$. Cette commande est en juste-à-temps, il s'agit de calculer

une trajectoire d'entrée (commande) telle que la réponse en sortie suive "au mieux" une trajectoire de consigne correspondant au comportement désiré en sortie du système. Une extension de cette commande aux graphes d'événements temporisés hybrides avec multiplieurs a été proposée. Dans ce domaine, commande des GET avec multiplieurs, une perspective de recherche peut être envisagée. Elle concerne la commande par modèle de référence [Cottenceau et al., 2001] de graphes d'événements temporisés continus avec multiplieurs. Pour cette classe de graphes, contrairement aux graphes d'événements temporisés discrets avec multiplieurs, le calcul de A^* (A , matrice d'état), pour exprimer la relation de transfert (cf. équation 2.18 dans 2.3.3), trouve une expression systématique. Le principe de cette commande est comme suit : partant d'un système représentable par un graphe d'événements temporisé continu avec multiplieurs, on cherche à synthétiser un correcteur (une loi de commande), descriptible par un graphes d'événements temporisés continu avec multiplieurs, de telle sorte que le comportement entrée-sortie du système corrigé soit aussi proche que possible de celui d'un modèle de référence donné. Ceci nous amènerait à considérer différents types de commande telles que la correction avec précompensateur et la correction par retour de sortie. Pratiquement, cela revient à étudier comment la dynamique d'un graphe d'événements temporisé continu avec multiplieurs (système nominal) peut être modifié par l'ajout d'un autre graphe d'événements temporisé continu (le correcteur) placé soit en amont, soit dans la boucle de retour de sortie vers l'entrée.

Le problème d'évaluation de performances des graphes d'événements temporisés discrets avec multiplieurs fait l'objet du quatrième chapitre. Nous proposons deux méthodes de linéarisation de ces graphes, dans le but d'obtenir des représentations linéaires dans l'algèbre $(min, +)$, et d'appliquer certains résultats de base de la théorie spectrale, habituellement utilisée pour évaluer les performances des graphes d'événements temporisés ordinaires. Les résultats de ce chapitre ouvrent la voie à d'autres développements liés à l'analyse des performances de ces modèles. En particulier, le calcul du temps de cycle pour tout graphe d'événements temporisé discret avec multiplieurs est, à notre connaissance, un problème ouvert. Aussi, comme perspective, la recherche d'une condition nécessaire et suffisante de linéarisation de ces graphes pourrait permettre d'appliquer tous les résultats développés dans le cadre de la théorie des systèmes linéaires dans les dioïdes.

Une autre perspective de recherche porte sur l'optimisation des ressources des graphes d'événements temporisés discrets avec multiplieurs. Le principe consiste à minimiser le nombre de ressources (machines, palettes, capacité de stocks...) d'un système tout en assurant un temps de cycle souhaité. Pour cela, on pourrait utiliser la méthode de linéarisation "exacte" des graphes d'événements temporisés discrets avec multiplieurs. L'utilisation de l'équation (4.7), décrite dans § 4.4.1, permettrait de déduire le graphe d'événements temporisé équivalent, le marquage de chaque place p de ce GET équivalent, noté

m'_p , étant égal à $\frac{1}{\theta_q} \lfloor \frac{m_p}{M_{qp}} \rfloor$, où m_p est le marquage, à déterminer, de la place p du graphe d'événements temporisé avec multiplieurs. Il resterait alors à utiliser une méthode permettant de traiter le problème d'optimisation de ressources dans le cas des graphes d'événements temporisés ordinaires. On peut citer par exemple une méthode proposée par Henri Hillion et Jean-Marie Proth [Hillion, 1989], basée sur la théorie des graphes, ou une méthode, proposée par Stéphane Gaubert [Gaubert, 1995], basée sur la théorie spectrale développée dans l'algèbre $(\min, +)$; ces deux méthodes aboutissant à la résolution d'un problème de programmation linéaire, sans contraintes.

Bibliographie

- [Alla and David, 1998] Alla, H. and David, R. (1998). Continuous and Hybrid Petri Nets. *Journal of Circuits, Systems and Computers*, 8(1):159–188.
- [Audry et al., 1994] Audry, N., Demongodin, I., and Prunet, F. (1994). Modellings of High Throughput Production Lines by Using Generic Models Described in Batches Petri Nets. In *International Conference on Robotics and Automation, IEEE/RA*, pages 807–812, San-Diego.
- [Baccelli et al., 1992] Baccelli, F., Cohen, G., Olsder, G.-J., and Quadrat, J.-P. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Wiley and Sons.
- [Balbo and Silva, 1998] Balbo, G. and Silva, M. (1998). *Performance Models for Discrete Event Systems with Synchronizations*. MATCH Advanced Schools, Editorial Kronos, Zaragoza, Espagne.
- [Baynat, 2000] Baynat, B. (2000). *Théorie des files d’attentes : des chaines de Markov aux réseaux à forme produit*. Ed Hermés.
- [Blyth and Janowitz, 1972] Blyth, T.-S. and Janowitz, M.-F. (1972). *Residuation Theory*. Pergamon press.
- [Boimond and Ferrier, 1996] Boimond, J.-L. and Ferrier, J.-L. (1996). Internal Model Control and Max-Algebra. *IEEE Trans. on Automatic Control*, 41(3):457–461.
- [Branicky, 1995] Branicky, M. (1995). *Studies in Hybrid Systems : Modeling, Analysis and Control*. PhD thesis, MIT.
- [Champagnat et al., 1998] Champagnat, R., Esteban, P., Pingaud, H., and Valette, R. (1998). Modeling and Simulation of Hybrid Systems Through PR/TR PN-DAE Model. In *International Conference on Automation of Mixed process : Hybrid Dynamic ADPM conference*, pages 131–137.
- [Chao et al., 1993] Chao, D., Zhou, M., and Wang, D. (1993). Multiple Weighted Marked Graphs. In *IFAC 12th Triennial World Congress*, pages 371–374, Sydney, Australie.
- [Chretienne, 1983] Chretienne, P. (1983). *Les réseaux de Petri temporisés*. Thèse d’état, Université de Paris VI.
- [Ciardo et al., 1997] Ciardo, G., Nicol, D., and Trivedi, K. (1997). Discrete-Event Simulation of Fluid Stochastic Petri Nets. In *IEEE/PNPM*, pages 217–225, France.
- [Cohen et al., 1998a] Cohen, G., Gaubert, S., and Quadrat, J.-P. (1998a). Algebraic System Analysis of Timed Petri Nets. In *Idempotency*, Publications of the Newton Institute. Cambridge University Press.
- [Cohen et al., 1998b] Cohen, G., Gaubert, S., and Quadrat, J.-P. (1998b). Timed-Event Graphs with Multipliers and Homogeneous Min-Plus Systems. *IEEE TAC*, 43(9):1296–1302.

- [Cohen et al., 1998c] Cohen, G., Gaubert, S., and Quadrat, J.-P. (1998c). Timed-Event Graphs with Multipliers and Homogenous Min-Plus Systems. *IEEE Trans. on Automatic Control*, 43(9):1296–1302.
- [Cohen et al., 1989] Cohen, G., Moller, P., Quadrat, J.-P., and Viot, M. (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proceedings: Special issue on Discrete Event Systems*, 77(1):39–58.
- [Compos et al., 1991] Compos, J., Chiola, G., and Silva, M. (1991). Properties and Performance Bounds for Closed Free Choice Synchronized Monoclass Queueing Networks. *IEEE Trans. on Automatic Control*, 36(12):1368–1382.
- [Cottenceau, 1999] Cottenceau, B. (1999). *Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes*. Thèse, LISA - Université d'Angers.
- [Cottenceau et al., 2001] Cottenceau, B., Hardouin, L., Boimond, J.-L., and Ferrier, J.-L. (2001). Model Reference Control for Timed Event Graphs in Dioid. *Automatica*, 37:1451–1458.
- [Cottenceau et al., 2000] Cottenceau, B., Lhommeau, M., Hardouin, L., and Boimond, J.-L. (2000). Data Processing Tool for Calculation in Dioid. In *Workshop On Discrete Event Systems (WODES'2000)*, Ghent, Belgium.
- [David, 1997] David, R. (1997). Modeling of Hybrid Systems Using Continuous and Hybrid Petri Nets. In *7th International Workshop on Petri Nets and Performance Models*, pages 47–58.
- [David and Alla, 1987] David, R. and Alla, H. (1987). Continuous Petri Nets. In *8th European workshop on application and theory of Petri Nets*, Saragoza-Spain.
- [David and Alla, 1992] David, R. and Alla, H. (1992). *Du Grafcet au réseaux de Petri*. Editions Hermès, Paris.
- [Demongodin et al., 1999] Demongodin, I., Mostefaoui, M., and Sauer, N. (1999). Liveness of Continuous Weighted Marked Graphs. In *3rd IMACS/IEEE CSCC99*, pages 2640–2649.
- [Gaubert, 1992] Gaubert, S. (1992). *Théorie des systèmes linéaires dans les dioïdes*. Thèse, Ecole des Mines de Paris.
- [Gaubert, 1995] Gaubert, S. (1995). Resource Optimization and (min,+) Spectral Theory. *IEEE Trans. on Automatic Control*, 40(11):1931–1934.
- [Giua and Usai, 1998] Giua, A. and Usai, E. (1998). Modeling Hybrid Systems by High-Petri Nets. volume 32.
- [Gondran and Minoux, 1985] Gondran, M. and Minoux, M. (1985). *Graphes et algorithmes*. Eyrolles, Paris, France.

- [Hamaci et al., 2004a] Hamaci, S., Boimond, J.-L., and Lahaye, S. (2004a). On Modeling and Control of Discrete Timed Event Graphs with Multipliers Using $(\min, +)$ Algebra. In *1st International Conference on Informatics in Control (ICINCO'04)*, volume 3, pages 32–37, Setùbal, Portugal.
- [Hamaci et al., 2005a] Hamaci, S., Boimond, J.-L., and Lahaye, S. (2005a). On Modeling and Control of Hybrid Timed Event Graphs with Multipliers Using $(\min, +)$ Algebra. *Journal of Discrete Event Dynamic Systems (à paraître)*.
- [Hamaci et al., 2005b] Hamaci, S., Boimond, J.-L., and Lahaye, S. (2005b). Performance Analysis of Timed Event Graphs with Multipliers Using $(\min, +)$ Algebra. In *2st International Conference on Informatics in Control Automation and Robotics (ICINCO'05)*, volume 4, pages 16–21, Barcelone, Espagne.
- [Hamaci et al., 2004b] Hamaci, S., Boimond, J.-L., Lahaye, S., and Mostefaoui, M. (2004b). On the Linearizability of Discrete Timed Event Graphs with Multipliers Using $(\min,+)$ Algebra. In *7th international Workshop on Discrete Event Systems (WODES'04)*, pages 367–372, France.
- [Hamaci et al., 2004c] Hamaci, S., Boimond, J.-L., Lahaye, S., and Mostefaoui, M. (2004c). Sur la linéarisation des graphes d'événements temporisés discrets avec multipliers dans l'algèbre $(\min,+)$. *Revue e-STA, Sciences et Technologies de l'Automatique*, 1(3).
- [Hillion, 1989] Hillion, H. (1989). *Modélisation et analyse des systèmes de production discrets par les réseaux de Petri temporisés*. Thèse, Université de Paris IV, France.
- [Komenda et al., 1998] Komenda, J., Zerhouni, N., and Moudni, A. (1998). The Use of Conventional and Minplus Algebra for the Modeling of Hybrid Petri Nets. *Cybernetics and Systems*, 29(8):869–884.
- [Lahaye, 2000] Lahaye, S. (2000). *Contribution à l'étude des systèmes linéaires non stationnaires dans l'algèbre des dioïdes*. Thèse, LISA - Université d'Angers.
- [Lüders and Santos-Mendes, 2002] Lüders, R. and Santos-Mendes, R. (2002). Multivariable Control of Discrete Event Systems in Dioids. In *IFAC World Congress on Automatic Control*, Barcelona.
- [Libeaut and Loiseau, 1996] Libeaut, L. and Loiseau, J.-J. (1996). On the Control of Timed Event Graphs. In *Proceedings of WODES'96*, Edinburgh.
- [Maia et al., 2003] Maia, C.-A., Hardouin, L., Santos-Mendes, R., and Cottenceau, B. (2003). Optimal Closed-Loop Control of Timed-Event Graphs in Dioids. *IEEE Trans. on Automatic Control*, 48:2284–2287.
- [Menguy, 1997] Menguy, E. (1997). *Contribution à la commande des systèmes linéaires dans les dioïdes*. Thèse, LISA - Université d'Angers.
- [Menguy et al., 2000a] Menguy, E., Boimond, J.-L., Hardouin, L., and Ferrier, J.-L. (2000a). A First Step Towards Adaptive Control for Linear in Max Algebra. *Journal of Discrete Event Dynamic Systems*, 10(4):347–367.

- [Menguy et al., 2000b] Menguy, E., Boimond, J.-L., Hardouin, L., and Ferrier, J.-L. (2000b). Just In Time Control of Timed Event Graphs: Update of Reference Input, Presence of Uncontrollable Input. *IEEE Trans. on Automatic Control*, 45(11):2155–2158.
- [Merlin, 1979] Merlin, P. (1979). Methodology for the Design and Implementation of Communication Protocole. *IEEE Transaction on Communication*, 24(6).
- [Munier, 1993] Munier, A. (1993). Régime asymptotique optimal d'un graphe d'événements temporisé généralisé: application à un problème d'assemblage. *APII*, 27(5):487–513.
- [Murata, 1989] Murata, T. (1989). Petri Nets : Properties, Analysis and Applications. In *IEEE Proceedings : Special issue on Discrete Event Systems*, volume 77, pages 541–581.
- [Nakamura and Silva, 1999] Nakamura, M. and Silva, M. (1999). Cycle Time Computation in Deterministically Timed Weighted Marked Graphs. In *IEEE-ETFA*, pages 1037–1046.
- [Petri, 1962] Petri, C.-A. (1962). *Kommunikation mit Automaten*. Thèse, Institut für Instrumentelle Mathematik, Bonn, Germany.
- [Proth and Xie, 1995] Proth, J. and Xie, X. (1995). *Les réseaux de Petri pour la conception et la gestion des systèmes de production*. Edition MASSON.
- [Ramchandani, 1974] Ramchandani, C. (1974). *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. Thèse d'état, Université de Paris VI.
- [Sauer, 2003] Sauer, N. (2003). Marking Optimization of Weighted Marked Graphs. *Journal of Discrete Event Dynamic Systems*, 13:245–262.
- [Sauer, 2004] Sauer, N. (2004). *Contribution à l'évaluation et l'optimisation des systèmes de production*. Rapport d'habilitation à diriger des recherches, Ecole des mines de Nantes.
- [Teruel et al., 1992] Teruel, E., Chrzasowski-Wachtel, P., Colom, J., and Silva, M. (1992). On Weighted T-Systems. In *Advances in Petri nets, Lecture Notes in Computer Science*, volume 616, pages 348–367.
- [Trouillet, 2003] Trouillet, B. (2003). *Contribution à la commande des systèmes linéaires dans les dioïdes*. Thèse, Ecole Centrale de Lille, France.
- [Trouillet et al., 2002] Trouillet, B., Benasser, A., and Gentina, J.-C. (2002). On the Modeling of the Dynamical Behavior of Weighted T-Systems. In *APII-JESA*, volume 36(7), pages 931–944.
- [Willems, 1991] Willems, J. (1991). Paradigms and Puzzles in the Theory of Dynamical Systems. *IEEE Trans. on Automatic Control*, volume 36.
- [Zerhouni and Alla, 1992] Zerhouni, N. and Alla, H. (1992). Sur l'analyse des lignes de fabrication par réseaux de Petri continus. In *APII*, pages 253–276.

Étude des graphes d'événements temporisés avec multiplieurs dans l'algèbre $(\min, +)$.

Samir HAMACI

Résumé

Les graphes d'événements temporisés avec multiplieurs forment une sous classe des réseaux de Petri dont le comportement ne peut pas être décrit par des équations linéaires dans l'algèbre $(\min, +)$ de par la présence des poids sur les arcs.

Ce mémoire est consacré à l'étude des systèmes dynamiques, modélisés par des graphes d'événements temporisés discrets, voire hybrides avec multiplieurs. Nous abordons le problème de la représentation entrée-sortie de ces graphes dans l'algèbre $(\min, +)$. Pour pallier le problème de non-linéarité, une approche de modélisation à base d'opérateurs est proposée. La commande en juste-à-temps est calculée pour les graphes d'événements temporisés discrets avec multiplieurs dans un dioïde d'opérateurs. Une extension de cette commande aux graphes d'événements temporisés hybrides avec multiplieurs est présentée. Est abordé ensuite, l'analyse des performances des graphes d'événements temporisés discret avec multiplieurs. Nous proposons deux méthodes de linéarisation de ces graphes, dans le but d'obtenir des représentations linéaires dans l'algèbre $(\min, +)$, ceci afin d'appliquer certains résultats de base de la théorie spectrale, habituellement utilisée pour évaluer les performances des graphes d'événements temporisés ordinaires.

Mots-clés : graphes d'événements temporisés discrets avec multiplieurs, graphes d'événements temporisés hybrides avec multiplieurs, dioïde, commande en juste-à-temps, algèbre $(\min, +)$, linéarisation, temps de cycle.

Abstract

Timed event graphs with multipliers form a subclass of Petri nets whose behavior can not be described by linear equations in $(\min, +)$ algebra. This nonlinearity is due to the presence of weights on arcs.

This report is devoted to the study of dynamic systems modelled by discrete timed event graphs with multipliers and hybrid timed event graphs with multipliers. We tackle the problem of input-output representation of these graphs in $(\min, +)$ algebra. To mitigate the problem of nonlinearity, an approach of modelling, based on operators, is proposed. A just in time control is calculated for discrete timed event graphs with multipliers. An extension of this control to the hybrid Timed event graphs with multipliers is presented. Then, the performance analysis of timed event graphs with multipliers, is tackled. We propose two linearization methods of these graphs, in the purpose to obtain a linear representation in $(\min, +)$ algebra, and to apply some basic results of the spectral theory, usually used to evaluate the performances of ordinary timed event graphs.

Keywords: discrete timed event graphs with multipliers, hybrid timed event graphs with multipliers, dioid, just in time control, $(\min, +)$ algebra, linearization, cycle time.