

*RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE*

*Ministère de l'Enseignement Supérieur et de la Recherche Scientifique*

**UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU**

**FACULTÉ DE GENIE ELECTRIQUE ET DE L' INFORMATIQUE**

**DÉPARTEMENT D'AUTOMATIQUE**

## **THÈSE DE DOCTORAT**

Présentée pour obtenir le diplôme de

**DOCTORAT**

Spécialité : AUTOMATIQUE

Par :

**Karima TEBANI**

**THEME**

**Commande temps réel d'une classe de systèmes**

**à événements discrets sous contraintes**

**par l'algèbre des dioïdes**

DEVANT LE JURY :

Président	Ahmed MAIDI	Professeur, Université Mouloud Mammeri de Tizi Ouzou
Rapporteur	Redoane KARA	Professeur, Université Mouloud Mammeri de Tizi Ouzou
Co/Rapporteur	Saïd AMARI	HDR, U/Paris XIII/ France
Examineur	Hassane ALLA	Professeur, INP Grenoble/ France
Examineur	Brahim BOUZOUIA	Directeur de recherche, CDTA/Alger
Examineur	Brahim OUKACHA	MCA, Université Mouloud Mammeri de Tizi Ouzou

ANNÉE :02 NOVEMBRE 2017



# Remerciements

Ce travail de thèse a été effectué au sein du Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), de l'université Mouloud Mammeri de Tizi-Ouzou, en collaboration avec l'université Paris XIII. Ce travail de thèse n'aurait pas pu être réalisé sans l'aide précieuse de plusieurs personnes que je souhaite remercier vivement un par un :

Je tiens tout d'abord à exprimer mes plus vifs remerciements à M.KARA Redouane, Professeur à l'université Mouloud Mammeri de Tizi Ouzou, pour son encadrement, sa collaboration totale sans limites, sa patience, ses conseils très précieux et son aide, tout au long de ce travail.

J'adresse de chaleureux remerciements à M.AMARI Said, Maître de conférences à l'université Paris XIII, pour avoir dirigé mes recherches, Je tiens à le remercier également pour son attention de tout instant sur mes travaux et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. Son énergie et sa confiance ont été des éléments moteurs pour moi. C'était vraiment un grand plaisir pour moi de travailler avec lui.

Je remercie M.MAIDI Ahmed, Professeur à l'université Mouloud Mammeri de Tizi Ouzou, pour avoir accepté d'examiner ce travail et de présider le jury de soutenance de cette thèse.

Je souhaite aussi exprimer ma profonde gratitude à Monsieur HACENE Alla, Professeur à l'université de Nantes, qui m'a fait l'honneur d'accepter d'être membre du jury. Je lui suis très reconnaissante pour l'intérêt qu'il a porté à mes travaux. Son avis, ses conseils et ses remarques seront sans doute très constructifs et précieux.

J'adresse de sincères remerciements à M.Bouzouia Ibrahim, Directeur de recherche au

centre de recherche CDTA d'avoir accepté d'examiner ce travail. Je le remercie infiniment pour le temps qu'il m'a consacré et pour l'intérêt qu'il a porté à évaluer mon travail.

Je tiens également à remercier M.OUKACHA Brahim Maître de conférences Classe B à L'université Mouloud Mammeri de Tizi Ouzou, qui a accepté d'être membre du jury de soutenance de ma thèse.

Je désire grandement remercier M.GAHAM Mehdi, chef de projet au centre de recherche CDTA pour sa disponibilité, sa compréhension, sa confiance en moi et d'avoir pris le temps de relire cette thèse et chacun de mes articles. Je tiens à le remercier pour son soutien et son encouragement.

Je tiens à remercier également Mme Semar-Bitah Kahina pour son soutien et les bons moments qu'on a partagé et d'avoir pris le temps de relire ce manuscrit.

Je remercie tous les thésards du Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), pour la bonne ambiance de travail mais également pour les nombreux beaux moments passés ensemble. Entre autres Ouerdia, Sofiane, Yacine, Sassa, Nabila, Hakima, Aldjia.

Je remercie tous les membres de l'équipe Systèmes Robotisés de Production de centre de recherche CDTA en particulier M.Bachir pour l'ambiance de travail.

Je tiens à remercier tout particulièrement M.Ravi pour les nombreuses relectures et corrections de mon anglais.

Je remercie tous mes amis, qui de près ou de loin m'ont supportée, soutenue et encouragée tout au long de ces années, M.Assia, S.Kahina Y.Hakima, F.Idir, T.Malika, M.Kahina, L.Roza, M.Hayet, M.Nassiba, A.Souad, A.Dyhia,...

Mes derniers remerciements, et non des moindres, je les destine à mes Parents, pour leur soutien indéfectible, sans vous je n'en serais pas là aujourd'hui sans oublier mes frères Amar et Mahdi et à ma sœur Nadia, pour leurs soutiens et leurs encouragements.

# Introduction Générale

Sous l'appellation Systèmes à Événements discrets (SED) sont regroupés des systèmes, généralement, de conception humaine comme : les systèmes de production (ateliers flexibles, lignes d'assemblage) [17], [18], les réseaux de communication (réseaux informatiques) [36] et les systèmes de transport (routier, ferroviaire ou aérien), [30].

L'intégration des technologies de l'information et de la communication dans les différents domaines de l'ingénierie conduit à plus de complexité, notamment l'aspect de commande temps réel de ces systèmes. Cet aspect se présente par des spécifications (contraintes) qui peuvent se présenter sous diverses formes (états interdits, intervalles de temps, durée de validité ...). Afin de garantir le bon fonctionnement de ces systèmes, plusieurs études ont été menées par la communauté des systèmes à événements discrets sur la synthèse de contrôleur qui permettent la satisfaction des différentes spécifications imposées.

Dans cette thèse, on s'intéresse à la commande des systèmes à événements discrets temporisés soumis à des spécifications de type : contraintes de temps ou contrainte de ressources. Dans un premier temps, nous avons traité des spécifications de type contraintes de temps. En effet, le paramètre temps dans les systèmes à événements discrets est un paramètre primordial. Les opérations doivent avoir lieu dans des délais précis et, parfois, elles doivent se terminer avant une date fixée à l'avance, le non-respect de ce paramètre (temps) conduit à des défaillances et parfois à des conséquences graves. On retrouve ces contraintes dans les systèmes de production incluant des traitements thermiques ou chimiques [1, 33], les systèmes de transport urbain ou ferroviaire [29, 52], et dans les systèmes de commande en réseau [21].

Afin de garantir des spécifications temporelles, plusieurs approches, basées sur des ou-

tils de modélisation différents (automates, réseaux de Petri, algèbre des dioïdes,...), ont été proposées dans la littérature. La contribution dans [14] est basée sur les automates temporisés, les auteurs ont synthétisé un contrôleur afin de garantir la satisfaction des contraintes imposées au système considéré. Cette approche donne une bonne solution de commande par supervision. Toutefois, comme toutes les méthodes basées sur les automates, elle souffre du problème d'explosion combinatoire souvent rencontré lorsqu'on utilise des automates à états. Afin d'éviter ce problème, plusieurs approches basées sur les réseaux de Petri ont été développées. On peut citer par exemple, [15, 34, 2, 1, 38]. Les auteurs dans [15, 34], s'intéressent, plus précisément, à des problèmes de rejet de perturbations et de poursuite d'un modèle de référence. La contrainte temporelle est une condition supplémentaire du problème. Les auteurs dans [2, 8, 3] ont proposé des méthodes de synthèse de contrôleur en boucle fermée et calculent des gains de retour d'état permettant de garantir des spécifications temporelles imposées au système modélisé. Ces approches utilisent les graphes d'événements temporisés (GETs), une sous-classe des réseaux de Petri et l'algèbre Max-Plus.

Notons toutefois que les contraintes considérées dans ces études sont particulières. Il s'agit plus précisément de contraintes de temps imposées sur des places (indépendantes) du GET considéré.

Dans cette thèse, nous généralisons l'approche proposée dans [2], pour le cas de graphes d'événements temporisés soumis à des contraintes de temps imposées sur des chemins. Les contraintes sur les places sont dans notre cas un cas particulier. On rencontre ce type de spécifications temporelles (contraintes sur les chemins) dans de nombreux systèmes à événements discrets tels que les systèmes de production, les systèmes de transport et les systèmes de commande en réseau.

Dans cette thèse, on s'intéresse à des architectures d'automatisation distribuées en réseau soumis à des contraintes de temps imposées sur leurs temps de réponse [49].

Nous avons par la suite, abordé le problème de ressources limitées (stock borné par exemple); ces contraintes s'expriment par des inégalités linéaires sur le vecteur de marquage des graphes d'événements temporisés. Ce problème est souvent traité comme étant

---

un problème d'état interdit [53], [24], [47], [10], [26]. Le problème d'états interdit consiste à empêcher l'occurrence d'un ensemble d'événements qui conduisent à violer les contraintes de marquage imposées. Plusieurs travaux sont dédiés à ce problème [54, 41, 50, 51].

Seulement, la majorité de ces travaux ne prennent pas en considération le paramètre temps ni dans l'étape modélisation ni dans l'étape de synthèse de contrôleur. Une approche basée sur les réseaux de Petri temporisés a été proposée dans [6], cette méthodologie est basée sur l'approche de Ramadge et Wonham et la théorie des régions, et sert à calculer un ensemble de places de contrôle à ajouter au modèle de réseau Petri afin de résoudre le problème des états interdites. Dans [9], un superviseur est synthétisé pour le contrôle des graphes marqués temporisés basés sur la contrainte d'exclusion de marquage est développé. Ces deux approches sont basées sur la théorie des graphes ce qui rend l'application de ces méthodes pour des systèmes complexes compliquées et parfois impossible.

Le travail présenté dans cette partie de la thèse, permet de synthétiser des superviseurs qui garantissent le respect des contraintes de marquage. La méthodologie de synthèse prend explicitement en compte le facteur temps dans l'étape de modélisation et dans l'étape de synthèse. De ce fait, les contrôleurs calculés sont aussi temporisés.

Cette thèse est décomposée en trois chapitres. Nous donnons dans cette introduction les grandes lignes du plan qui sera détaillé par la suite.

Dans le premier chapitre, nous rappelons quelques définitions et notations de base concernant l'algèbre des dioïdes, les réseaux de Petri et graphes d'événements temporisés. par la suite, nous montrons comment décrire le comportement d'un graphe d'événement temporisé par des équations linéaires dans l'algèbre Max-Pus (resp. l'algèbre Min-Plus ) basées sur les fonctions dateurs (resp. compteurs).

Dans le deuxième chapitre, nous représentons dans la première partie notre première contribution concernant la commande des graphes d'événements temporisés sous contraintes de temps. Nous utilisons les GETs pour la modélisation de système à commander et l'algèbre Max-Plus pour la représentation de la dynamique. Nous calculons des lois de commande qui garantissent les contraintes de temps qui sont imposées sur certains chemins du graphe qui modélise le système à commander. Par la suite, dans la deuxième partie de

ce chapitre, nous nous intéressons à la commande d'une Architecture d'Automatisation en Réseau (AAR) soumis à une contrainte de temps imposé sur le temps de réponse. Nous représentons tout d'abord et de manière détaillée les différentes composantes et le fonctionnement d'une AAR dans lequel les contrôleurs logiques communiquent avec les modules d'entrées/sorties via un réseau de communication TCP/IP. Ensuite, nous passons à la modélisation de l'AAR avec un graphe d'événement temporisé et la représentation de la dynamique de ce graphe dans l'algèbre Max-Plus afin d'appliquer les résultats de notre contribution pour satisfaire les contraintes de temps imposées sur l'AAR.

Le troisième chapitre est consacré à la synthèse de commande des GETs sous contrainte de marquage. Dans la première partie de ce chapitre, nous représentons le problème des contraintes de marquage dans les systèmes à événements discrets et nous montrons comment les représentés par des inéquations linéaires dans l'algèbre Min-Plus. Notre approche de commande des GETs soumis à des contraintes de marquage est ensuite introduite. Nous traitons dans un premier temps, des graphes d'événement temporisés avec une seule transition de commande. L'approche est ensuite généralisée au cas des graphes d'événements temporisés avec plusieurs transitions de commande.

Nous terminons cette thèse par une conclusion générale et quelques perspectives.

# Table des matières

<b>1</b>	<b>Modélisation des systèmes à événements discrets dans les dioïdes</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Théorie des dioïdes . . . . .	2
1.2.1	Rappels algébriques . . . . .	2
1.2.2	Propriétés des dioïdes . . . . .	3
1.2.3	Résolution des équations dans les dioïdes . . . . .	6
1.3	Réseaux de Petri et graphe d'événements temporisé . . . . .	6
1.3.1	Réseaux de Petri (RDP) . . . . .	6
1.3.2	Graphe d'événements temporisé . . . . .	10
1.4	Conclusion . . . . .	20
<b>2</b>	<b>Commande Max-Plus des systèmes à événements discrets sous contraintes de temps</b>	<b>21</b>
2.1	Introduction . . . . .	21
2.2	Motivations et problématique . . . . .	24
2.2.1	Problème de commande des GETs sous contraintes temporelles imposées sur des places [2] . . . . .	25
2.2.2	Contrainte temporelle imposée sur un chemin d'un GET . . . . .	27
2.3	Commande temporelle des systèmes Max-Plus linéaires . . . . .	28
2.3.1	Commande monovariante . . . . .	28
2.3.2	Commande multivariante . . . . .	35
2.3.3	Application sur une architecture d'automatisation en réseau . . . . .	44

2.4	Conclusion . . . . .	56
<b>3</b>	<b>Commande des graphes d'événements temporisés sous contraintes de marquages</b>	<b>57</b>
3.1	Introduction . . . . .	57
3.2	Problème de contraintes de marquages (état interdit) . . . . .	60
3.2.1	Formulation du problème de commande . . . . .	60
3.3	Commande des graphes d'événements temporisés sous contraintes de marquage . . . . .	63
3.4	Conclusion . . . . .	87



---

$SED$ :	Système à Evénements Discret.
$RDP$ :	Réseau de Petri.
$GET$ :	Graphe d'Événements Temporisé.
$P$ :	Ensemble non vide de places dun réseau de Petri.
$T$ :	Ensemble non vide de transitions dun réseau de Petri.
$M_{ij0}$ :	Nombre de jetons initialement contenus dans la place $p_{ij}$ .
$M_{ij}(t)$ :	Nombre de jetons contenus dans la place $p_{ij}$ à l'instant $t$ .
$\bar{\mathbb{R}}_{min}$ :	Algèbre Min-Plus.
$\bar{\mathbb{R}}_{max}$ :	Algèbre Max-Plus.
$\oplus$ :	addition dans un dioïde.
$\otimes$ :	multiplication dans un dioïde.
$\varepsilon$ :	Elément neutre pour la loi $\oplus$ .
$e$ :	Elément neutre pour la loi $\otimes$ .
$D$ :	Dioïde. $\leq$ relation d'ordre liée à un dioïde.
$a^*$ :	étoile de kleene.
$x_i(t)$ :	fonction compteur associée à la transition $t_i$ d'un graphe d'événements temporisé.
$u(t)$ :	fonction compteur associée à la transition d'entrée $t_u$ d'un graphe d'événements temporisé.
$x_i(k)$ :	fonction dateur associée à la transition $t_i$ d'un graphe d'événements temporisé.
$u(k)$ :	fonction dateur associée à la transition d'entrée $t_u$ d'un graphe d'événements temporisé.
$\mu_0$ :	vecteur marquage initial.

# Table des figures

1.1	Assemblage d'une paire de lunette . . . . .	3
1.2	Exemple d'un RDP . . . . .	8
1.3	Représentation de marquage d'un RDP . . . . .	8
1.4	Exemple d'un RDP avec son graphe d'accessibilité . . . . .	9
1.5	Exemple d'un GET . . . . .	12
1.6	modèle GET d'un système composé d'un four et d'un stock . . . . .	17
1.7	Graphe étendu de système montré dans la Fig.1.6 . . . . .	18
1.8	Graphe étendu de système montré dans la Fig.1.6 . . . . .	20
2.1	Contrainte temporelle imposée sur une place . . . . .	25
2.2	Schéma d'un chemin d'un GET . . . . .	27
2.3	Système de transmission d'informations . . . . .	33
2.4	modèle GET de système de transmission d'informations . . . . .	33
2.5	modèle GET de système de transmission d'informations avec la place de contrôle . . . . .	35
2.6	Modèle Graphe d'évènements temporisé d'un système d'assemblage . . . . .	40
2.7	Modèle GET de système d'assemblage avec les places de contrôle . . . . .	43
2.8	Architecture d'automatisation en réseau . . . . .	45
2.9	Comportement d'une architecture d'automatisation en réseau . . . . .	47
2.10	Temps de réponse d'une architecture d'automatisation en réseau . . . . .	48
2.11	Modèle GET d'une architecture d'automatisation en réseau . . . . .	50
2.12	Modèle GET du l'ARR avec la place du contrôle . . . . .	55

3.1	Modèle d'une ligne de production . . . . .	61
3.2	Modèle d'une ligne de production avec un stock limité à 2 . . . . .	61
3.3	modèle d'une ligne de production . . . . .	69
3.4	Graphe de marquage de la Fig.3.3 . . . . .	69
3.5	Graphe étendu de la Fig.3.3 . . . . .	69
3.6	Modèle de ligne de production avec la place de contrôle . . . . .	71
3.7	Graphe de marquage du modèle de la Fig.3.6 . . . . .	71
3.8	Succession de places d'un GET . . . . .	72
3.9	Système manufacturier (deux machines et un robot) . . . . .	74
3.10	Modèle GET de système de la Fig.3.9 . . . . .	75
3.11	Graphe de marquage de Modèle GET de la Fig.3.10 . . . . .	75
3.12	Modèle GET de système de la figure . . . . .	76
3.13	Modèle GET de système manufacturier avec la place de control . . . . .	78
3.14	Graphe de marquage de Modèle GET de la Fig.3.13 . . . . .	78
3.15	Modèle GET de système d'assemblage . . . . .	84
3.16	Graphe de marquage de Modèle GET de la Fig.3.15 . . . . .	84
3.17	Modèle GET de système d'assemblage avec les places de contrôle . . . . .	87
3.18	Graphe de marquage de Modèle GET de la Fig.3.17 . . . . .	88

# Chapitre 1

## Modélisation des systèmes à événements discrets dans les dioïdes

### 1.1 Introduction

Nous nous intéressons dans cette thèse à la commande des systèmes à événements discrets soumis à différents types de spécifications. Afin d'atteindre cet objectif, on a besoin d'outils de modélisation et d'analyse mathématique. Il existe plusieurs formalismes de modélisation et d'étude des systèmes à événements discrets tels que les automates à états finis, les réseaux de Petri (RDP), les chaînes de Markov, l'algèbre des dioïdes, etc. Notre travail est basé sur l'utilisation des graphes d'événements temporisés (une classe des réseaux de Petri) comme outil de modélisation et l'algèbre des dioïdes pour décrire le comportement dynamique de ces graphes par des équations linéaires et pour la synthèse des lois de commande.

Dans ce premier chapitre, nous introduisons les notions de base concernant l'algèbre des dioïdes et les réseaux de Petri qu'on va utiliser dans la suite de ce manuscrit.

## 1.2 Théorie des dioïdes

Dans cette section, nous présentons un rappel sur les concepts de base et les propriétés de l'algèbre des dioïdes. Le but de cette section est d'introduire les notions et notations qui seront utiles pour la commande des graphes d'événements temporisés. Ces rappels sont issus des références suivantes, [11], [12], [4], [39], [48] et [5].

### 1.2.1 Rappels algébriques

**Définition 1.** *Un dioïde est un ensemble  $D$  muni de deux opérations internes notées  $\oplus$  et  $\otimes$  appelées "addition" et "multiplication" respectivement, telles que, pour  $a, b, c \in D$  :*

*l'addition est associative :  $(a \oplus b) \oplus c = a \oplus (b \oplus c)$  ;*

*l'addition est commutative :  $a \oplus b = b \oplus a$  ;*

*l'addition admet un élément neutre : noté  $\varepsilon$  et appelé "zero" :  $a \oplus \varepsilon = a$  ;*

*la multiplication est associative :  $(a \otimes b) \otimes c = a \otimes (b \otimes c)$  ;*

*la multiplication admet un élément neutre : noté  $e$  et appelé "identité" :  $a \otimes e = e \otimes a = a$  ;*

*la multiplication est distributive par rapport à l'addition :  $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$  et  $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$*

*le zéro est absorbant pour la multiplication :  $\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon$  ;*

*l'addition est idempotente :  $a \oplus a = a$ .*

Le dioïde est dit commutatif si la multiplication est commutative ( $a \otimes b = b \otimes a$ ).

**Exemple 1.** 1. *L'ensemble  $(\mathbb{R} \cup \{+\infty, \min, +\})$  est un dioïde commutatif pour lequel  $\varepsilon = +\infty$  et  $e = 0$ . Ce dioïde est noté  $\mathbb{R}_{\min}$ , et traditionnellement appelé " Algèbre Min-Plus ". Dans ce dioïde, la loi  $\oplus$  correspond à l'application minimum ( $7 \oplus 4 = \min(7, 4) = 4$ ), et la loi  $\otimes$  est la somme usuelle ( $4 \otimes 3 = 4 + 3 = 7$ ).*

• **Application 1 :**

*On s'intéresse par exemple à la production des lunettes (Fig.1.1). On peut dire qu'une paire de lunettes est la "somme" d'une paire de verres et d'un cadre. Cette nouvelle "somme", notée  $\oplus$  pour la distinguer du  $+$  habituel, correspond à l'opération*

d'assemblage. La somme  $\oplus$  permet alors d'ajouter des grandeurs qui ne s'expriment pas dans les mêmes unités. Si on a 2 verres et 1 cadre, on ne peut fabriquer qu'une seule paire de lunettes, donc  $1 \oplus 2 = 1$ , ce qui est équivalent dans l'algèbre usuelle au  $\min(2, 1) = 1$ .

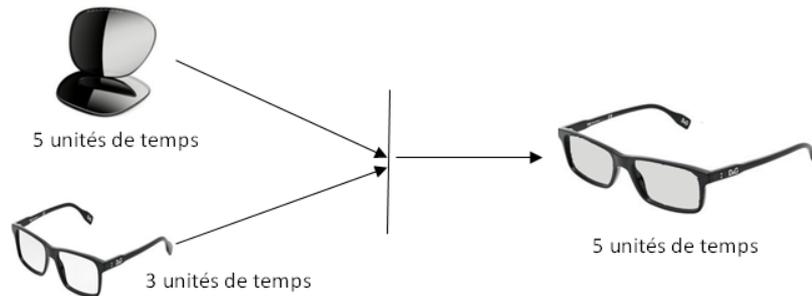


FIG. 1.1: Assemblage d'une paire de lunette

2. L'ensemble  $(\mathbb{R} \cup \{-\infty\}, \max, +)$  est un dioïde commutatif pour lequel  $\varepsilon = -\infty$  et  $e = 0$ . Ce dioïde est noté  $\mathbb{R}_{\max}$ , et traditionnellement appelé " Algèbre Max-Plus " Dans ce dioïde, la loi  $\oplus$  correspond à l'application maximum ( $7 \oplus 4 = \max(7, 4) = 7$ ), et la loi  $\otimes$  est la somme usuelle ( $4 \otimes 3 = 4 + 3 = 7$ )

• **Application 2 :**

On prend toujours l'exemple de lunette et on s'intéresse maintenant au temps de production d'une paire de lunettes. Ce temps est donné par le temps maximal entre le temps de fabrication de deux verres et le temps de production du cadre. On suppose que le temps de production de deux verres est de 5 unités de temps, et le temps de production de cadre est de 3 unités de temps. Alors le temps de production d'une paire de lunettes est de  $(5 \oplus 3) = 5$  ce qui exprime dans l'algèbre usuelle par  $\max(5, 3) = 5$ .

## 1.2.2 Propriétés des dioïdes

### 1.2.2.1. Dioïde complet

Un dioïde  $(D, \oplus, \otimes)$  est dit complet si :

- il est fermé pour les sommes infinies,
- la loi  $\otimes$  distribue (à gauche et à droite) sur les sommes infinies, c'est-à-dire si pour tout  $b \in D$  et tout sous-ensemble  $A \subset D$ ,

$$b \otimes (\oplus_{a \in A} a) = \oplus_{a \in A} (b \otimes a) \text{ et } (\oplus_{a \in A} a) \otimes b = \oplus_{a \in A} (a \otimes b).$$

Un dioïde complet admet un élément maximum  $\bigoplus_{b \in D} b$ , que l'on notera  $T$ . Il est absorbant pour l'addition, autrement dit,  $\forall a \in D, T \oplus a = T$ .

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

**Exemple 2.** (dioïde complet de  $\bar{\mathbb{R}}_{max}$ ).

Le dioïdes  $(\mathbb{R}, -\infty, +\infty, max, +)$ , avec  $\varepsilon = -\infty, e = 0$  et  $T = +\infty$  avec  $(+\infty) + (-\infty) = -\infty$ , est un dioïde complet qu'on notera  $\bar{\mathbb{R}}_{max}$ .

Dans l'algèbre Max-Plus on a aussi par hypothèse :  $\varepsilon \otimes (+\infty) = (-\infty) + (+\infty) = \varepsilon = (-\infty)$ .

### 1.2.2.2. Relation d'ordre dans un dioïde

**Définition 2.** (Relation d'ordre).

Une relation  $\mathcal{R}$  sur un ensemble  $\mathbb{S}$  est une relation d'ordre si les trois axiomes suivants sont vérifiés  $\forall x, y, z \in S$  :

- $R$  est réflexive :  $x R x$
- $R$  est transitive : si  $x R y$  et  $y R z$ , alors  $x R z$
- $R$  est antisymétrique : si  $x R y$  et  $y R x$ , alors  $x = y$ .

Dans un dioïde  $D$  donné, la propriété d'idempotence de la loi additive  $\oplus$  induit une relation d'ordre, notée  $\leq$  définie par  $\forall (a, b) \in D^2, a \leq b \iff a \oplus b = b$ .

De plus, cette relation d'ordre est compatible avec les lois de structure de  $D$ , c'est-à-dire,

$$\forall (a, b, c) \in D^3, a \leq b \implies a \oplus c \leq b \oplus c$$

$$a \leq b \implies a \otimes c \leq b \otimes c \text{ et } c \otimes a \leq c \otimes b$$

Dans  $\bar{\mathbb{R}}_{max}$ , la relation d'ordre naturel coïncide avec l'ordre usuel. Dans  $\bar{\mathbb{R}}_{min}$  l'ordre naturel correspond à l'ordre usuel,  $x \leq y \iff x \oplus y = \min(x, y) = y$ .

**Exemple 3.** La relation  $\leq$ , associée à l'application  $max$  est une relation d'ordre qui

correspond à l'ordre usuel  $\leq$ ,  $a \leq b \iff b = \max(a, b)$ .

La relation  $\leq$ , associée à l'application **min** est une relation d'ordre qui correspond à l'inverse de l'ordre usuel  $\geq$ ,  $a \leq b \iff b = \min(a, b)$ .

### 1.2.2.3. Dioïde matriciel :

Soit  $(D, \oplus, \otimes)$  un dioïde. On note  $D^{(p \times n)}$ , avec  $p, n \in \mathbb{N}^*$ , l'ensemble des matrices de dimension  $(p \times n)$  à coefficients dans  $D$ . La somme matricielle de deux matrices  $A, B \in D^{(p \times n)}$ , est une matrice notée  $A \oplus B$ , avec des coefficients qui sont donnés par l'expression suivante :

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij}.$$

Étant donnés les matrices  $A \in D^{(p \times n)}$ ,  $B \in D^{(n \times q)}$ , avec  $p, n, q \in \mathbb{N}^*$ . Le produit matriciel de ces deux matrices est une matrice notée  $A \otimes B$  ou  $A \cdot B$ , avec des coefficients qui sont donnés comme suit :

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^n (A_{ik} \otimes B_{kj}).$$

$A_{ij}$ ,  $B_{ij}$  désignent l'élément de la  $i^{me}$  rangée et de la  $j^{ime}$  colonne des matrices  $A$ ,  $B$  respectivement et  $C_{kj}$ , désigne l'élément de la  $k^{me}$  rangée et de la  $j^{ime}$  colonne de la matrices  $C$ .

**Exemple 4.** Soient  $A, B$  deux matrices dans l'algèbre  $(\max, +)$  tels que :

$$A = \begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix} \text{ et } B = \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}.$$

On a

$$A \oplus B = \begin{bmatrix} \max(3, 2) & \max(7, 0) \\ \max(2, 3) & \max(4, 1) \end{bmatrix} = \begin{bmatrix} 3 & 7 \\ 3 & 4 \end{bmatrix},$$

et

$$A \otimes B = \begin{bmatrix} \max(3+2, 7+3) & \max(3+0, 7+1) \\ \max(2+2, 4+3) & \max(2+0, 4+1) \end{bmatrix} = \begin{bmatrix} 10 & 8 \\ 7 & 5 \end{bmatrix}.$$

## 1.2.3 Résolution des équations dans les dioïdes

Certaines équations définies dans des dioïdes complets admettent des solutions particulières extrêmes, des solutions plus petite ou plus grande que toutes autre solution. Des résultats généraux relatifs à la résolution des équations sur des dioïdes complets sont

fournis dans [11, 12]. Nous rappelons ici simplement le résultat concernant la résolution de l'équation implicite  $x = a \otimes x \oplus b$  qu'on va utiliser par la suite.

**Définition 3.** (*Etoile de Kleene*)([11], [12])

Soit  $D$  un dioïde complet et  $a$  un élément de  $D$ , l'étoile de Kleene de  $a$ , noté  $a^*$ , est définie comme suit :  $a^* = \bigoplus_{k \in \mathbb{N}} a^k$  avec  $a^0 = e$ . De la même façon l'étoile de Kleene d'une matrice carrée  $A \in D^{n \times n}$ , notée  $A^*$ , est définie par :  $A^* = \bigoplus_{(i \in \mathbb{N})} A^i$  avec  $A^0 = I_n$  et  $I_n$  désigne la matrice identité.

**Théoreme 1.** [11] Dans un dioïde complet, la quantité  $(a^*b)$  est la plus petite solution de l'équation  $x = ax \oplus b$  et de l'inéquation  $x \geq ax \oplus b$ .

**Démonstaration.** On vérifie que  $x = a^*b$  est solution de  $x = ax \oplus b$ .

On a :

$$a(a^*b) \oplus b = a(e \oplus a \oplus a^2 \oplus \dots)b \oplus b = (e \oplus a \oplus a^2 \oplus a^3 \dots)b = a^*b. \quad \square$$

## 1.3 Réseaux de Petri et graphe d'événements temporisé

### 1.3.1 Réseaux de Petri (RDP)

Les réseaux de Petri (RDP) ont été introduits en 1962 par l'allemand Carl Adam Petri [44]. Ils constituent un outil puissant de description des systèmes à événements discrets. Ils permettent une modélisation graphique et mathématique des systèmes de production. Ils sont particulièrement adaptés à l'étude des processus complexes mettant en jeu des propriétés de synchronisation et de partage de ressources. Le lecteur intéressé par plus de détails sur les réseaux de Petri et les GETs trouvera une présentation générale et des références bibliographiques dans [42], [43], [20]. Pour l'étude des SEDs dans l'algèbre des dioïdes, une classe des réseaux de Petri appelée Graphe dévénements Temporisé (GET) est souvent utilisé comme outil de modélisation. Ces graphes peuvent être représentés par des équations linéaires dans cette algèbre. La démarche consiste en effet à modéliser le système étudié en premier lieu par un GET, puis à établir, à partir du graphe obtenu, les

équations linéaires dans le dioïde propre à l'analyse de ce système.

### 1.3.1.1. Définitions et notations

#### Définition 4. (*Réseau de Petri*)

Un réseau de Petri est un graphe biparti composé essentiellement de deux types d'objets : les places (représentées par des cercles) et les transitions (représentées par des barres). Des arcs orientés relient certaines places à certaines transitions, ou certaines transitions à certaines places (voir Fig.1.2). Des poids ( $W$ ) sont associés aux arcs, lorsque tous les arcs orientés ont un poids égal à 1, le réseau de Petri est dit ordinaire. S'il existe un arc orienté dont le poids est strictement supérieur à 1, le réseau de Petri est dit généralisé. Le nombre de transitions et de place est fini, et non nul. Chaque place peut comporter un ou plusieurs jetons représentés par des points modélisent l'état du système.

#### Définition 5. (*Réseau de Petri temporisé*)

Un réseau de Petri est dit temporisé si des temporisations sont associées aux places et/ou transitions.

Dans la suite de cette thèse, nous considérons des réseaux de Petri temporisé ordinaires où les temporisations sont associées aux places.

**Définition formelle** : un réseau de Petri est un 5-uple  $Q = (P, T, A, \mu_0, \tau)$  où :

- $P = \{p_1, p_2, \dots, p_n\}$  est un ensemble fini, non vide de places ,
- $T = \{t_1, t_2, \dots, t_m\}$  est un ensemble fini, non vide de transitions ,
- $A \subseteq (P \times T) \cup (T \times P)$  est l'ensemble fini des arcs qui connectent les places aux transition, et les transitions aux places.
- $\mu_0$  : est un vecteur d'entiers non négatifs dont la  $i$ ème composante indique le marquage initial de la place  $p_i$ .
- $\tau \in \mathbb{N}^P$  : définit des temporisations associées aux places. Une temporisation désigne la durée minimale de séjour d'un jeton dans la place  $p_i$ .

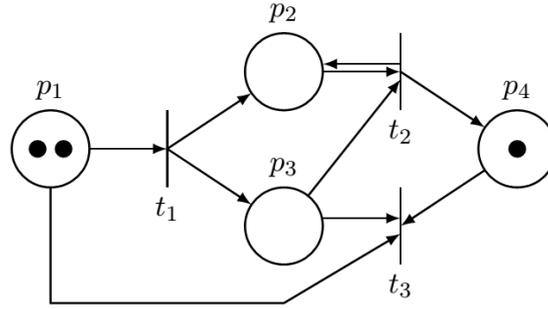


FIG. 1.2: Exemple d'un RDP

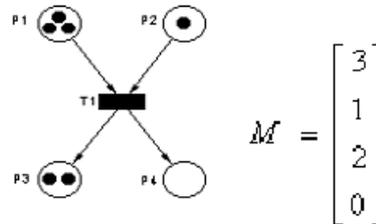


FIG. 1.3: Représentation de marquage d'un RDP

### • Marquage

Chaque place d'un RDP contient un nombre entier positif ou nul de jetons. Le marquage  $\mu$  définit l'état du système décrit par le réseau à un instant donné. C'est un vecteur colonne dont la dimension est le nombre de places dans le réseau. Le  $i$ ème élément du vecteur correspond au nombre de jetons contenus dans la place  $p_i$  (voir Fig.1.3).

### • Franchissement d'une transition

Une transition est franchissable lorsque toutes les places qui lui sont en amont (ou toutes les places d'entrée de la transition) contiennent au moins un jeton.

La transition  $t_i$  est franchissable à partir de l'état  $\mu$  à l'instant  $\tau_i$  et son franchissement conduit à l'état  $\mu'$ , on peut désigner cela par  $:\mu \xrightarrow{\tau_i} \mu'$ .

### Définition 6. (Graphe d'accessibilité)

Soit  $\eta = (R, \mu_0)$  un réseau de Petri. L'ensemble des marquages accessibles  $A(R, \mu_0)$  d'un

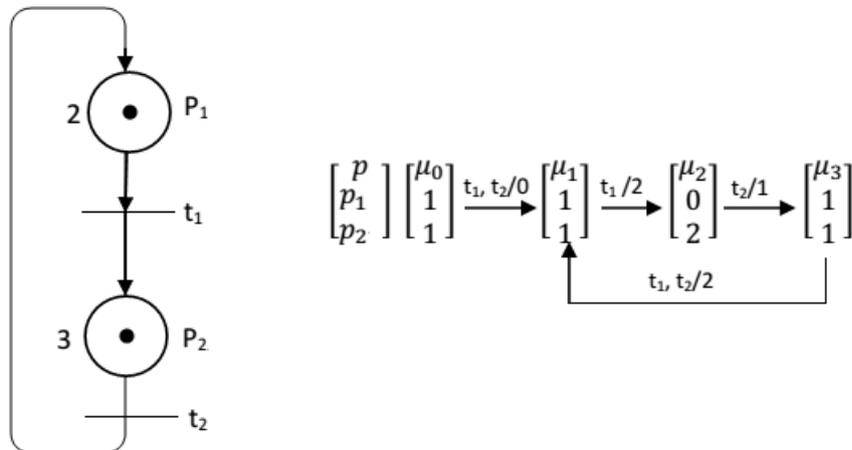


FIG. 1.4: Exemple d'un RDP avec son graphe d'accessibilité

réseau de Petri marqué est l'ensemble des marquages que l'on peut atteindre à partir du marquage initial  $\mu_0$  par une séquence de franchissement, après un temps  $\tau$ , c.à.d.

$$A(R, \mu_0) = \{\mu_i, \exists t_q, \mu_0 \xrightarrow{t_q/\tau} \mu_i\}$$

On peut, lorsque cet ensemble est fini, le représenter sous forme d'un graphe (Fig.2.11). Les sommets de ce graphe correspondent aux marquages accessibles de  $A(R, \mu_0)$ . Un arc orienté relie deux sommets  $\mu_i$  et  $\mu_j$  s'il existe une transition  $t_q$  franchissable permettant de passer d'un marquage à un autre après un temps  $\tau$  :  $\mu_i \xrightarrow{t_q/\tau} \mu_j$ .

### 1.3.1.2. Quelques propriétés des réseaux de Petri

**Définition 7. (Vivacité).**

Étant donné un réseau de Petri  $R$  et un marquage initial  $\mu_0$ , une transition  $t_i$  est dite vivante pour le réseau marqué  $(R, \mu_0)$  si pour tout marquage  $\mu$  accessible depuis  $\mu_0$ , il existe une suite de transitions  $T'$ , comportant au moins une fois la transition  $t_i$ , et telle que  $T'$  soit franchissable pour  $\mu$ .

Un réseau de Petri marqué  $(R, \mu_0)$  est dit vivant si toutes ses transitions sont vivantes.

**Définition 8. (RDP sans blocage).**

Un réseau de Petri  $R$  est dit sans blocage pour un marquage initial  $\mu_0$  si aucun marquage accessible n'est un blocage. autrement dit, un blocage est un marquage tel qu'aucune transition n'est validée.

**1.3.1.3. Différents structures des réseaux de Petri**

- 1- Un **graphe d'état** est un RDP telle que toute transition à une place d'entrée et une place de sortie.
- 2- Un **graphe d'événement** est un RDP telle que toutes place a exactement une transition d'entrée et une transition de sortie.
- 3- Un **RDP pur** s'il n'existe aucune transition telle que une des places d'entrée soit également une place de sortie.

**1.3.2 Graphe d'événements temporisé**

Dans notre travail, on s'intéresse à la classe des graphes d'événements temporisés qui est une sous classe des réseaux de Petri (notés GET) et qui permet de modéliser les systèmes où interviennent des phénomènes de synchronisation et de délai (les phénomènes de conflits étant supposés résolus au préalable). Les graphes d'événements temporisés admettent une représentation linéaire sur une structure algébrique de dioïde, les résultats sont bien connus [15],[40] et [19]. Dans cette section, nous représentons quelques définitions et notations sur les GETs et la représentation de la dynamique de ces dernières dans l'algèbre Max-Plus et l'algèbre Min-Plus.

**Définition 9. Graphe d'événements temporisés**

Un graphe d'événements  $G$  est un réseau de Petri ordinaire où chaque place possède exactement une transition d'entrée et une transition de sortie. Un graphe d'événement est dit temporisé si des temporisations sont associées aux places et/ou transitions.

Dans la suite de ce mémoire, nous considérons des graphes d'événement P-temporisés où les temporisations sont associées aux places. Pour chaque couple de transitions  $t_i, t_j \in T$ , tel que  $T$  est l'ensemble des transitions du graphe considérée. On note par  $p_{ij}$ , la place

qui relie la transition  $t_j$  à  $t_i$ . Si cette place existe, la temporisation correspondante est notée  $\tau_{ij}$  et son marquage est noté  $m_{ij}$ .

### 1.3.2.1. Propriétés des graphes d'événements

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

- **Chemin**

Un chemin entre deux transitions  $t_i$  et  $t_j$  d'un réseau de Petri, est une succession de place et de transitions relié par des arcs, tel que la transition initiale du premier arc est  $t_i$ , le sommet terminal du dernier arc est  $t_j$ .

Dans la suite de ce travail on note par  $\alpha$  le chemin qui relie la transition de commande  $t_u$  à la transition  $t_j$ , et par  $\rho$  le chemin qui relie la transition  $t_j$  à la transition  $t_i$ .

- **circuit**

Un circuit est un chemin où la transition terminal est aussi la transition initial.

- **Chemin élémentaire**

Un chemin élémentaire est chemin où aucun arc n'est répété.

**Proposition 1.** [39] *Dans un graphe d'événements, le nombre de jetons d'un circuit élémentaire est constant.*

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

### 1.3.2.2. Représentation de la dynamique ([11], [12])

Les graphes d'événements temporisés sont utilisés comme outil de modélisation intermédiaire. Si cette modélisation graphique constitue une première étape, la seconde étape est la mise en équation du modèle graphique, c'est-à-dire la définition d'une représentation analytique du système. Le comportement d'un graphe d'événements temporisé peut

s'exprimer sous forme d'équations linéaires dans les dioïdes  $\bar{\mathbb{R}}_{min}$  ou  $\bar{\mathbb{R}}_{max}$ . Ces équations représentent l'évolution du graphe dans les domaines temporel et événementiel. Si on considère  $\tau_{ij}$  la temporisation correspondante à la place  $p_{ij}$  (c'est-à-dire la place qui relie la transition  $t_j$  à la transition  $t_i$ ) avec un marquage initial noté  $m_{ij}$ . La plus grande temporisation du graphe d'événements considéré est notée  $\tau^{max}$ , définie par :  $\tau^{max} = \max(\tau_{ij})$ . Le plus grand marquage initial des places du graphe d'événements considérés est défini par :  $b = \max(m_{ij})$ . D'une manière générale, le comportement d'un graphe d'événements temporisés est représenté sous l'une des deux formes définies dans le paragraphe qui suit.

**a) Fonctions compteurs, domaine temporel :** La mise en équation de GET peut se faire dans le domaine temporel, où le système est écrit par des fonctions dépendantes du temps  $t$ . Dans ce cas, on s'intéresse au nombre d'activation des transitions jusqu'à l'instant considéré. En effet, cette représentation consiste à associer à chaque transition une fonction  $\theta(t) \in \bar{\mathbb{R}}_{min}^n$  cette fonction est appelée compteur.

Les compteurs correspondants aux transitions sources (transitions d'entrées) sont les composantes du vecteur  $u(t) \in \bar{\mathbb{R}}_{min}^m$ .

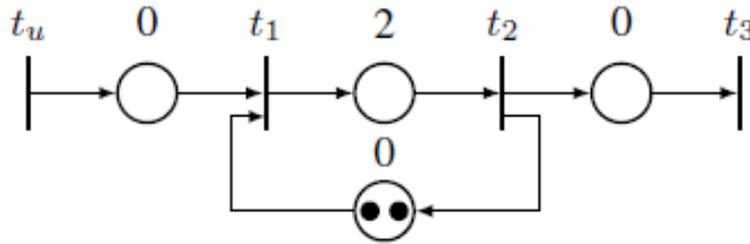


FIG. 1.5: Exemple d'un GET

Le comportement du graphe d'événements temporisé de la Fig.1.5 est modélisé par les inéquations suivantes :

$$\begin{cases} \theta_1(t) \leq \min(0 + u(t), 2 + \theta_2(t)), \\ \theta_2(t) \leq 0 + \theta_1(t - 2), \\ \theta_3(t) \leq 0 + \theta_2(t). \end{cases}$$

En utilisant des opérateurs algébrique de Min-Plus ( $\oplus$ ) pour l'opérateur min et ( $\odot$ ) pour

la somme classique  $+$ , ces équations peuvent être réécrites comme suit :

$$\begin{cases} \theta_1(t) \leq e \cdot u(t) \oplus 2 \cdot \theta_2(t), \\ \theta_2(t) \leq e \cdot \theta_1(t-2), \\ \theta_3(t) \leq e \cdot \theta_2(t). \end{cases}$$

Ces équations peuvent s'écrire sur la forme suivante :

$$\boldsymbol{\theta}(t) \geq A_0 \cdot \boldsymbol{\theta}(t) \oplus A_2 \cdot \boldsymbol{\theta}(t-2) \oplus \mathbf{B}_0 \cdot u(t),$$

avec :

$$\boldsymbol{\theta}(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \end{bmatrix}, A_0 = \begin{bmatrix} \varepsilon & 2 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}, A_2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \mathbf{B}_0 = \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \end{bmatrix}. \text{ En général, le}$$

comportement d'un graphe d'événements temporisé est représenté par l'inéquation suivante :

$$\theta(t) \leq \oplus_{\tau=0}^{\tau^{max}} (A_{\tau} \cdot \theta(t-\tau) \oplus B_{\tau} \cdot u(t-\tau)).$$

où  $A_{\tau} \in \bar{\mathbb{R}}_{min}^{n \times n}$  est la matrice dont le terme  $A_{\tau,ij}$  est égal à  $m_{ij}$ , qui correspond au nombre de jetons initiaux dans la place  $p_{ij}$  si cette place existe et  $\varepsilon$  sinon. Les termes des matrices  $B_{\tau}(k) \in \bar{\mathbb{R}}_{min}^{n \times m}$  correspondent aux marquages initiaux des places de sortie des transitions sources.

En général, on s'intéresse à l'évolution au plus tôt des graphes d'événements temporisés, c'est-à-dire qu'une transition est franchie dès qu'elle est franchissable. Cette évolution correspond à la solution maximale de l'inéquation précédente dans  $\bar{\mathbb{R}}_{min}$ . Cette solution satisfait l'équation linéaire suivante :

$$\theta(t) = \oplus_{\tau=0}^{\tau^{max}} (A_{\tau} \cdot \theta(t-\tau) \oplus B_{\tau} \cdot u(t-\tau)). \quad (1.1)$$

L'équation (1.1) est implicite en général. Elle est souvent remplacée par la solution explicite suivante :

$$\theta(t) = \oplus_{\tau=1}^{\tau^{max}} (A_0^* \cdot A_{\tau} \cdot \theta(t-\tau) \oplus A_0^* \cdot B_{\tau} \cdot u(t-\tau)). \quad (1.2)$$

Où :  $A_0^*$  est l'étoile de Kleene de la matrice  $A_0$ .

**Equation d'état dans  $\bar{\mathbb{R}}_{min}$ .**

Par analogie avec la théorie des systèmes linéaires classiques, l'équation explicite (1.2) peut être transformée en une forme d'état. Pour obtenir un modèle d'état, nous décomposons toutes les places dont la temporisation  $\tau > 1$  en  $\tau$  places temporisées à 1. Nous ajoutons donc  $(\tau - 1)$  transitions intermédiaires. On associe des compteurs à ces transitions intermédiaires, au nombre de  $n'$  qui sont les composantes du vecteur  $\bar{\theta}(t) \in \bar{\mathbb{R}}_{min}^{n'}$ . Le vecteur d'état résultant, noté  $x(t)$ , appartient à  $\bar{\mathbb{R}}_{min}^N$ , avec  $N = n + n'$ , et est défini par :

$$x(t) = \begin{bmatrix} \theta(t) \\ \bar{\theta}(t) \end{bmatrix}$$

Le comportement dynamique du graphe d'événements temporisé étendu est décrit par une équation de la forme :

$$x(t) = \hat{A}_0 \cdot x(t) \oplus \hat{A}_1 \cdot x(t-1) \oplus \hat{B}_0 \cdot u(t), \quad (1.3)$$

Qui peut s'écrire sous la forme explicite suivante :

$$x(t) = A \cdot x(t-1) \oplus B \cdot u(t), \quad (1.4)$$

avec

$$A = \hat{A}_0^* \cdot \hat{A}_1 \text{ et } B = \hat{A}_0^* \cdot \hat{B}_0$$

**Propriété 1.** [4] *Pour un graphe d'événements temporisé avec des temporisations commensurables, l'équation d'état (1.4) est équivalente à la formulation suivante :*

$$x(t) = A^\tau \cdot x(t-\tau) \oplus \left[ \bigoplus_{k=0}^{\tau-1} A^k \cdot B \cdot u(t-k) \right] \quad (1.5)$$

Pour tout  $\tau$  entier tel que  $\tau \geq 1$ .

**Démonstration.** Nous allons démontrer cette propriété par récurrence. Etant donné un graphe d'événement temporisé dont le comportement est décrit par l'équation d'état (1.4). Il est clair que la propriété est vérifiée pour  $\tau = 1$ . Supposons qu'elle est aussi vérifiée pour  $\tau = K$ , c'est-à-dire :

$$x(t) = A^K \cdot x(t-K) \oplus \left[ \bigoplus_{k=0}^{K-1} A^k \cdot B \cdot u(t-K) \right] \quad (1.6)$$

Du fait que l'hypothèse que l'équation d'état est satisfaite, nous avons

$$x(t - K) = A.x(t - K - 1) \oplus B.u(t - K). \quad (1.7)$$

En remplaçant  $x(t - K)$  par son expression dans (1.6), nous obtenons alors

$$x(t) = A^{K+1} \cdot x(t - (K + 1)) \oplus \left[ \bigoplus_{k=0}^{K-1} A^k \cdot B \cdot u(t - K) \right]. \quad (1.8)$$

□

### b) Fonctions dateurs, domaine événementiel :

Pour la représentation en dateurs, on s'intéresse aux dates d'activation des transitions du GET. Dans ce cas, on associe à chaque transition une fonction  $\theta(k) \in \bar{\mathbb{R}}_{max}^n$  cette fonction est appelée dateur. Les dateurs correspondants aux transitions sources sont les composantes du vecteur  $u(k) \in \bar{\mathbb{R}}_{max}^m$ . La dynamique d'un graphe d'événements temporisé est représentée par l'inéquation suivante :

$$\theta(k) \geq \bigoplus_{l=0}^{m_{max}} (A_l \cdot \theta(k - l) \cdot B_l \cdot u(k - l)).$$

où  $A_l \in \bar{\mathbb{R}}_{max}^{n \times n}$  est la matrice dont le terme  $A_{l,ij}$  est égal à  $\tau_{ij}$ , qui correspond à la temporisation de la place  $p_{ij}$  marquée à  $l$ . Si cette place n'existe pas, le terme  $A_{l,ij}$  est égale à  $\varepsilon$ . Les termes des matrices  $B_l \in \bar{\mathbb{R}}_{max}^{n \times m}$  correspondent aux temporisations des places de sortie des transitions source. En général, on s'intéresse à l'évolution au plus tôt des graphes d'événements temporisés, c'est à dire qu'une transition est franchie dès qu'elle est franchissable. Cette évolution correspond à la solution minimale de l'inéquation précédente dans  $\bar{\mathbb{R}}_{max}$ . Cette solution satisfait l'équation linéaire suivante :

$$\theta(k) = \bigoplus_{l=0}^{m_{max}} (A_l \cdot \theta(k - l) \cdot B_l \cdot u(k - l)). \quad (1.9)$$

L'équation 1.9 est implicite en général. Elle est souvent remplacée par sa solution explicite suivante :

$$\theta(k) = \bigoplus_{l=1}^{m_{max}} (A_0^* \cdot A_l \theta(k - l) \cdot A_0^* \cdot B_l u(k - l)). \quad (1.10)$$

où  $A_0^*$  est l'étoile de Kleene de la matrice  $A_0$ .

### Equation d'état dans $\bar{\mathbb{R}}_{max}$ .

Par analogie avec la théorie des systèmes linéaires classiques, l'équation explicite 1.10 peut être transformée en une forme d'état. Pour obtenir un modèle d'état, nous décomposons toutes les places dont le marquage  $m > 1$  en  $m$  places marquées à 1, et donc, nous ajoutons  $(m - 1)$  transitions intermédiaires. On associe des dateurs à ces transitions intermédiaires au nombre de  $n'$  qui sont les composantes d'un vecteur  $\theta(k) \in \bar{\mathbb{R}}_{max}^{n'}$ . Le vecteur d'état résultant, noté  $x(k)$ , appartient à  $\bar{\mathbb{R}}_{max}^N$ , avec  $N = n + n'$ , et est défini par :

$$x(k) = \begin{bmatrix} \theta(k) \\ \bar{\theta}(k) \end{bmatrix},$$

Dans notre cas, nous considérons des graphes d'événements dont le marquage des place qui ont des transitions source en amont est nul. Le comportement dynamique du graphe d'événements temporisé étendu est décrit par une équation de la forme :

$$x(k) = \hat{A}_0 \cdot x(k) \oplus \hat{A}_1 \cdot x(k - 1) \oplus \hat{B}_0 \cdot u(k),$$

qui peut s'écrire sous la forme explicite suivante :

$$x(k) = A \cdot x(k - 1) \oplus B \cdot u(k), \quad (1.11)$$

Avec  $A = \hat{A}_0^* \cdot \hat{A}_1$  et  $B = \hat{A}_0^* \cdot \hat{B}_0$ .

**Propriété 2.** *on considère un graphe d'événements temporisé dont le comportement est représenté par l'équation (1.9). Cette représentation d'état est équivalente à la formulation suivante :*

$$x(k) = A^\phi \cdot x(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B) \cdot u(k - k') \right], \quad (1.12)$$

Pour tout  $\phi$  entier tel que  $\phi \geq 1$ .

**Exemple 5.** *Nous considérons l'exemple de graphe d'événements temporisé de la Fig.1.6. Ce graphe représente le fonctionnement d'un système qui se compose essentiellement d'un four et d'un stock. Le four est modélisé par la place  $p_{21}$  qui peut contenir jusqu'à deux*

pièces, le temps de cuisson est 1 unité du temps. La place  $p_{32}$  modélise le stock qui représente aussi la zone de refroidissement, ce stock peut contenir une seule pièce à la fois, le temps de refroidissement est de 2 unités du temps.

- Fonctions dateurs

Nous considérons le fonctionnement au plutôt, c'est-à-dire que toute transition validée est immédiatement franchie. Les fonctions dateurs vérifient alors les équations suivantes :

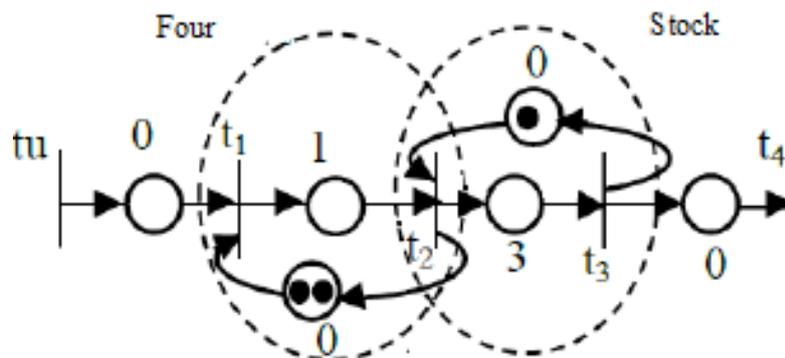


FIG. 1.6: modèle GET d'un système composé d'un four et d'un stock

$$\begin{cases} \theta_1(k) = \max(0 + u(k), 0 + \theta_2(k - 2)), \\ \theta_2(k) = \max(1 + \theta_1(k), 0 + x_3(k - 1)), \\ \theta_3(k) = 3 + \theta_2(k), \\ \theta_4(k) = 0 + \theta_3(k). \end{cases}$$

Ces équations deviennent linéaires dans l'algèbre Max-Plus,

$$\begin{cases} \theta_1(k) = e \cdot u(k) \oplus e \cdot \theta_2(k - 2), \\ \theta_2(k) = 1 \cdot \theta_1(k) \oplus e \cdot x_3(k - 1), \\ \theta_3(k) = 3 \cdot \theta_2(k), \\ \theta_4(k) = e \cdot \theta_3(k). \end{cases}$$

Ce qui donne :

$$\theta(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix} \cdot \theta(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot \theta(k-1) \oplus \begin{bmatrix} \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot \theta(k-2) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \cdot u(k).$$

Pour avoir la forme d'état, nous avons étendu le graphe d'événements temporisé initial pour avoir un nouveau graphe équivalent avec des marquages égaux à 1 ou 0. Nous obtenons alors le graphe de la Fig.1.7.

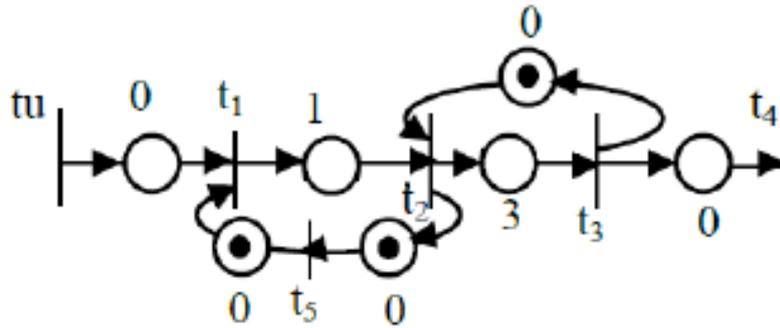


FIG. 1.7: Graphe étendu de système montré dans la Fig.1.6

$$x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & e & \varepsilon & 1 \\ \varepsilon & \varepsilon & 3 & \varepsilon & 4 \\ \varepsilon & \varepsilon & 3 & \varepsilon & 4 \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot x(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 4 \\ 4 \\ \varepsilon \end{bmatrix} \cdot u(k). \quad (1.13)$$

où  $x_i(k)$  est la fonction dateur de la transition  $t_i$  pour  $i = 1$  à 5, et  $u(k)$  correspond à la fonction dateur de la transition d'entrée  $tu$ .

#### • Fonctions compteurs

Nous associons maintenant pour chaque transition de graphe d'événements de la Fig.1.6 une fonction compteur  $\theta_i(t)$ . Pour un franchissement au plus tôt, les fonctions compteurs

vérifient les équations suivantes :

$$\begin{cases} \theta_1(t) = \min(0 + u(t), 2 + \theta_2(t)), \\ \theta_2(t) = \min(1 + \theta_3(t), 0 + x_1(t - 1)), \\ \theta_3(t) = 0 + \theta_2(t - 3), \\ \theta_4(t) = 0 + \theta_3(t). \end{cases}$$

Dans l'algèbre  $(\min,+)$  , ces équations sont écrites comme suit :

$$\begin{cases} \theta_1(t) = 2 \cdot \theta_2(t) \oplus e \cdot u(t), \\ \theta_2(t) = 1 \cdot \theta_3(t) \oplus e \cdot \theta_1(t - 1), \\ \theta_3(t) = e \cdot \theta_2(t - 3), \\ \theta_4(t) = e \cdot \theta_3(t). \end{cases}$$

Ce qui donne :

$$\theta(t) = \begin{bmatrix} \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix} \cdot \theta(t) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot \theta(t-1) \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \cdot \theta(t-3) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \cdot u(t).$$

Pour avoir la forme d'état, nous avons étendu le graphe d'événements temporisés initial pour avoir un nouveau graphe équivalent avec des temporisations égales à 1 ou 0. Nous obtenons alors le graphe de Fig.1.8.

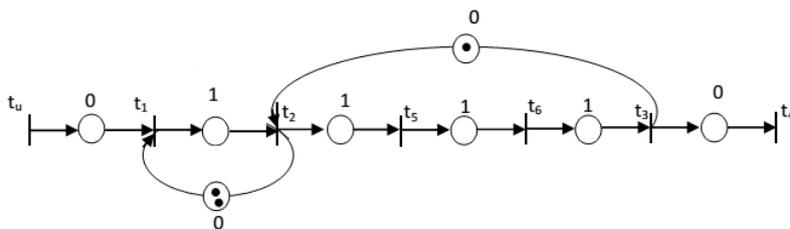


FIG. 1.8: Graphe étendu de système montré dans la Fig.1.6

Le modèle d'état obtenu est donné par :

$$x(t) = \begin{bmatrix} 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix} \cdot x(k-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \cdot u(k).$$

## 1.4 Conclusion

Dans ce premier chapitre, nous avons présenté les outils mathématiques utilisés dans nos travaux. Nous avons ainsi donné quelques définitions et propriétés relatives aux dioïdes. Nous avons introduit les graphes d'événements temporisés après un petit rappel sur les réseaux de Petri. Par la suite, nous avons montré comment, à partir d'un GET, obtenir des modèles linéaires dans l'algèbre Max-plus et Min-plus.

# Chapitre 2

## Commande Max-Plus des systèmes à événements discrets sous contraintes de temps

### 2.1 Introduction

Commander un système à événements discret consiste à synthétiser des lois de commande qui agissent sur le système considéré afin de garantir des spécifications imposées par le cahier des charges de ce dernier. Ainsi, il apparaît que la synthèse de lois de commande représente un concept fondamental dans la structure et la conception de ces systèmes. Cette notion a attiré l'attention de plusieurs chercheurs de la communauté des systèmes à événements discrets. Plusieurs approches ont été développées pour synthétiser un contrôleur (superviseur) afin de garantir un fonctionnement souhaité pour cette catégorie de systèmes [46], [53],[32].

Par ailleurs, il est à noter que le temps est une composante primordiale dans les systèmes à événements discrets i.e. les opérations doivent avoir lieu dans des délais bien précis (définis selon l'application), et avant une date fixée à l'avance, afin de garantir le bon fonctionnement du système. Le non-respect des délais ou des contraintes temporelles peut conduire à des états non souhaitables et parfois même à des conséquences graves. Il est

donc, très important de disposer de techniques d'analyse et de contrôle afin de garantir les spécifications temporelles imposées par le cahier des charges. Plusieurs approches de commande basées sur les différents outils de modélisation et d'analyse sont développées dans le but de garantir des spécifications temporelles imposées sur cette catégorie de systèmes. A titre d'exemple, l'approche proposée dans [14] est basée sur les automates temporisés et la discrétisation du temps. Deux types d'automates sont utilisés, le premier pour la modélisation des états et les transitions du système, et le second pour la modélisation des contraintes temporelle. Cette approche donne une bonne solution de commande par supervision des systèmes à événements discrets temporisés. Comme toutes méthodes basées sur l'utilisation des automates, son inconvénient majeur réside dans le problème de l'explosion combinatoire du nombre d'états. Afin d'éviter le problème d'explosion combinatoire, plusieurs approches basées sur les réseaux de Petri ont été proposées. Dans ([13]), une approche basée sur les réseaux de Petri temporisés a été présentée. L'auteur a développé une nouvelle technique énumérative pour l'analyse des systèmes à temps critique. Il est à noter que ces dernières années, plusieurs approches formelles ont été développées pour la commande des SEDs sous contrainte de temps en utilisant de l'algèbre des dioïdes [2], [8], [15], [28], [34], [38]. Les auteurs de [15], [34], s'intéressent plus précisément à des problèmes de rejet de perturbations et de poursuite de modèles de références. La contrainte temporelle dans ces cas est considérée comme une condition supplémentaire à satisfaire. Par contre, les auteurs dans [2], [8], ont abordé le problème de la commande des systèmes à événements discrets sous contraintes temporelles strictes. Ils ont calculé des feedbacks causaux pour satisfaire des contraintes de temps. Leur méthode est basée sur la modélisation du système considéré par un graphe d'événements temporisé. L'algèbre Max-Plus a été utilisé pour décrire le comportement du graphe par des équations linéaires. Toutes les transitions du graphe considéré ont été considérées comme observables, mais non toutes commandables (que les transitions d'entrées sont considérées commandables). Une extension de cette méthode est représentée dans [31], où les auteurs ont abordé le problème de contraintes du temps imposées à un SED, les auteurs ont considéré que les transitions du graphe ne sont pas toutes observables (que les transitions de sortie sont

observables). L'objectif des auteurs est de calculer des lois de commande en fonction des transitions de sorties. Ces dernières doivent satisfaire les contraintes de temps imposées sur le GET qui modélise le système considéré.

Une autre méthode basée sur l'algèbre Max-Plus est présentée dans [28], les auteurs ont considéré que les systèmes sont soumis à des contraintes temporelles et à des contraintes de marquage imposées sur certaines places ou chemins du graphe. Le contrôleur synthétisé est un sous graphe qui se compose de places et de transitions connectées au GET du système.

Les méthodes que nous avons citées précédemment sont limitées au cas où la contrainte temporelle est imposée sur une place de GET. Dans ce chapitre, nous allons traiter le cas général où les contraintes sont imposées sur certains chemins d'un GET, sachant qu'une place dans un GET est un chemin de longueur 1. Notre travail, consiste à synthétiser un contrôleur qui garantit des contraintes temporelles imposées sur certains chemins d'un graphe d'événements temporisé. Ce type de contraintes est très fréquent dans les architectures d'automatisation en réseaux (AAR) [49], dans les systèmes de transports [28]...etc. Ce chapitre est structuré comme suite : La section 2.2 est consacrée à la présentation de notre problématique et notre motivation. La commande Max-Plus des GET sous contraintes de temps est représentée dans la section 2.3. Nous considérons dans un premier temps le cas de la commande d'un graphe d'événements temporisé avec une seule transition de commande. Par la suite nous généralisons notre approche pour le cas de la commande d'un GET avec plusieurs transitions de commande. Pour les deux cas, nous considérons tout d'abord le cas où le système est soumis à une seule contrainte temporelle. Ensuite, nous étendrons l'approche au cas où le graphe est soumis à plusieurs contraintes temporelles différentes. Dans la dernière section, nous appliquons les résultats de l'approche que nous avons proposée à une Architecture d'Automatisation en Réseau (AAR). Nous décrivons tout d'abord les différentes parties de ces systèmes et leurs principes de fonctionnement, puis nous passerons à la commande d'une AAR soumise à des contraintes temporelles imposées sur le temps de réponse.

## 2.2 Motivations et problématique

Dans certaines applications industrielles le temps est contraint. Ces contraintes peuvent prendre diverses formes (échéance, intervalle de temps, durée de validité, etc.) et s'appliquer à des objets variés. Dans la littérature, de nombreuses approches sont proposées pour la synthèse de contrôleurs afin de satisfaire les contraintes temporelles imposées aux systèmes à événements discrets temporisés. Nous pouvons citer à titre d'exemple [2],[8]. Les auteurs traitent le problème de la contrainte temporelle imposée aux places d'un graphe d'événements temporisé. Ces approches sont basées sur l'utilisation des graphes d'événements temporisés pour la modélisation des systèmes considérés. Et par la suite utiliser l'algèbre Max-Plus pour décrire le comportement des GETs via des équations linéaire dans l'algèbre Max-Plus. Dans le cas pratique, l'objectif des auteurs est de garantir la durée de l'exécution d'une tâche ou plusieurs tâches indépendantes.

Dans les systèmes industriels en générale, les taches sont dépendantes les unes des autres. Chaque tâche est reliée à la tache qui la précède, il existe alors des contraintes temporelles imposées sur plusieurs taches dépendantes. Dans les architectures d'automatisation en réseau, par exemple, le temps de réponse est représenté par une succession de tâches de traitement et de communication dépendantes. Ces systèmes généralement sont soumis à une contrainte temporelle imposée sur leurs temps de réponse. Cette contrainte à satisfaire n'est pas imposée sur une place du graphe modélisant ce système, mais sur une succession de place du GET considéré. Les méthodes qu'on a citées précédemment sont limitées et ne permettent pas de considérer des contraintes sur des chemins. Dans ce chapitre, on s'intéresse plus précisément aux contraintes temporelles imposées sur certains chemins d'un graphe d'événements temporisé modélisant une succession de taches dépendantes. Nous cherchons à synthétiser un contrôleur sous la forme de  $u(k) = F \cdot x(k-1)$  afin de garantir la satisfaction de contraintes temporelles imposées sur certains chemins d'un GET. Notre méthode est basée sur l'utilisation de GETs pour la modélisation de système considéré et sur l'algèbre Max-plus pour exprimer le comportement de GETs par des équations linéaires. Les contraintes à satisfaire sont exprimées par des inéquations linéaires dans cette algèbre. L'approche que nous proposons dans ce chapitre est

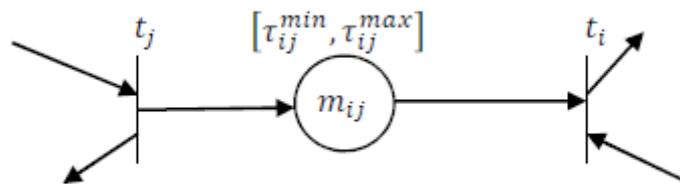


FIG. 2.1: Contrainte temporelle imposée sur une place

une extension de l'approche proposée dans [2]. Nous généralisons la méthode proposée dans [2] pour le cas où les contraintes temporelles sont imposées sur les chemins d'un graphe d'événements temporisé au lieu des contraintes imposées sur des places. Avant de présenter notre contribution sur la commande des graphes d'événements temporisés sous contraintes de temps, nous commençons tout d'abord par présenter un petit résumé sur la méthode proposée dans [2].

### 2.2.1 Problème de commande des GETs sous contraintes temporelles imposées sur des places [2]

Plusieurs méthodes ont été proposées pour la commande des GETs sous contraintes temporelles imposées sur des places de ce dernier. Nous pouvons citer à titre d'exemple les méthodes proposées dans [34], [2], ces méthodes sont basées sur l'utilisation des GETs et de l'algèbre Max-Plus.

L'objectif des auteurs de l'approche proposée dans ([2]) est de synthétiser un contrôleur sous la forme de  $u(k) = F.x(k-1)$  afin de garantir les contraintes temporelles imposées sur des places d'un GET.

Dans les graphes d'événements temporisés, la temporisation de chaque place correspond à la durée minimale de séjour des jetons dans la place. La durée maximale apparaît comme une contrainte supplémentaire qui devrait être vérifiée. Alors des intervalles de temps sont associés pour chaque place comportant une contrainte temporelle.

Soit  $p_{ij}$  une place soumise à une contrainte de temps de séjour, un intervalle de temps  $[\tau_{ij}^{min}, \tau_{ij}^{max}]$  est associé à cette place, comme illustrée à la Fig.2.1.

En associant à chaque transition du graphe dévénements temporisé un dateur. Les dateurs associés aux transitions  $t_i$  et  $t_j$  sont notés respectivement par :  $x_i(k)$  et  $x_j(k)$ . La contrainte temporelle imposée sur la place  $p_{ij}$  est exprimée par l'inéquation Max-plus linéaire suivante :

$$x_i(k) \leq \tau_{ij}^{max} \cdot x_j(k - m_{ij}) \quad (2.1)$$

La condition (2.1) impose aux jetons de passer au maximum  $\tau_{ij}^{max}$  unités de temps dans la place  $p_{ij}$ . Les auteurs ont ainsi synthétisé des lois de commande causales  $u(k)$  qui permettent de garantir le respect de cette contrainte temporelle. La méthode développée est basée sur la résolution d'un système d'inéquation dans l'algèbre Max-Plus. Les auteurs supposent qu'il existe au moins un chemin  $\alpha$  qui relie la transition source à la transition  $t_j$  ( $t_j$  étant la transition d'entrée de la place sous contrainte  $p_{ij}$ ), le marquage de ce chemin est noté par  $m_\alpha$ .

Les auteurs de [2], ont défini deux conditions suffisantes et nécessaires pour l'existence des lois de commande qui garantissent la contrainte temporelle imposée sur la place  $p_{ij}$ . La formule du contrôleur  $u(k)$  est donné par une formule explicite en fonction des transitions de GET qui modélise le système considéré. Les auteurs ont considéré tout d'abord le cas de commande d'un GET avec une seule contrainte temporelle, puis ils ont généralisé leur approche pour le cas d'un GET avec plusieurs transitions sources.

Le Théorème 2 est dédié pour le cas simple, commande d'un GET avec une seule transition de commande et soumis à une seule contrainte de temps. Les résultats obtenus sont résumés dans le Théorème suivant :

**Théoreme 2.** [2]. *L'équation*

$u(k) \geq \oplus_{r=1}^N [(A_{ir} - B_j - \tau_{ij}^{max}) \cdot x_r(k - 1)]$ , *définit des lois de commande causales qui garantissent le respect de la contrainte  $x_i(k) \leq \tau_{ij}^{max} \cdot x_j(k - m_{ij})$  si les deux conditions suivantes :*

$$(A^\phi)_{ir} \geq \tau_{ij}^{max} \cdot (nA^{m_\alpha} \cdot B)_j \text{ avec } r = 1 \text{ à } N \text{ et } \phi = m_\alpha + m_{ij} + 1 .$$

$$(A^{k'} \cdot B)_i \leq \tau_{ij}^{max} \cdot (A^{m_\alpha} \cdot B)_j \text{ avec } 0 \leq k' \leq \phi - 1$$

*sont satisfaites, et si l'hypothèse ( $m_\alpha = m_{ij} = 0$ ) est bien respectée.*

### 2.2.2 Contrainte temporelle imposée sur un chemin d'un GET

Au lieu de considérer une contrainte temporelle imposée sur des places d'un graphe d'événements temporisé on peut chercher à limiter le délai de séjour d'un jeton dans certains chemins d'un GET. Si ce graphe modélise un système de production, ce type de contrainte a pour but de limiter le temps d'exécution d'une succession de tâches dépendantes. Soit  $\rho$  un chemin d'un GET schématisé sur la Fig2.2.

Le nombre de jetons initiaux contenus dans ce chemin est  $m_\rho$ , l'intervalle de temps as-

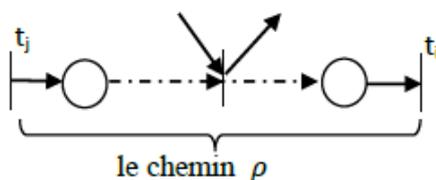


FIG. 2.2: Schéma d'un chemin d'un GET

socié à ce chemin est  $[\tau_\rho, \tau_\rho^{max}]$ ,  $\tau_\rho^{max}$  représente la borne maximale de temps à ne pas dépasser. On associe aux transitions  $t_j$  et  $t_i$  deux dateurs notés respectivement  $x_j(k)$ ,  $x_i(k)$ . La contrainte temporelle imposée sur le chemin  $\rho$  peut être décrite en fonction de ces dateurs par la relation suivante :

$$x_i(k) - x_j(k - m_\rho) \leq \tau_\rho^{max} \quad (2.2)$$

Daprès la définition des dateurs  $(x_i(k) - x_j(k - m_\rho))$  représente le temps de séjour du kème jeton dans le chemin  $\rho$ .

L'inéquation (2.14) peut être écrite comme suite :

$$x_i(k) \leq \tau_\rho^{max} \cdot x_j(k - m_\rho) \quad (2.3)$$

La contrainte (2.3) exprime la durée maximale à ne pas dépasser par un jeton entrant dans le chemin  $\rho$ . Notre travail consiste à synthétiser un contrôleur afin de respecter cette contrainte temporelle.

## 2.3 Commande temporelle des systèmes Max-Plus linéaires

Dans cette section, nous étendons la méthode proposée dans [2] pour le cas de contraintes temporelles imposées sur des chemins d'un graphe d'événements temporisé [49]. Dans un premier temps nous considérons le cas de commande des graphes d'événements temporisés à une seule transition d'entrée : ce dernier est soumis à une seule contrainte temporelle où plusieurs contraintes temporelles. Ensuite, nous généralisons notre étude pour le cas de commande de GET à plusieurs transitions d'entrées.

Dans la suite de ce chapitre nous considérons les hypothèses suivantes :

**Hypothèse 1.** *Il existe au moins un chemin qui relie la transition de commande à la transition d'entrée du chemin sous contrainte.*

**Hypothèse 2.** *Chaque transition de graphe d'événements temporisé est franchie dès qu'elle est franchissable (franchement au plus tôt).*

### 2.3.1 Commande monovariante

Dans cette section, nous considérons un système à événements discret modélisé par un graphe d'événements temporisé soumis à des contraintes temporelles. Ces contraintes sont imposées sur certains chemins de ce GET. Le comportement de ce graphe est représenté par une équation d'état (1.11), avec une seule entrée de commande qui est un scalaire  $u(k) \in \bar{\mathbb{R}}_{max}^{N \times 1}$ . Notre objectif est de calculer une loi de commande sous la forme de  $u(k) = F \cdot x(k-1)$  où  $F$  est un vecteur ligne avec  $N$  composantes. On note les composantes du vecteur  $F$  par  $F_r$  pour ( $r = 1$  à  $N$ ). La loi de commande que nous cherchons est fonction des dateurs des transitions internes de graphe considéré. La composante  $F_r$  représente la temporisation de la place qui relie la transition interne  $t_r$  à la transition de commande  $t_u$ , dans le cas où y a pas de liaison entre une transition  $t_r$  et la transition de commande  $t_u$ ,  $F_r$  prend la valeur  $\varepsilon$ .

### 2.3.1.1. Une seule contrainte

Nous considérons un graphe d'événements temporisé  $G$  comportant une seule contrainte temporelle. Cette contrainte est imposée sur le chemin  $\rho$  qui relie la transition  $t_j$  à la transition  $t_i$  du graphe  $G$ . La temporisation associée à ce chemin est notée  $\tau_\rho$ , cette temporisation représente la somme des temporisations associées aux places de ce chemin. Les transitions du graphe  $G$  ne sont pas toutes commandables, la transition d'entrée notée  $t_u$  est la seule transition commandable. Le comportement de ce graphe est représenté par l'équation Max-Plus linéaire (1.11). Notre objectif est de synthétiser un contrôleur  $u(k)$  qui garantit la satisfaction de la contrainte temporelle (2.3).

En appliquant la propriété 2, pour  $\tau = \phi$ , tel que  $\phi$  est un entier avec  $\phi \geq 1$ , à l'équation Max-plus linéaire du graphe d'événement temporisé, on obtient l'expression explicite suivante :

$$x(k) = A^\phi \cdot x(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B) \cdot u(k - k') \right], \quad (2.4)$$

La  $i^{\text{me}}$  composante des termes de cette équation, est donné par :

$$x_i(k) = \bigoplus_{r=1}^N A_{ir}^\phi \cdot x_r(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i \cdot u(k - k') \right] \quad (2.5)$$

En remplaçant la fonction  $x_i(k)$  par son expression donnée en (2.5) dans l'inéquation (2.3), on obtient l'inéquation suivante :

$$\bigoplus_{r=1}^N A_{ir}^\phi \cdot x_r(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i \cdot u(k - k') \right] \leq \rho^{\max} \cdot x_j(k - m_\rho). \quad (2.6)$$

Du fait qu'il existe un chemin  $\alpha$  qui relie la transition de commande  $t_u$  à la transition  $t_j$  (Hypothèse 1) et à partir de la définition de la fonction dateur de la transition  $t_j$ , la relation entre les deux transitions est donnée par :

$$x_j(k) \geq (A^{m_\alpha} \cdot B)_j \cdot u(k - m_\alpha). \quad (2.7)$$

Avec  $m_\alpha$  représente le nombre de jetons de chemin  $\alpha$ .

En prenant en considération l'hypothèse 2 (franchissement au plus tôt), et en remplaçant la fonction  $x_j(k)$  par son expression (2.7) dans l'inéquation (2.6) on obtient :

$$\bigoplus_{r=1}^N A_{ir}^\phi \cdot x_r(k - \phi) \oplus \left[ \bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i \cdot u(k - k') \right] \leq \tau_\rho^{\max} \cdot (A^{m_\alpha} \cdot B)_j \cdot u(k - m_\rho - m_\alpha) \quad (2.8)$$

Alors, la contrainte (2.3) est satisfaite si les deux inéquations suivantes sont vérifiées :

$$\bigoplus_{r=1}^N A_{ir}^{\phi} \cdot x_r(k - \phi) \leq \tau_{\rho}^{max} \cdot (A^{m_{\alpha}} \cdot B)_j \cdot u(k - m_{\rho} - m_{\alpha}) \quad (2.9)$$

$$\bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i \cdot u(k - k') \leq \tau_{\rho}^{max} \cdot (A^{m_{\alpha}} \cdot B)_j \cdot u(k - m_{\rho} - m_{\alpha}) \quad (2.10)$$

Les deux inéquations (2.9) et (2.10) peuvent être décrites par les deux inéquations suivantes :

$$u(k - m_{\rho} - m_{\alpha}) \geq \bigoplus_{r=1}^N A_{ir}^{\phi} - \tau_{\rho}^{max} - (A^{m_{\alpha}} \cdot B)_j \cdot x_r(k - \phi) \quad (2.11)$$

$$\bigoplus_{k'=0}^{\phi-1} (A^{k'} \cdot B)_i \cdot u(k - k') \leq \tau_{\rho}^{max} \cdot (A^{m_{\alpha}} \cdot B)_j \cdot u(k - m_{\rho} - m_{\alpha}) \quad (2.12)$$

La contrainte temporelle (2.3) est satisfaite si les deux inéquations (2.11) et (2.12) sont satisfaites.

Dans le Théorème 3 on définit la formule des lois de commande causales qui satisfont la contrainte temporelle (2.3), dans le cas où la condition (2.14) est satisfaite.

**Théoreme 3.** *Nous considérons un graphe d'événements temporisé  $G$  avec une seule transition de commande et soumis à une seule contrainte temporelle sous la forme (2.3). Nous supposons qu'il existe un chemin  $\alpha$  qui relie la transition de commande  $t_u$  à la transition d'entrée de chemin sous contrainte  $\rho$  avec ( $m_{\alpha} = 0$ ). Le chemin  $\rho$  n'est pas marqué initialement. Le contrôleur décrit par l'équation suivante :*

$$u(k) = \bigoplus_{r=1}^N F_r \cdot x_r(k - 1). \quad (2.13)$$

avec  $F_r = \max(0, A_{ir} - \tau_{\rho}^{max} - B_j)$ , satisfait la contrainte temporelle (2.3) si la condition suivante :

$$B_i \leq \tau_{\rho}^{max} \cdot B_j \quad (2.14)$$

est vérifiée.

**Démonstaration.** Comme nous l'avons expliqué précédemment, on cherche un contrôleur sous la forme  $u(k) = F \cdot x(k - 1)$ . Pour cela, on prend  $\phi = 1$  ce qui signifie que

$m_\alpha = m_\rho = 0$ , alors  $0 \leq k' \leq \phi - 1$  ce qui implique  $k' = 0$ .

Nous remplaçons  $\phi$ ,  $m_\alpha$  et  $m_\rho$  par leurs valeur dans les deux inéquations (2.11) et (2.12). L'inéquation (2.11) donne la formule de la loi de commande qui garantit la satisfaction de la contrainte temporelle (2.3) si la condition donnée par l'inéquation (2.12) est bien versifiée.  $\square$

### 2.3.1.2. Plusieurs contraintes

Nous considérons dans ce cas un graphe d'événements temporisé décrit par son équation d'état (1.11), soumis à  $S$  contraintes temporelles (notées  $\rho_s$ , avec  $s = 1$  à  $S$ ). Les transitions  $t_i$  et  $t_j$  notent respectivement la transition d'entrée et la transition de sortie de chemin  $\rho_s$ . Pour chaque chemin  $\rho_s$ , nous avons  $m_{\rho_s}$ ,  $\tau_{\rho_s}$  et  $\tau_{\rho_s}^{max}$  qui représentent respectivement le marquage initial, la temporisation minimale et la temporisation maximale de ce chemin. Les fonctions  $x_i(k)$  et  $x_j(k)$  définissent les dateurs correspondants aux transitions  $t_i$  et  $t_j$ .

Les contraintes temporelles imposées sur ce graphe sont exprimées par l'inéquation suivante :

$$x_i(k) \leq \tau_{\rho_s}^{max} \cdot x_j(k), \quad s = 1, \dots, S \quad (2.15)$$

Pour  $s$  allant de 1 à  $S$ , nous prenons toujours en considération l'Hypothèse 1, et nous supposons que pour tout  $s$  allant de 1 à  $S$ , il existe au moins un chemin qui relie la transition de commande  $t_u$  à la transition  $t_j$  qu'est la transition d'entrée de chemin sous contrainte, nous notons ce chemin par  $\alpha_s$ .

Le marquage initial du ce chemin est notée par la valeur de  $m_{\alpha_s}$ . La relation entre ces deux transitions est donnée en fonction de  $s$  par l'équation :

$$x_j(k) \geq (A^{m_{\alpha_s}} \cdot B)_j \cdot u(k - m_{\alpha_s}) \quad (2.16)$$

Nous pouvons appliquer les résultats du Théorème 3 pour chaque contrainte temporelle imposée sur ce graphe. Nous notons par  $u_s(k)$  les lois de commande qui garantit la  $s^{me}$  contrainte. Les lois de commande  $u_s(k)$  sont obtenues en fonction du paramètre  $s$ , et elles sont données par :

$$u_s(k) = \oplus_{r=1}^N (A_{ir} - \tau_{\rho_s}^{max} - B_j) \cdot x_r(k - 1)$$

avec  $\phi_s = m_{\alpha_s} + m_{\rho_s} + 1$  pour  $s$  allant de 1 à  $S$ .

Le contrôleur qui garantit le respect de toutes les contraintes, correspond au contrôleur maximum entre les contrôleurs  $u_s(k)$  calculé pour  $s$  allant de 1 à  $S$ .

Ces résultats sont résumés dans le théorème suivant :

**Théorème 4.** *Soit  $G$  un graph d'événements temporisé, dont le comportement est décrit par l'équation (1.11), ce graphe est soumis à  $S$  contraintes temporelles sous la forme de (2.15), tel que  $m_{\alpha_s} = m_{\rho_s} = 0$  et  $\phi_s = 1$ . Le contrôleur donné par :*

$$u(k) = \bigoplus_{s=1}^S u_s(k) \quad (2.17)$$

avec,

$$u_s(k) = F_{r_s} \cdot x_r(k-1) \text{ avec } F_{r_s} = \max(0, (A_{ir} - \tau_{\rho_s}^{max} - B_j))$$

satisfait toutes les  $S$  contraintes temporelles si les conditions suivantes

$$B_i \leq \tau_{\rho_s}^{max} \cdot B_j \quad (2.18)$$

sont vérifiées pour chaque indice  $s$  correspondant à la contrainte imposée sur le chemin  $\rho_s$ .

**Démonstaration.** D'après le Théorème 3, si la condition 2.14 est satisfaite, alors les lois de commande données par  $u_s(k)$  satisfaisant la  $s^{me}$  contrainte temporelle 2.15.

Pour  $s = 1$  à  $S$ , nous avons l'inéquation suivante :

$$\bigoplus_{s=1}^S u_s(k) \geq u_s(k)$$

À partir de cette équation, il est évident que le contrôleur  $u(k) = \bigoplus_{s=1}^S u_s(k)$  satisfait les  $S$  contraintes temporelles 2.15.

□

**Exemple 6.** *Nous considérons un système de transmission de l'information. Ce système effectue deux tâches A et B en série. Après l'exécution de la tâche A l'information est transmise afin d'effectuer la tâche B. Une contrainte temporelle est imposée sur le temps*

d'exécution de la tâche A et le temps de transmission de l'information. Le temps de l'exécution de la tâche A n'est pas fixe, il varie en entre 2 ms et 5 ms, le temps d'exécution de la tâche B est fixé à 3 ms. Le temps de transmission ( $i_1$ ) est fixé à 10 ms. Le temps de l'exécution de la tâche A et le temps de transmission  $i_1$  ne doivent pas dépasser 15 ms.



FIG. 2.3: Système de transmission d'informations

Nous prenons en considération le cahier des charges de système de transmission de l'information représenté à la figure Fig.2.3.

Ce système peut être modélisé par un graphe d'événements temporels donné à la Fig.2.4.

La contrainte temporelle imposée à ce système est imposée sur le chemin allant de la

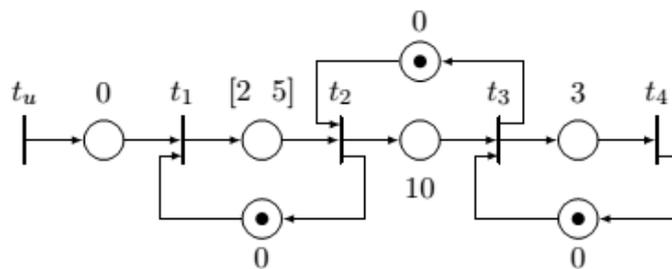


FIG. 2.4: modèle GET de système de transmission d'informations

transition de  $t_1$  à la transition  $t_3$ . Cette contrainte peut être représentée par l'inéquation suivante :

$$x_3(k) \leq 15.x_1(k)$$

Notre objectif est de calculer une loi de commande  $u(k)$  à appliquer sur le système de transmission d'informations afin de garantir les spécifications de système. Le système possède une seule transition d'entrée, cette transition est la seule transition du système qui est contrôlable.

Nous appliquons le Théorème 3 pour calculer les lois de commande qui garantissent la

contrainte temporelle imposée sur ce système. Pour cela, nous donnons tout d'abord l'équation d'état du graphe de la Fig.2.4.

En appliquant les résultats de la section 1.3.2.2, l'équation d'état de ce système est représentée par l'équation (2.19) :

$$x(k) = \begin{bmatrix} \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & 2 & e & \varepsilon \\ \varepsilon & 12 & 10 & e \\ \varepsilon & 15 & 13 & 3 \end{bmatrix} \cdot x(k) \oplus \begin{bmatrix} e \\ 2 \\ 12 \\ 15 \end{bmatrix} \cdot u(k). \quad (2.19)$$

L'hypothèse du Théorème 3 est vérifiée du fait qu'il existe un chemin vide  $\alpha$  qui relie la transition de commande  $t_u$  à la transition  $t_1$  qui est la transition d'entrée de chemin sous contrainte  $\rho$ . La temporisation minimale de ce chemin est  $\tau_\rho = 12$ , et la temporisation maximale qui ne devrait pas le dépasser est  $\tau_\rho^{\max} = 15$ .

D'après l'équation (2.19) on a :  $B_j = B_1 = e, B_i = B_3 = 12, A_{ir} = A_{3r} = (\varepsilon \ 12 \ 10 \ e)$ . On a  $B_3 \leq \tau^{\max} \cdot B_1$ , la condition du Théorème est donc bien vérifiée. Alors il existe une loi de commande qui satisfait la contrainte temporelle imposée sur le système. Cette loi de commande est donnée par l'équation suivante :

$$\begin{aligned} u(k) &= \bigoplus_{r=1}^4 (A_{3r} - \tau_\rho^{\max} - B_1) \cdot x_r(k-1) \\ &= \bigoplus_{r=1}^4 (A_{3r} - 15 - 0) \cdot x_r(k-1) \end{aligned}$$

On prend en considération la remarque 2.1 et on remplace les termes  $(A_{3r} - 15 \leq 0)$  par l'élément neutre  $e$ . Alors, Le contrôleur est donné par l'équation suivante :

$$u(k) = e \cdot x_2(k-1) \oplus e \cdot x_3(k-1) \oplus e \cdot x_4(k-1)$$

Ce contrôleur peut être simplifié. D'après le graphe de la Fig.2.4, Il est clair que, la date de franchissement de la transition  $t_3$  est supérieure à la date de franchissement la transition  $t_2$ , et que la date de franchissement de la transition  $t_4$  supérieur à la date de franchissement de la transition  $t_3$  ce qui donne  $x_4(k-1) \geq x_3(k-1) \geq x_2(k-1)$ .

Alors la loi de commande est donnée par l'équation suivante :

$$u(k) = e.x_4(k-1) = 3.x_3(k-1)$$

Ce contrôleur peut être représenté par une place marquée et temporisée connectée au modèle de graphe d'événements temporisés de système (Fig.2.5.) afin de garantir la contrainte temporelle imposée à ce système.

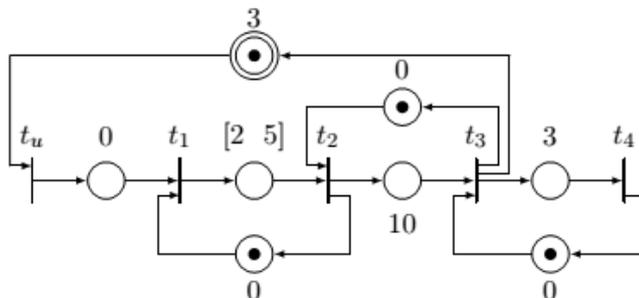


FIG. 2.5: modèle GET de système de transmission d'informations avec la place de contrôle

### 2.3.2 Commande multivariables

Dans cette section, nous traitons le problème de la commande d'un graphe d'événements temporisé avec plusieurs transitions de commande. Comme dans la section précédente, nous supposons que ce graphe est soumis à des contraintes temporelles, et que le comportement de ce graphe est représenté par une équation linéaire dans l'algèbre Max-Plus sous forme de l'équation 1.11, avec  $u(k) \in \bar{\mathbb{R}}_{max}^m$ , et  $B \in \bar{\mathbb{R}}_{max}^{N \times m}$ , tel que  $m$  représente le nombre de transitions d'entrées et  $N$  le nombre des transitions internes. Notre objectif est de calculer les composantes de vecteur de commande  $u(k)$  qui garantit la satisfaction des contraintes temporelles imposées sur le graphe considéré. Dans un premier temps, nous considérons le cas d'un GET avec plusieurs transitions d'entrées et soumis à une seule contrainte temporelle, puis, nous généralisons notre étude pour le cas d'un GET avec plusieurs transitions d'entrées est soumis à plusieurs contraintes temporelles.

#### 2.3.2.1. Une seule contrainte

Nous considérons dans ce cas un GET avec  $m$  transitions d'entrées et soumis à une seule contrainte temporelle imposée sur le chemin  $\rho$ . On associe pour chaque transition source, une fonction dateur notée  $u_l(k)$  pour  $l = 1 \dots, m$ . Nous supposons toujours qu'il existe un chemin, noté  $\alpha$  avec un marquage initial  $m_\alpha$ , qui relie la transition  $t_{uc}$  à la transition  $t_j$  qu'est la transition d'entrée du chemin sous contrainte  $\rho$ .

En prenant en considération cette hypothèse, la relation entre la transition source  $tu_c$  et la transition  $t_j$  est donnée par :

$$x_i(k) \geq (A^{m_\alpha} B)_{jc} u_c(k - m_\alpha)$$

Nous supposons maintenant que le marquage du chemin  $\alpha$  et du chemin  $\rho$  sont nul, c'est-à-dire,  $m_\alpha = m_\rho = 0$ . Dans ce cas, on a,  $\phi_c = m_{\alpha_c} + m_\rho + 1 = 1$ . Alors les inéquations 2.9, 2.10 peuvent s'écrire comme suite :

$$\bigoplus_{r=1}^N A_{ir} \cdot x_r(k - 1) \leq \tau_\rho^{max} \cdot (B)_{jc} \cdot u_c(k) \quad (2.20)$$

$$\bigoplus_{l=1}^m B_{il} \cdot u_l(k) \leq \tau_\rho^{max} \cdot B_{jc} \cdot u_c(k) \quad (2.21)$$

Les deux inéquations (2.20) et (2.21) peuvent être écrites par ces deux inéquations :

$$u_c(k) \geq \bigoplus_{r=1}^N [\tau_\rho^{max} \cdot (B)_{jc} - A_{ir}] x_r(k - 1) \quad (2.22)$$

$$\bigoplus_{l=1}^m B_{il} \cdot u_l(k) \leq \tau_\rho^{max} \cdot B_{jc} \cdot u_c(k) \quad (2.23)$$

L'inéquation 2.22 donne la formule de la  $S^{\text{ème}}$  composante du vecteur de commande  $u(k)$ . Cette composante permet de vérifier la contrainte temporelle 2.3, si la condition 2.23 est vérifiée.

La condition 2.23 est vérifiée si les deux conditions suivantes sont vérifiées pour  $l = 1$  à  $m$  :

$$B_{il} \leq \tau_\rho^{max} \cdot B_{jc}, \quad (2.24)$$

$$u_l(k) \leq u_c(k). \quad (2.25)$$

**Théoreme 5.** *Soit  $G$  un GET avec  $m$  transitions d'entrées et soumis à une seule contrainte sous la forme de (2.3). Le respect de la contrainte temporelle (2.3) est garanti, s'il existe un indice  $c$  tel que*

$$u_c(k) \geq \bigoplus_{r=1}^N F_{cr} = 1^N [F_{cr} \cdot x_r(k - 1)]$$

avec :  $F_{cr} = \max(0, \tau_{\rho}^{max} \cdot (B)_{jc} - A_{ir})$  et  $u_l(k) = \varepsilon$  pour  $l \neq c$  et si la condition (2.24) est satisfaite.

**Démonstaration.** Comme indiqué précédemment, le contrôleur (2.22) garantit la satisfaction de la contrainte temporelle (2.3) si les deux conditions données par 2.24 et (2.25) sont vérifiées.

La condition  $B_{il} \leq \tau_{\rho}^{max} \cdot B_{jc}$  est simple à vérifier pour  $l = 1$  à  $m$ . Il suffit de trouver une seule composante de vecteur  $u(k)$  qui assure la satisfaction de la contrainte (2.3), alors il suffit de choisir  $u_l(k) = \varepsilon$  pour  $l \neq c$ , pour que la condition (2.25) soit vérifiée pour  $l = 1$  à  $m$ .

Alors les feedbacks donnés par le Théorème 5 assurent la satisfaction de la contrainte (2.3) si la condition (2.24) est bien vérifiée. Nous avons montré dans cette section que pour respecter une contrainte temporelle, il suffit de trouver une composante  $u_c(k)$  de vecteur de commande  $u(k)$  pour laquelle la condition (2.24) est vérifiée et il faut aussi fixer les autres composantes du vecteur  $u(k)$  à  $\varepsilon$ .

Dans la section suivante, nous généralisons notre étude pour le cas de la commande des graphes d'événements temporisés avec plusieurs transitions d'entrées et soumis à plusieurs contraintes temporelles.

### 2.3.2.2. Plusieurs contraintes

Soit  $G$  un graphe d'événements temporisé avec  $m$  transitions d'entrées. Ce graphe est soumis à  $s$  contraintes temporelles. Ces contraintes temporelles sont imposées sur certains chemins qu'on note  $\rho_s$  pour  $s = 1$  à  $S$ . Nous notons par  $t_j$  et  $t_i$  respectivement la transition d'entrée et la transition de sortie de chemin  $\rho_s$ . Nous notons par  $\tau_{\rho_s}^{max}$  le temps maximal qui ne doit pas être dépassé par un jeton entré dans le chemin  $\rho_s$ .

Nous supposons, qu'il existe au moins un chemin  $\alpha_s$  qui relie une transition de commande  $t_{u_{cs}}$  à la transition d'entrée du chemin sous contrainte  $\rho_s$ . Cette hypothèse est exprimée par l'inéquation suivante :

$$x_j(k) \geq (A^{m_{\alpha_s}} \cdot B)_{jc_s} \cdot u_{cs}(k - m_{\alpha_s}). \quad (2.26)$$

Pour  $s=1$  à  $S$ .

Les contraintes temporelles imposées sur le graphe  $G$  sont exprimées par le système d'équation suivant :

$$x_j \leq \tau_{\rho_s}^{max} x_i(k - m)_{\rho_s}, s = 1$$

a S. (2.27)

L'équation 2.4 peut s'écrire sous la forme suivante :

$$x(k) = \oplus_{r=1}^N (A^{\phi_s} . x(k - \phi_s)) \oplus \left[ \bigoplus_{k'=0}^{\phi_s-1} (A^{k'} . B) . u(k - k') \right], \quad (2.28)$$

La fonction  $x_i(k)$  est donnée par l'inéquation :

$$x_i(k) = \oplus_{r=1}^N (A^{\phi_s} . x(k - \phi_s)) \oplus \left[ \bigoplus_{k'=0}^{\phi_s-1} (\bigoplus_{l=1}^m (A^{k'} . B)_{il} . u_l(k - k')) \right], \quad (2.29)$$

Nous remplaçons maintenant les deux inéquations 2.26 et 2.29 dans l'inéquation de contrainte 2.3.2.

Nous constatons que les contraintes temporelles sont satisfaites, si les deux inéquations suivantes sont vérifiées :

$$\bigoplus_{r=1}^N (A^{\phi_s}_{ir} . x(k - \phi_s)) \leq \tau_{\rho_s}^{max} (A^{m_{\alpha_s}} . B)_{jc_s} . u_{c_s}(k - m_{\alpha_s} - m_{\rho_s}) \quad (2.30)$$

$$\bigoplus_{k'=0}^{\phi_s-1} (\bigoplus_{l=1}^m (A^{k'} . B)_{il} . u_l(k - k')), \leq \tau_{\rho_s}^{max} (A^{m_{\alpha_s}} . B)_{jc_s} . u_{c_s}(k - m_{\alpha_s} - m_{\rho_s}) \quad (2.31)$$

Pour  $s=1$  à  $S$ , et  $\phi_s = m_{\rho_s} + m_{\alpha_s} + 1$

À partir de ces deux inéquations nous constatons que si la condition 2.31 est vérifiée, alors le contrôleur tiré de l'inéquation 2.30 donné par la formule qui suit :

$$u_{c_s}(k) \geq \bigoplus_{r=1}^N [((A^{\phi_s})_{ir} - (A^{m_{\alpha_s}} . B)_{jc_s} - \tau_{\rho_s}^{max}) . x_r(k - 1)] \quad (2.32)$$

garantit la satisfaction des contraintes temporelles imposées sur le graphe  $G$ .

Nous remarquons que la condition 2.31 est vérifiée si les deux inéquations suivantes sont vérifiées :

$$(A^{k'} . B)_{il} \leq \tau_{\rho_s}^{max} (A^{m_{\alpha_s}} . B)_{jc_s} \quad (2.33)$$

$$u_l(k - k') \leq u_{c_s}(k - m_{\alpha_s} - m_{\rho_s}) \quad (2.34)$$

avec  $k' = 0$  à  $\phi_s - 1$  et  $l=1$  à  $m$ .

Le Théorème suivant définit le vecteur de commande  $u(k)$  qui assure la satisfaction de toutes les  $s$  contraintes temporelles.

**Théorème 6.** *Les composantes du vecteur  $u(k)$  définies par :*

$$u_l(k) = \bigoplus_{s=1}^S u_{c_s}(k)$$

pour  $l=1$  à  $m$  et  $u_l(k) = \varepsilon$  pour  $l \neq c_s$  avec  $s=1$  à  $S$ .

Tel que

$$u_{c_s}(k) = \bigoplus_{r=1}^N F_{rc} \cdot x_r(k-1)$$

avec  $F_{rc} = \max(0, [((A^{\phi_s})_{ir} - (A^{m_{\alpha_s}} \cdot B)_{jc_s} - \tau_{\rho_s}^{max}))]$

pour  $m_{\alpha_s} = m_{\rho_s} = 0$ , et  $\phi_s = 1$

représentent les lois de commandes causales qui garantissent le respect de toutes les contraintes temporelles (2.3.2) si la condition (2.33) est vérifiée.

**Démonstaration.** Comme nous l'avons démontré précédemment, les contraintes 2.3.2 sont satisfaites si les deux conditions 2.30, 2.31 sont bien vérifiées. Pour  $m_{\alpha_s} = m_{\rho_s} = 0$ , nous pouvons extraire à partir de l'inéquation 2.30 la formule du contrôleur qui va satisfaire les contraintes 2.3.2 si la condition 2.31 est bien vérifiée.

La satisfaction de la condition 2.31 est équivalente à la satisfaction des deux conditions 2.33 et 2.34.

La condition 2.33 est simple à vérifier. Pour la deuxième condition, en prenant en considération l'hypothèse  $u_l(k) = \bigoplus_{s=1}^S u_{c_s}(k)$ , la condition 2.34 est vérifiée pour tout  $u_l(k) = \varepsilon$  pour  $l \neq c_s$ .

Finalement les lois de commande données dans le Théorème 6 garantissent le respect de  $S$  contraintes temporelles imposées sur le système commandé.  $\square$

**Exemple 7.** Nous considérons dans cet exemple le graphe événements temporisé décrit par la Fig.2.6. Ce graphe représente un atelier d'assemblage, constitué de : deux fours  $p_{21}$ ,  $p_{65}$  ; deux zones de refroidissement modélisent par les deux places  $p_{32}$  et  $p_{76}$  ; deux stocks limités à une seule pièce représentée par les deux places  $p_{43}$  et  $p_{47}$  et une machine d'assemblage modélisée par la place  $p_{84}$ .

A chaque place est associée une temporisation qui représente le temps de séjour minimum pour qu'il puisse contribuer au franchissement de la transition située en aval de cette place. Les transitions étiquetées  $t_{u1}$ ,  $t_{u2}$  représentent les transitions d'entrée de ce graphe. Ces deux transitions sont les seules transitions commandables.

Le graphe est soumis à deux contraintes temporelles imposées sur les deux chemins  $\rho_1$  et

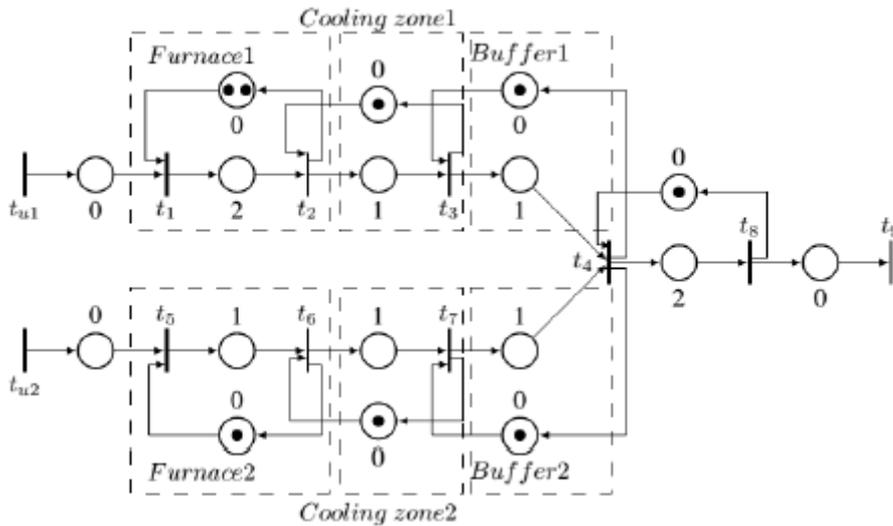


FIG. 2.6: Modèle Graphe d'évènements temporisé d'un système d'assemblage

$\rho_2$ . Le chemin  $\rho_1$  relie la transition  $t_1$  à la transition  $t_3$  et le chemin  $\rho_2$  connecte la transition  $t_5$  à la transition  $t_7$ . Ces deux contraintes sont représentées par les deux inéquations Max-Plus suivantes :

$$x_3(k) \leq 3 \cdot x_1(k)$$

$$x_7(k) \leq 2 \cdot x_5(k)$$

Où  $x_1(k)$ ,  $x_3(k)$ ,  $x_5(k)$ ,  $x_7(k)$  représentent respectivement les fonctions dateurs des transitions  $t_1$ ,  $t_3$ ,  $t_5$ ,  $t_7$ .

Notre objectif est de calculer le vecteur de commande  $u(k)$  qui garantit la satisfaction de

ces deux contraintes temporelles.

Afin d'atteindre cet objectif, nous appliquons le Théorème 6. Nous commençons tout d'abord par le calcul de l'équation d'état de graphe de la Fig.2.6. Pour cela, nous décomposons la place  $p_{21}$  qui comporte deux jetons en deux places avec un jeton pour chacune, et on rajoute une transition supplémentaire  $t_{10}$ . L'équation d'état de système est représentée par l'équation suivante :

$$x(k) = \begin{bmatrix} \varepsilon & e \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & 1 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 2 & 1 & \varepsilon & 3 & 2 & e & \varepsilon & 4 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & 2 & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 4 & 3 & \varepsilon & 5 & 4 & 2 & \varepsilon & 6 \\ \varepsilon & \varepsilon & 4 & 3 & \varepsilon & 5 & 4 & 2 & \varepsilon & 6 \\ \varepsilon & e & \varepsilon \end{bmatrix} \cdot x(k-1) \oplus \begin{bmatrix} e & \varepsilon \\ 2 & \varepsilon \\ 3 & \varepsilon \\ 4 & 3 \\ \varepsilon & e \\ \varepsilon & 1 \\ \varepsilon & 2 \\ 6 & 5 \\ 6 & 5 \\ \varepsilon & \varepsilon \end{bmatrix} \cdot u(k) \quad (2.35)$$

Nous commençons maintenant par le calcul de la première composante du vecteur de commande  $u(k)$  qui garantit la première contrainte temporelle  $x_3(k) \leq 3 \cdot x_1(k)$ .

Nous remarquons que sur le graphe de la Fig.2.6 il existe un chemin  $\alpha_1$  qui relie la transition de commande  $t_{u1}$  à la transition  $t_1$  ( $t_1$  est la transition d'entrée du chemin sous contrainte  $\rho_1$ ) et nous avons  $m_{\alpha_1} = 0$ , alors l'hypothèse de Théorème 6 est bien vérifiée.

On a aussi d'après l'équation (2.35),  $\tau_{\rho_1}^{max} = 3$ ,  $B_{11} = e$ ,  $B_{3l} = (B_{31} \ B_{32}) = (3 \ \varepsilon)$ .

Nous pouvons facilement vérifier que la condition de Théorème 6  $B_{3l} \leq \tau_{\rho_1}^{max} + B_{11}$  est vérifiée pour  $l = 1$  à  $2$ . Alors il existe une loi de commande qui garantit la satisfaction de la première contrainte temporelle.

La première composante du vecteur  $u(k)$  est donnée par l'équation suivante :

$$u_1(k) = \bigoplus_{r=1}^{10} (A_{3r} - 3 - B_{11}) \cdot x_r(k-1)$$

Pour  $r = 1$  à  $10$  où  $A$  et  $B$  sont respectivement la matrice d'état et la matrice de commande donnée par l'équation (2.35).

Nous avons  $B_{11} = e = 0$ , et  $A_{3r}$  représente les composantes de la ligne 3 de la matrice  $A$ . Nous remplaçons  $B_{11}$  par sa valeur et nous remplaçons aussi chaque terme  $(A_{3r} - 3) \leq 0$  par l'élément neutre  $e$  en se basant sur le Théorème 6.

Le contrôleur peut être écrit sur la forme suivant :

$$u_1(k) = e \cdot x_3(k-1) \oplus e \cdot x_4(k-1) \oplus e \cdot x_8(k-1).$$

C'est claire que la date de franchissement de la transition  $t_8$  est plus grande que la date de  $(k-1)$ ème fois franchissement de la transition  $t_3$  et  $t_4$ . Alors, nous avons  $x_8(k-1) \geq x_4(k-1) \geq x_3(k-1)$ , Finalement, la première composante du vecteur  $u(k)$  est donnée par :

$$u_1(k) = e \cdot x_8(k-1).$$

Avec la même procédure, on calcule la deuxième composante du vecteur de commande  $u(k)$  qui garantira la satisfaction de la deuxième contrainte temporelle  $x_7(k) \leq 2 \cdot x_5(k)$ . D'après la Fig.2.6, Nous constatons qu'il existe un chemin  $\alpha_2$  qui relie la transition de commande  $t_{\alpha_2}$  à la transition  $t_5$  qu'est la transition d'entrée de deuxième chemin sous contrainte  $\rho_2$ , et on a  $m_{\alpha_2} = 0$ , alors l'hypothèse de théorème 5 est bien vérifiée. Nous vérifions maintenant la condition à satisfaire pour l'existence des lois de commande qui garantissent la satisfaction de la contrainte temporelle. La condition du Théorème 6 est donnée par :

$$B_{7l} \leq \tau_{\rho_2}^{max} + B_{52}$$

Où  $B$  est la matrice de commande et  $l = 1$  à  $2$ .

Nous avons d'après l'équation d'état (2.35),  $\tau_{\rho_2}^{max} = 2$ ,  $B_{52} = e$ ,  $B_{7l} = (B_{71} \ B_{72}) = (\varepsilon \ 2)$ . La condition est bien vérifiée, alors il existe des lois de commande qui garantissent la contrainte temporelle imposée sur le chemin  $\rho_2$ . Ces lois de commande sont données par l'équation suivante :

$$\begin{aligned} u_2(k) &= \bigoplus_{r=1}^{10} (A_{7r} - 2 - B_{52}) \cdot x_r(k-1) \\ &= \bigoplus_{r=1}^{10} (A_{7r} - 2) \cdot x_r(k-1) \end{aligned}$$

D'après le Théorème 6, Nous avons  $F_r = \max(0, A_{ir} - \tau_{\rho_s}^{max} - B_{sj})$  où  $F_r = A_{ir} - \tau_{\rho_s}^{max} - B_{sj}$ . Nous remplaçons les éléments  $(A_{7r} - 2) \leq 0$  par l'élément neutre  $e = 0$ . Les lois de commande qui représentent la deuxième composante du vecteur de commande  $u(k)$  sont données par l'équation suivante :

$$u_2(k) = e \cdot x_4(k - 1) \oplus e \cdot x_6(k - 1) \oplus e \cdot x_7(k - 1).$$

A partir du graphe de la Fig.2.6 nous constatons que la date de  $(k-1)$ ème franchissement de la transition  $t_4$  est plus grande que la date de  $(k-1)$ ème franchissement de la transition  $t_7$  et  $t_6$ , alors nous avons,  $x_4(k - 1) \geq x_7(k - 1) \geq x_6(k - 1)$ . Finalement, la deuxième composante du vecteur de  $u(k)$  est donnée par :

$$u_2(k) = e \cdot x_4(k - 1).$$

Le vecteur de commande qui garantit la validité des deux contraintes temporelles imposées aux deux chemins  $\rho_1$  et  $\rho_2$  du système considéré est donné par :

$$\mathbf{u}(k) = \begin{bmatrix} e \cdot x_8(k - 1) \\ e \cdot x_4(k - 1) \end{bmatrix}$$

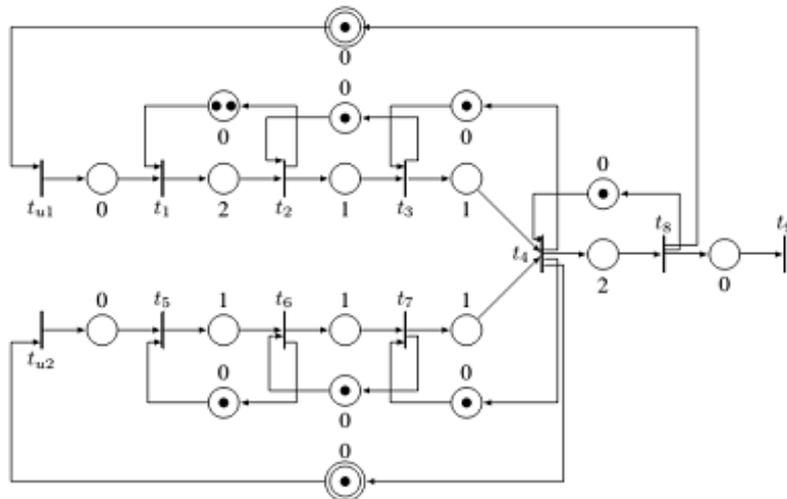


FIG. 2.7: Modèle GET de système d'assemblage avec les places de contrôle

Ces lois de commande sont des feedbacks représentées par des places marquées connectées au modèle GET du système d'assemblage considéré. Ces lois de commande sont représentées par des places en double cercle dans la Fig.2.7.

### 2.3.3 Application sur une architecture d'automatisation en réseau

Les architectures d'automatisation en réseau (AAR) sont des systèmes temps réel et embarqués. Le domaine des architectures d'automatisation en réseau est relativement nouveau dans la communauté académique du contrôle, mais il ne l'est pas dans l'industrie. On les retrouve dans les systèmes de production, les avions, les voitures...etc. Ces systèmes sont caractérisés par l'existence de contrainte temporelle dont il faut tenir compte. Dans les systèmes temps réel et embarqués en général ou dans les architectures d'automatisation en réseau en particulier, les données ont une durée de vie limitée et deviennent obsolètes après un certain temps, les événements apparaissent à des instants particuliers et doivent être pris en compte au bout de délai connu à l'avance. Les traitements (ou actions) ont souvent des instants de début, de fin et des durées d'exécutions fixées. Alors l'analyse des performances temporelles de ces systèmes de commande est nécessaire. Dans la littérature, il existe plusieurs travaux qui s'intéressent à l'évaluation des performances temporelles des systèmes de commande en réseau.

Nous pouvons citer à titre d'exemple la méthode analytique publiée dans [7]. Cette approche est basée sur l'utilisation des graphes d'événements temporisés pour modéliser l'architecture d'automatisation en réseau et l'exploitation du formalisme de l'algèbre  $(\text{Max}, +)$  pour calculer les bornes de temps de réponse de ces systèmes. Par contre, les auteurs de [23] s'intéressent plus précisément à l'évaluation du temps des architectures d'automatisation en réseau de bout en bout (délais de traversée réseau de communication). L'outil utilisé dans cette méthode est le calcul réseau. Les auteurs dans [37] ont utilisé un simulateur basé sur les réseaux de Petri colorés et des tests expérimentaux pour estimer les bornes du temps de réponse des systèmes de commande en réseau.

Dans cette section, nous traitons le problème de commande d'une architecture d'automatisation en réseau soumis à une borne maximale du temps de réponse. Nous appliquons les résultats que nous avons obtenus dans le Théorème 3, pour calculer des lois de commande causales qui garantissant la satisfaction de la contrainte temporelle stricte imposée sur la réactivité d'une architecture d'automatisation distribuée sur un réseau. Avant de calculer

les lois de commande qui satisfaisant les contraintes temporelles imposées sur une AAR, nous commençons tout d'abord par la représentation de ces systèmes dont nous allons décrire les différentes parties, ainsi que leurs principes de fonctionnement. Ensuite, nous passons à la modélisation de ce système par un graphe d'événements temporisé, puis à la représentation de leur comportements par des équations linéaires dans l'algèbre Max-Plus.

### 2.3.3.1. Représentation des architectures d'automatisation en réseau

Les architectures d'automatisation en réseau sont des systèmes distribués où les capteurs, les actionneurs et la partie de commande (PLC) sont interconnectés en temps réel par un réseau de communication Fig.2.8. Le capteur transmet l'information au contrôleur, ce dernier exécute le programme d'utilisateur et envoie par la suite la commande calculée à l'actionneur. Ces systèmes sont généralement composés de 3 parties :

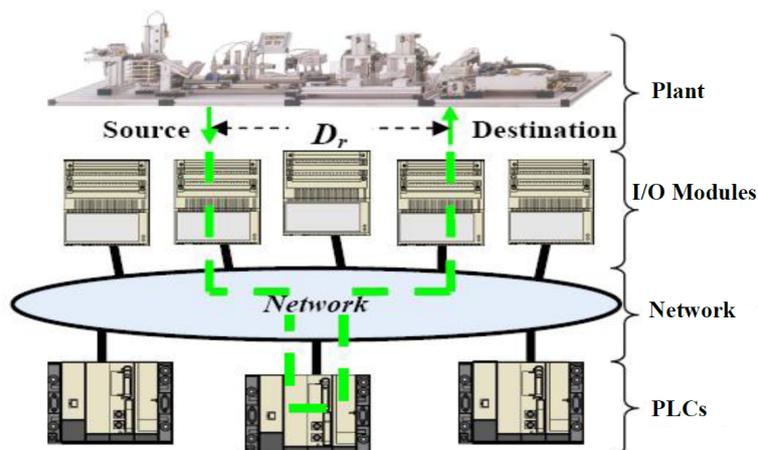


FIG. 2.8: Architecture d'automatisation en réseau

#### a) Le contrôleur

Les contrôleurs logiques jouent un rôle important dans les architectures d'automatisation en réseau. Ils doivent accomplir deux fonctions : la première est d'exécuter le programme utilisateur et la seconde est de gérer les échanges avec les modules d'entrées/sorties (pour récupération des valeurs en entrées et transmission des valeurs en sorties). Il se décompose donc en deux parties : le processus de calcul (CPU) et la carte de

communication (COM) (voir Fig.2.8).

- **CPU** : Le CPU effectue quatre tâches consécutives : il lit les données des modules d'entrées enregistrés dans la mémoire tampon 1 (durée fixe) , puis exécute le programme utilisateur (durée variable), et écrit les données de sortie dans la mémoire tampon 2 (durée fixe). A la fin il exécute d'autres traitements, indépendamment du programme (durée variable). Comme la lecture des données d'entrée et l'écriture des données de sortie sont beaucoup plus rapide que le calcul (du programme ou des autres traitements), elles sont négligées. Le CPU possède un comportement cyclique, le temps du cycle de CPU est constant.

- **carte de communication COM** : Le rôle de la carte de communication est d'assurer la transmission des données entre le CPU et les modules d'entrées/sorties. La carte de communication possède un fonctionnement périodique : elle attend d'abord les requêtes des modules d'entrées (durée variable), écrit les valeurs des modules d'entrées arrivent dans un ordre fixé par l'utilisateur dans la mémoire tampon 2 (ce dernier est située entre la carte de communication et le processeur de calcul) (durée fixe). La carte attend que le CPU exécute le programme d'utilisateur et copie toutes les sorties calculées par le CPU et les enregistre dans la mémoire tampon 1 (située entre le processeur de calcul et la carte de communication) dans sa mémoire propre, puis elle émet les requêtes (une à une) aux modules de sorties dans un ordre fixé par l'utilisateur (durée fixe). Une fois toutes les réponses sont envoyées elle attend la fin du cycle (durée variable).

#### **b)Interface de communication (NET) :**

L'interface de communication est dédié uniquement aux communications entre l'automate et les modules d'entrées /sorties. Elle est toujours prête à transmettre des requêtes des modules d'entrée à la carte de communication (COM) (délai fixe), et la trame de la carte de communication (COM) aux modules de sorties (délai fixe).

#### **b)Modules d'entrées/sorties :**

Les modules d'entrées/ sorties représentent le lien entre la partie opérative et le contrôleur.

L'état normal d'un MES est un état d'attente, dont il ne sort que lors de changement

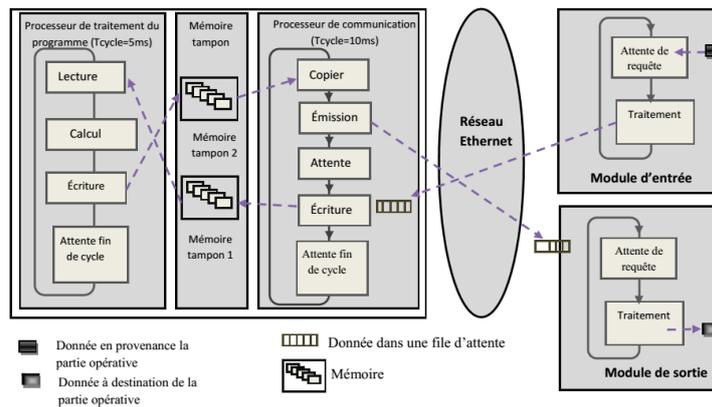


FIG. 2.9: Comportement d'une architecture d'automatisation en réseau

d'état d'un capteur de la partie opérative.

Le traitement de cette requête consiste à lire les valeurs des entrées physiques connectées au module d'entrée et d'envoyer le rapport contenant les nouvelles valeurs à la carte de communication. Cette dernière transfère et enregistre ces valeurs dans la mémoire tampon qui se trouve entre la carte de communication, la CPU exécute le programme utilisateur dans le but de calculer les valeurs de sorties. Ces nouvelles valeurs des sorties sont envoyées de la carte de communication vers le module de sortie à travers le réseau de communication.

### 2.3.3.2. Temps de réponse d'une architecture d'automatisation en réseau

Le temps de réponse ou bien le temps de réactivité d'une architecture d'automatisation en réseau peut être défini comme étant le temps qui s'écoule entre le changement d'état d'un capteur au niveau du module d'entrée et sa conséquence sur un actionneur au niveau du module de sortie issue du PLC sur le système commandé (Fig.2.10). Ce temps de réponse n'est pas constant, par contre il est décomposable en un ensemble de délais élémentaires, subis aux différents niveaux de l'AAR [16], [27].

Trouver les bornes supérieures et inférieures de cette distribution est un problème crucial pour ces systèmes. Dans la littérature, il existe plusieurs contributions dédiées à l'évaluation des performances temporelles des AARs. Nous pouvons citer à titre d'exemple, les travaux de [7], [37] où les auteurs déterminent les bornes supérieures et inférieures du

temps de réponse pour différentes catégories d'architectures d'automatisation.

Dans cette thèse, nous nous intéressons au contrôle d'une AAR sous contrainte de temps imposée sur son temps de réponse. Notre objectif est d'assurer un temps de réponse souhaité dans les systèmes d'automatisation. Pour cela, nous appliquerons les approches théoriques de la section 2.3 pour synthétiser un contrôleur qui garantit le respect de cette spécification temporelle. Tout d'abord, nous modélisons l'AAR par un graphe d'événements temporisé. Par la suite, nous décrivons le comportement de ce graphe avec des équations linéaires dans l'algèbre Max-Plus. Les détails de cette étude sont décrits dans le reste de ce chapitre.

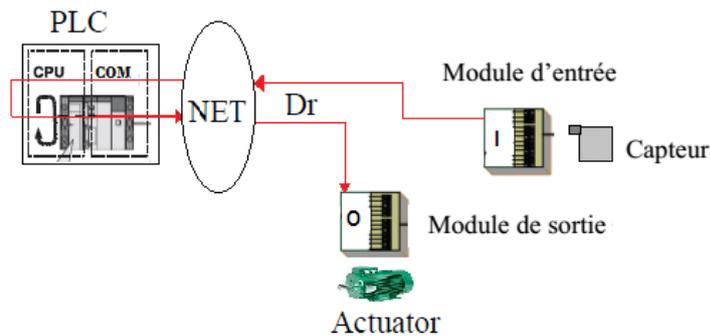


FIG. 2.10: Temps de réponse d'une architecture d'automatisation en réseau

### 2.3.3.3. Commande d'une Architecture d'automatisation en réseau

#### 2.3.3.3.1. Modélisation d'une architecture d'automatisation en réseau

Dans cette section, nous utilisons les graphes d'événements temporisés pour la modélisation d'une architecture d'automatisation en réseau. En prenant en considération le principe de fonctionnement d'une architecture d'automatisation en réseau décrits précédemment dans la Fig.2.9, le système peut être modélisé par le graphe d'événements temporisé montré dans la Fig.2.11. Ce GET comporte trois parties principales. La partie de la CPU et la partie de la carte de communication (COM), les deux parties sont liées par deux mémoires tampon qui sont le seul lien entre la CPU et la carte de communication. La troisième partie représente le module d'entrée/sortie du système. Ce dernier est relié à la carte de communication via un réseau de terrain qui assure l'échange entre ces deux

modules.

Le début du cycle du système commence par le franchissement de la transition  $t_7$  qui modélise l'arrive de requête au niveau de module d'entrée. Le module d'entrée est en phase d'attente tant qu'il n'y a pas de requête à traiter. Cette attente est représentée par le jeton de la place  $p_{78}$ . Le franchissement de  $t_7$  signifie l'arrivée d'une requête au module d'entrée. Immédiatement après la réception de cette requête, l'attente est interrompue. Un jeton dans  $p_{87}$  indique que le traitement de la requête est commencé. Ce processus dure  $\tau_{87}$  unités de temps avant de se terminer par le tire de la transition  $t_8$  et la réponse résultante du traitement est envoyée par le module d'entrée vers le calculateur à travers le réseau de communication. La requête entre dans le réseau de communication modélisé par la place  $p_{58}$  et y subit un délai de bout-en-bout total de  $\tau_{58}$  unités de temps. Le franchissement de la transition  $t_5$  indique que la requête a traversé le réseau de communication et indique également que le début du cycle de la carte de communication a commencé. Ce cycle est modélisé par la place  $p_{55}$  avec une temporisation de  $\tau_{55}$  unités de temps. La place  $p_{36}$  modélise la phase d'attente de réponse de l'unité de calcul (CPU) sur la requête envoyée par le module d'entrée. Un jeton dans la place  $p_{16}$  modélisant la mémoire tampon située entre la carte de communication et l'unité de calcul (CPU). Il indique que toutes les nouvelles valeurs des modules d'entrées sont écrites dans la mémoire tampon 1 par la carte de communication. La CPU au début du cycle modélisé par la place  $p_{11}$  avec une temporisation,  $\tau_{11}$  unités de temps exécute le programme d'utilisateur. Cette tâche est modélisée par la place  $p_{21}$  et effectuée sur trois phases : lecture, calcul et écriture. Ces phases durent  $\tau_{21}$  unités de temps. Cette tâche se termine par le franchissement de la transition  $t_2$ . Après l'exécution du programme d'utilisateur, la CPU enregistre toutes les nouvelles valeurs des modules de sorties dans la mémoire tampon 2 (située entre la CPU et la carte de communication).

Le franchissement de la transition  $t_3$  indique que la carte de communication est entrain de copier les valeurs des sorties écrites par la CPU dans la mémoire tampon 2, dans sa propre mémoire pour qu'elle l'envoie par la suite vers les modules de sorties à travers le réseau de communication. Cette tâche dure  $\tau_{43}$  unités de temps. Un jeton dans la place

$p_{109}$  indique que la réponse de la CPU est arrivée au module de sortie à travers le réseau de communication modélisé par la place  $p_{58}$ .

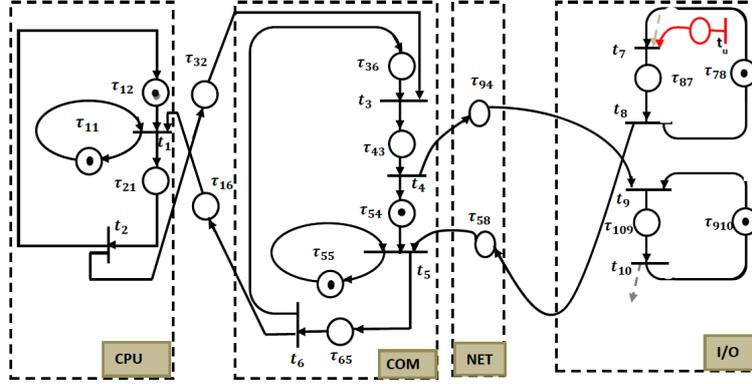


FIG. 2.11: Modèle GET d'une architecture d'automatisation en réseau

**a) Présentation Max-Plus de comportement d'une AAR**

Nous avons vu dans le premier chapitre que le comportement dynamique d'un graphe d'événements temporisé peut être décrit par des équations Max-Plus linéaire. Pour représenter le comportement dynamique du graphe de la Fig.2.11. Nous associons à chaque transition  $t_i$ , avec  $i = 1$  à  $10$ , une fonction dateur  $\theta_i(k)$ , qui représente le temps de franchissement de  $k^{\text{ème}}$  jeton. La fonction dateur attribuée à la transition d'entrée (la commande) est notée  $u(k)$ . Nous considérons un franchissement au plus tôt, c'est-à-dire que toute transition validée est immédiatement franchie.

Nous associons les temporisations ci-dessus aux places de graphe d'événements temporisé de la FIG.2.11  $\tau_{12} = \tau_{36} = \tau_{54} = \tau_{78} = \tau_{910} = \tau_{16} = \tau_{32} = 0$ ,  $\tau_{43} = \tau_{87} = \tau_{109} = \tau_{58} = \tau_{94} = \tau_{65} = 1$ ,  $\tau_{11} = 5$ ,  $\tau_{55} = 10$ ,  $\tau_{21} = 2$ .

L'équation implicite (1.9) du graphe d'événement temporisé de la FIG.2.11 s'écrit comme suit :

$$x(k) = A_0 \cdot x(k) \oplus A_1 \cdot x(k - 1) \oplus B_0 \cdot u(k)$$

tel que,

$$A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon \end{bmatrix}, \quad A_1 = \begin{bmatrix} 5 & e & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \end{bmatrix},$$

$$B_0 = \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ e \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix}.$$

Tenant compte de la section 2.3 et du Théorème 3, le comportement de l'AAR peut être représenté par une équation d'état dans l'algèbre Max-Plus.

Cette équation est représentée par la formule suivante :

$$x(k) = A \cdot x(k - 1) \oplus B \cdot u(k) \tag{2.36}$$

Tel que

$$A = A_0^* \cdot A_1 = \begin{bmatrix} 5 & e & \varepsilon & 1 & 11 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon \\ 7 & 2 & \varepsilon & 3 & 13 & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon \\ 7 & 2 & \varepsilon & 3 & 13 & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon \\ 8 & 3 & \varepsilon & 4 & 14 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & 10 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & 11 & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon \\ 9 & 4 & \varepsilon & 5 & 15 & \varepsilon & \varepsilon & 7 & \varepsilon & e \\ 10 & 5 & \varepsilon & 6 & 16 & \varepsilon & \varepsilon & 8 & \varepsilon & 1 \end{bmatrix}, B = A_0^* \cdot B_0 = \begin{bmatrix} 3 \\ 5 \\ 5 \\ 6 \\ 2 \\ 3 \\ 0 \\ 1 \\ 7 \\ 8 \end{bmatrix}$$

L'architecture d'automatisation en réseau comporte une contrainte temporelle imposée sur le temps de réponse qui est représenté par la temporisation du chemin  $\rho$  reliant la transition  $t_7$  à la transition  $t_{10}$ . Ce temps représente aussi le temps du cycle du système. Un intervalle de temps  $[\tau_\rho \ \tau_\rho^{max}]$ , tel que  $\tau_\rho$  correspond à leur temporisation minimale par contre  $\tau_\rho^{max}$  correspond à la valeur maximale à ne pas dépasser. Cette contrainte temporelle peut être représentée par une inéquation linéaire dans l'algèbre Max-Plus exprimée par l'inéquation suivante :

$$x_{10}(k) \leq \tau_\rho^{max} \cdot x_7(k)$$

Notre objectif consiste à synthétiser des lois de commande sous forme de  $u(k) = F \cdot x(k-1)$  pour garantir la contrainte temporelle imposée sur le temps de réponse de ce système.

Afin de calculer les lois de commande qui satisfont la contrainte imposée sur ce système, nous appliquons le Théorème 3 A partir du graphe de la FIG.2.11 qui modélise le système de commande en réseau, on peut remarquer qu'il existe un chemin  $\alpha$  qui relié la transition de commande  $t_u$  à la transition  $t_7$  qui est la transition d'entrée de chemin sous contrainte  $\rho$  et nous avons  $m_\alpha = 0$ . Alors hypothèse du Théorème 3. est vérifiée. Et d'après l'équation d'état 2.36 nous avons,  $B_i = B_{10} = 8$  et  $B_j = B_7 = 0$

Avant de passer au calcul des lois de commande qui garantissent la contrainte temporelle imposée sur le temps de réponse du système de commande en réseau, nous vérifions tout d'abord la condition d'existence de ces lois de commande :

$$B_{10} \leq \tau_{\rho}^{max} + B_7$$

En remplaçant  $B_{10}$ , et  $B_7$  par leurs valeurs, nous constatons alors que la condition est vérifiée si ( $\tau_{\rho}^{max} \geq 8$ )

En prenant en considération le principe de fonctionnement du l'AAR décrit précédemment, et la valeur de chaque temporisation associée aux places du modèle de ce dernier (FIG.2.11) ( $\tau_{12} = \tau_{36} = \tau_{54} = \tau_{78} = \tau_{910} = \tau_{16} = \tau_{32} = 0$ ,  $\tau_{43} = \tau_{87} = \tau_{109} = \tau_{58} = \tau_{94} = \tau_{65} = 1$ ,  $\tau_{11} = 5$ ,  $\tau_{55} = 10$ ,  $\tau_{21} = 2$ .)

Le temps de réponse du l'architecture d'automatisation en réseau prend sa valeur minimal dans le cas où il n'y a pas de délais d'attentes pour l'exécution des requêtes. La CPU et la carte de communication sont toutes les deux dans l'état de début de cycle. Dans ce cas, le temps de réponse maximal est égale à la somme des temporisations des places du graphe considéré , ce temps de réponse est donné par cette équation :

$$\tau_{\rho}^{max} \geq \tau_{87} + \tau_{58} + \tau_{65} + \tau_{16} + \tau_{21} + \tau_{32} + \tau_{43} + \tau_{94} + \tau_{109} .$$

La valeur minimale que le temps de réponse peut prendre est la valeurs 8, nous avons alors  $\tau_{\rho}^{max} \geq 8$  . Alors la condition du Théorème 3 est alors toujours vérifiée. Les lois de commande qui garantissent la contrainte temporelle imposée sur le temps de réponse est donnée par l'équation suivante :

$$u(k) \geq \bigoplus_{r=1}^{10} (A_{10r} - B_7 - \tau_{\rho}^{max}) \cdot x_r(k-1)$$

( $A_{10r}$  représenté les composantes de la ligne 10 de la matrice A,

où ( $A_{10r} = (10 \ 5 \ \varepsilon \ 6 \ 16 \ \varepsilon \ \varepsilon \ 8 \ \varepsilon \ 1)$ ). Nous remplaçons les terme de ( $A_{ir}$ ) par leurs valeur pour  $r = 1$  à 10 et  $B_7$  par sa valeur (0).

L'équation précédente devient donc :

$$u(k) \geq (10 - \tau_{\rho}^{max}) \cdot x_1(k-1) \oplus (5 - \tau_{\rho}^{max}) \cdot x_2(k-1) \oplus (6 - \tau_{\rho}^{max}) \cdot x_4(k-1) \oplus (16 - \tau_{\rho}^{max}) \cdot x_5(k-1) \oplus (8 - \tau_{\rho}^{max}) \cdot x_8(k-1) \oplus (1 - \tau_{\rho}^{max}) \cdot x_{10}(k-1).$$

Nous remplaçons maintenant les termes ( $A_{8r} - B_7 - \tau_{\rho}^{max}) \leq 0$  par l'élément neutre

$e = 0$  puisque  $\tau_\rho^{max} \geq 8$ , on a  $(5 - \tau_\rho^{max}) = e$ ,  $(6 - \tau_\rho^{max}) = e$  et  $(8 - \tau_\rho^{max}) = e$ .

Les lois de commande sont données par :

$$u(k) \geq (10 - \tau_\rho^{max}) \cdot x_1(k-1) \oplus e \cdot x_2(k-1) \oplus e \cdot x_4(k-1) \oplus (16 - \tau_\rho^{max}) \cdot x_5(k-1) \oplus e \cdot x_8(k-1) \oplus e \cdot x_{10}(k-1).$$

Dans le GET de la FIG.2.11, nous remarquons que le chemin qui relie les transitions  $t_2$ ,  $t_4$  et  $t_8$  et la transition  $t_{10}$ , la transition  $t_{10}$  est la dernière transition qui soit franchissable par rapport à ces transitions. Alors la date du kème franchissement de la transition  $t_{10}$  est supérieur à la date du kème franchissement de transitions  $t_2$ ,  $t_4$  et  $t_8$ .

On peut écrire :  $x_{10}(k-1) \geq x_4(k-1) \geq x_2(k-1) \geq x_8(k-1)$ . Alors l'inéquation des lois de commande devient :

$$u(k) \geq (10 - \tau_\rho^{max}) \cdot x_1(k-1) \oplus (16 - \tau_\rho^{max}) \cdot x_5(k-1) \oplus e \cdot x_{10}(k-1)$$

ou bien :

$$u(k) \geq F_1 \cdot x_1(k-1) \oplus F_5 \cdot x_5(k-1) \oplus F_{10} \cdot x_{10}(k-1) \tag{2.37}$$

avec

$$F_1 = (10 - \tau_\rho^{max}), F_5 = (16 - \tau_\rho^{max}), F_{10} = e.$$

Les lois de commande garantissant la satisfaction de la contrainte temporelle imposée sur le temps de réponse de l'architecture d'automatisation en réseau, sont données par l'inéquation (2.37) en fonction de  $\tau_\rho^{max}$  représentent la limite de temps de réponse à ne pas dépasser. Dans la sous-section suivante, nous calculons ces lois de commande pour une valeur de temps de réponse maximale bien définie.

• **Cas particulier** Dans cette section, nous prenons un cas d'une architecture d'automatisation en réseau sous contrainte temporelle imposée sur son temps de réponse. Cette contrainte exige au temps de réponse à ne pas dépasser la valeur 19 unités du temps.

Notre objectif est de calculer les lois de commande qui garantissent cette contrainte temporelle.

Nous prenons les lois de commande calculées par l'inéquation (2.37), et nous remplaçons  $\tau_\rho^{max}$  par sa valeur. nous avons d'après le Théorème 3,  $F_1 = \max(0, 10 - 19)$  et

$F_5 = \max(0, 16 - 19)$ , Alors les lois de commande sont données par l'inéquation suivante :

$$u(k) \geq e \cdot x_1(k-1) \oplus e \cdot x_5(k-1) \oplus e \cdot x_{10}(k-1) \tag{2.38}$$

Cette inéquation peut être simplifier. nous aurons alors :  $x_{10}(k-1) \geq x_1(k-1) \geq x_5(k-1)$ , puisque , la date de  $(k-1)^{em}$  franchissement de la transition  $t_{10}$  est plus grand que la date de  $(k-1)^{em}$  franchissement de la transition  $t_1$ , et la date  $(k-1)^{em}$  franchissement de la transition  $t_1$  est plus grande à la date de  $(k-1)^{em}$  franchissement de la transition  $t_5$ .

Finalement la lois de commande est donnée par l'équation suivante :

$$u(k) = e \cdot x_{10}(k-1) = 1 \cdot x_9(k-1).$$

Cette loi de commande peut être représentée par une place marquée et temporisée connectée au modèle de du l'AAR (place représentée par un double cercle sur la figure Fig.2.12).

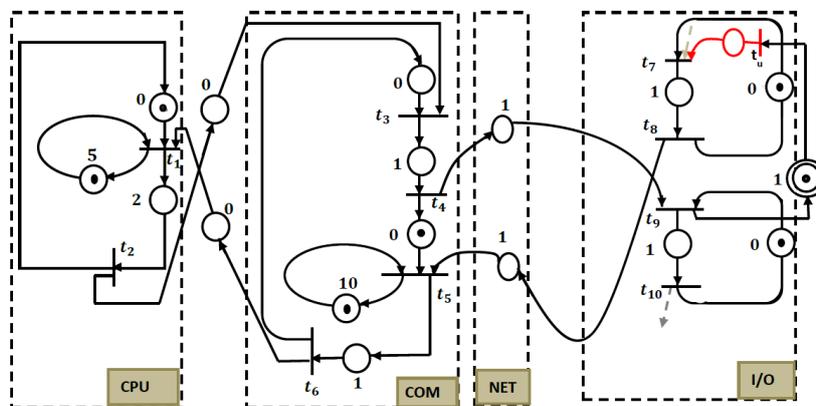


FIG. 2.12: Modèle GET du l'ARR avec la place du contrôle

### 2.3.3.3.3. E. Interprétation de résultats

La loi de commande calculée correspond à un retour d'état causal pour lequel la contrainte temporelle de l'AAR est satisfaite. Cette loi de commande implique un retard qui pourrait être représenté par une place marquée et temporisée à 1. Cette loi de commande va retarder la réception d'une autre requête au niveau de module d'entrée après l'envoi d'une requête

par une unité de temps. En pratique, cette loi de commande correspond à un contacteur temporisé réglé à une unité de temps.

## 2.4 Conclusion

Dans ce chapitre, nous avons présenté une approche formelle pour la synthèse de lois de commande en boucle fermée afin de satisfaire un ensemble de contraintes temporelles imposées sur les systèmes à événements discrets modélisés par des graphes d'événements temporisés (GETs). L'approche proposée est basée sur l'utilisation des GETs et de l'algèbre Max-Plus, où le comportement dynamique du GET est représenté par un système d'équations linéaires dans l'algèbre Max-Plus. Les contraintes temporelles imposées sur certains chemins du GET sont exprimées par un ensemble d'inéquations linéaires Max-Plus.

Nous avons obtenu une condition suffisante pour l'existence des lois de commande qui garantissent la satisfaction de ces contraintes. Les lois de commande que nous avons obtenue peuvent être représentées par des places de contrôle connectées au graphe initial.

Nous avons par la suite, appliqué les résultats obtenus sur une architecture d'automatisation en réseau AAR, afin de garantir la contrainte temporelle imposée sur leur temps de réponse. Les événements dans ces systèmes apparaissent à des moments particuliers et doivent être pris en compte dans un délai connu à l'avance. Mais les différents composants de ces AAR (contrôleurs, capteurs, actionneurs) échangent leurs données via un réseau de communication. Cela induit un délai considérable entre l'apparition d'un événement généré par le contrôleur et son impact sur le processus contrôlé. Ce qui impose certaines contraintes temporelles sur le temps de réponse du l'AAR. Les lois de commande calculées sont des feedbacks causales qui peuvent être représentées par des places de contrôle connectées au modèle du l'AAR.

# Chapitre 3

## Commande des graphes d'événements temporisés sous contraintes de marquages

### 3.1 Introduction

Dans le chapitre précédent, nous avons abordé le problème de la commande des systèmes à événements discrets sous contraintes de temps. Dans ce chapitre, nous intéressons toujours à la commande des systèmes à événements discrets temporisés, mais par rapport une autre catégorie de spécifications. Nous s'intéressons plus précisément au problème d'états interdits, autrement dit problème de contraintes de marquage.

Dans un réseau de Petri, l'état du système est représenté par le vecteur de marquage, et les états interdits sont des marquages atteignables, mais interdits par des spécifications imposées sur le système. Le problème consiste alors à empêcher l'occurrence d'un ensemble d'événements qui conduisent à ces états.

Dans ce chapitre, nous développerons une approche qui permet de synthétiser un contrôleur qui va agir en boucle fermée avec le système considéré afin de garantir des spécifications de marquage imposées sur un système à événements discret modélisé par une classe de réseau de Petri.

Ce problème de synthèse de superviseur pour les SEDs, dans l'objectif de garantir des spécifications imposées par un cahier de charges a été initiée par les travaux de Ramadge et Wonham [46, 45]. Leur approche est basée sur l'utilisation des modèles automates et des langages formels. L'inconvénient de cette technique est l'explosion combinatoire du nombre d'états souvent rencontré lorsqu'on a affaire à des SEDs.

Afin de résoudre ce problème, plusieurs méthodes basées sur les réseaux de Petri non temporisés ont été développées. Nous pouvons citer les travaux présentés dans [53], [24], [54], [47], [10], [26]. Dans [53, 24, 54], les auteurs ont proposé des approches basées sur les réseaux de Petri sauf pour synthétiser des contrôleurs en boucle fermée. Le but est d'assurer la satisfaction de contraintes sur le vecteur marquage. Les spécifications de marquage ont été exprimées par des inéquations linéaires. Les approches utilisent des invariants de marquage, cette technique permet de remplacer  $r$  contraintes constituées de  $n$  places par une seule contrainte constituée de  $(n-1)$  places. Les places de contrôle satisfaisant ces contraintes ont été calculées. Une autre méthode pour résoudre ce problème d'états interdits, dans les systèmes à événements discrets modélisés par des réseaux de Petri avec des transitions non observables, est proposée dans [47].

Dans [10, 26] deux autres méthodes ont été présentées, ces dernières permettent la synthèse des contrôleurs maximalement permissif pour les RdP avec des transitions incontrôlables. Ces méthodes sont basées sur la théorie des régions. Ces approches sont plus puissantes que les méthodes basées sur les invariants de marquage, dans le sens où elles permettent de trouver une solution de contrôle optimale (dans le cas où elle existe).

Les travaux que nous avons cités précédemment ne prennent pas en compte explicitement l'influence du temps. Ils considèrent que les événements peuvent avoir lieu dans un procédé à des moments indéterminés. Comme nous l'avons vu dans le deuxième chapitre, le temps dans les systèmes à événements discrets en général et dans les systèmes de production en particulier, est un paramètre très important. Malgré le grand rôle qui joue le temps dans ces systèmes, peu d'approches ont été développées pour la commande des systèmes à événements discrets temporisés sous contraintes de marquage.

Ces dernières années ont vu apparaître des travaux sur la commande des SEDs basées

sur les réseaux de Petri temporisés. Dans [6], les auteurs proposent une méthodologie de synthèse d'un contrôleur fondé sur des réseaux de Petri temporisés pour résoudre le problème des états de transitions interdites. Cette méthodologie est basée sur l'approche de Ramadge et Wonham et la théorie des régions. Elle permet de calculer un ensemble de places de contrôle à rajouter au réseau de Petri afin de résoudre le problème des états de transitions interdites.

Dans [9], un superviseur est introduit pour des graphes marqués temporisés pour résoudre un problème d'exclusion de marquage. Ces deux approches sont basées sur la théorie des graphes ce qui rend l'application de ces résultats dans des systèmes complexes compliqué et parfois impossible.

Dans cette partie de la thèse, nous abordons le problème de la synthèse du contrôleur en boucle fermée pour une classe de réseaux de Petri temporisés sous contraintes de marquage. L'objectif étant de limiter le nombre de jetons dans une ou plusieurs places dans un GET. Ces places peuvent représenter des stocks dans les systèmes industriels, la mémoire tampon dans les systèmes de communication ou des stations dans les réseaux de transport.

Nous avons considéré des SEDs modélisés par des graphes d'événements temporisés dont la dynamique est régit par des équations linéaires dans l'algèbre Min-plus [11, 12].

Avant de calculer des lois de commande pour satisfaire les contraintes de marquage imposées sur le système, nous traduisons tout d'abord ces contraintes par des inéquations linéaires dans l'algèbre Min-Plus. La méthode formelle que nous proposons dans ce chapitre repose sur la résolution d'un système d'équation et d'inéquations dans l'algèbre Min-Plus. Contrairement aux approches citées précédemment, dans notre travail le temps est pris en considération dans l'étape modélisation du système et dans l'étape synthèse du contrôleur.

Dans la première partie de ce chapitre, nous formulons le problème de commande des GETs sous contraintes de marquage dans l'algèbre Min-Plus. Nous montrons notamment comment écrire ces contraintes par des inéquations linéaires dans l'algèbre Min-Plus.

Dans la deuxième partie, nous présenterons d'une manière très détaillée notre approche

de commande des GETs sous contrainte de marquage. Nous considérons dans un premier temps le cas des graphes d'événements temporisés avec une seule transition d'entrée. Nous supposons que ce graphe est soumis à des contraintes de marquage imposées sur des places indépendantes. Puis, l'approche est adaptée au cas des graphes soumis à des contraintes de marquages imposées sur des chemins de GET.

Par la suite, nous avons étendu notre étude à la commande des GETs avec plusieurs transitions de commande, sous plusieurs contraintes de marquage.

## 3.2 Problème de contraintes de marquages (état interdit)

Nous définissons dans cette partie le problème de contrainte de marquage et nous montrons comment le traduire mathématiquement par des équations et des inéquations linéaires dans l'algèbre Min-Plus.

### 3.2.1 Formulation du problème de commande

Comme nous l'avons déjà mentionné dans le premier chapitre, l'état d'un RDP temporisé en général et d'un GET en particulier indique le nombre de jetons dans chacune de ses places à un moment donné.

Une contrainte sur le marquage dans un RDP consiste à imposer une borne maximale à une place ou un ensemble de places dépendantes ou indépendantes.

Pour bien expliquer le problème de contraintes de marquage, nous considérons l'exemple d'un système de production modélisé par le GET de la Fig.3.1.

Ce système est composé de deux machines  $M_1$  et  $M_2$  avec un stock intermédiaire  $B$ , limité en pièce. La transition  $t_1$  (resp.  $t_3$ ) indique le début du traitement sur la machine  $M_1$  (resp.  $M_2$ ). L'événement de la fin du traitement dans  $M_1$  (resp.  $M_2$ ) est représenté par la transition  $t_2$  (resp.  $t_4$ ). Le franchissement de la transition  $t_2$  (resp.  $t_3$ ) désigne également qu'une pièce est entrée (sortie) au/de stock  $B$ . La transition de commande  $t_u$  est la seule transition contrôlable dans ce graphe. Le stock  $B$  est limité à 2 pièces.

Nous remarquons que le temps de cycle de  $M_2$  est supérieur au cycle de  $M_1$ , ce qui

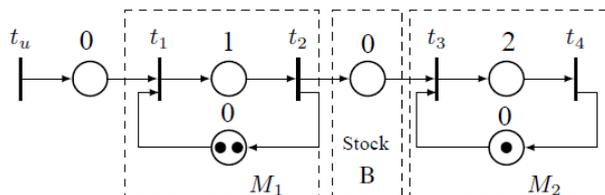


FIG. 3.1: Modèle d'une ligne de production

conduit au dépassement d'un nombre maximal de pièces dans le stock  $B$ .

Dans le cas où la transition  $t_2$  est commandable, la solution de ce problème est intuitive, similairement à ce qui se fait dans les réseaux de Petri en général. Il suffit d'ajouter la place  $p_{23}$  marquée par 2 jetons, afin d'assurer que le nombre de jetons dans la place  $p_{32}$  ne dépasse pas 2 jetons (voir la Fig3.2). La résolution du problème dans le cas où la

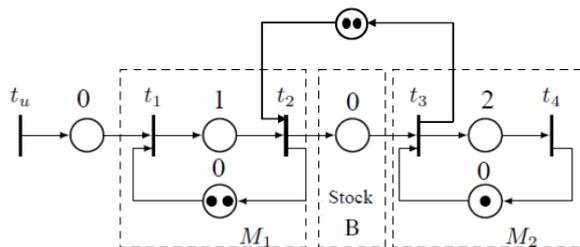


FIG. 3.2: Modèle d'une ligne de production avec un stock limité à 2

transition d'entrée de la place sous contrainte est incontrôlable n'est pas évidente. Il faut donc disposer d'autres techniques pour garantir ces contraintes de marquage.

Dans ce chapitre, nous traitons la question comme un problème de contrôle, et nous déterminons des lois de commande garantissant la satisfaction des contraintes de marquage imposées sur des places d'un GET. Nous supposons que les transitions d'entrées de ce graphe sont les seules transitions contrôlables. Notre cherchons des lois de commande pour garantir le respect des contraintes de marquage imposées sur ce graphe. Dans un premier temps, nous montrons comment exprimer ces contraintes par des inéquations linéaires dans l'algèbre Min-Plus, ensuite, nous définissons des conditions d'existence des lois de commande.

### 3.2.1. Contraintes de marquage

Nous considérons un SED modélisé par un graphe d'événements temporisé. Nous supposons qu'une place  $p_{ij}$  de ce graphe est soumise à une contrainte de marquage (ne peut pas contenir plus de  $b$  jetons au long de fonctionnement du système). Nous notons par  $M_{ij}(t)$  le nombre de jetons dans la place  $p_{ij}$  à l'instant  $t$  et la contrainte de marquage peut s'exprimer par l'inéquation suivante :

$$M_{ij}(t) \leq b \quad (3.1)$$

Nous associons à chaque transition du graphe un compteur, et nous notons par  $x_i(t)$  et  $x_j(t)$  les compteurs associés respectivement aux transitions  $t_i$  et  $t_j$ . Par définition, le compteur  $x_j(t)$  indique le nombre de franchissements, depuis l'instant initial, de la transition  $t_j$  à l'instant  $t$ . Autrement dit, il représente le nombre de jetons entrants à la place  $p_{ij}$  à l'instant  $t$ . Le compteur  $x_i(t)$  représente le nombre de jetons sortis de la place  $p_{ij}$  à l'instant  $t$ .

D'après ces définitions, le nombre de jetons dans la place  $p_{ij}$  peut s'exprimer en fonction de ces deux compteurs par la formule suivante :

$$M_{ij}(t) = x_j(t) - x_i(t) + M_{ij0}$$

avec  $M_{ij0}$  représente le marquage initial de la place  $p_{ij}$ .

En remplaçant  $M_{ij}(t)$  par son expression donnée précédemment, l'inéquation (3.1) peut s'écrire comme par :

$$x_j(t) - x_i(t) + M_{ij0} \leq b \quad (3.2)$$

ce qui donne :

$$x_j(t) \leq (b - M_{ij0}) \cdot x_i(t) \quad (3.3)$$

tel que  $(\cdot)$  représente l'addition usuel  $((+))$  dans l'algèbre Min-Plus.

Cette inéquation (3.3) correspond à la contrainte de marquage à satisfaire.

## 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

Nous proposons dans cette section, une approche formelle basée sur l'utilisation de l'algèbre Min-Plus pour le calcul des lois de commande pour garantir des bornes maximales des jetons dans des places d'un GET. Comme nous l'avons indiqué dans le premier chapitre, le comportement d'un GET peut être représenté par une équation linéaire dans l'algèbre Min-Plus. Dans ce qui suit, nous considérons que le comportement dynamique d'un GET est décrit par l'équation d'état suivante :

$$x(t) = A \cdot x(t-1) \oplus B \cdot u(t), \quad (3.4)$$

Où :  $x \in \bar{\mathbb{R}}_{min}^N$ ,  $A \in \bar{\mathbb{R}}_{min}^{N \times N}$ ,  $B \in \bar{\mathbb{R}}_{min}^{N \times m}$ ,  $u \in \bar{\mathbb{R}}_{min}^{\bar{n}}$ .

Les deux entiers  $N$  et  $m$  correspondent respectivement au nombre de transitions internes et au nombre de transitions de commande.

Notre approche est basée sur la résolution d'un système d'équations et d'inéquations dans l'algèbre Min-Plus. Nous recherchons un contrôleur sous la forme de  $u(t) = F \cdot u(t-1)$  garantissant la satisfaction des contraintes de marquage.

Dans la suite de ce chapitre, nous considérons les hypothèses suivantes :

**Hypothèse 3.** *Il existe au moins un chemin qui relie la transition de commande à la transition d'entrée de la place où du chemin sous contrainte.*

**Hypothèse 4.** *Chaque transition du graphe d'événements temporisé est franchissable dès qu'elle est validée (franchissement au plus tôt).*

### 3.3.1. Commande monovariante

Nous considérons un graphe d'événements temporisé avec une seule transition d'entrée et soumis à des contraintes de marquage. Notre objectif est de calculer une loi de commande sous la forme de  $u(t) = F \cdot u(t-1)$ . Où  $u(t)$  est un scalaire ( $u(t) \in \bar{\mathbb{R}}_{min}^{N \times 1}$ ) et  $F$  est un vecteur ligne avec  $N$  composantes qui sont soit  $F_r = \varepsilon$  où  $F_r \geq 0$ . Les composantes de ce vecteur correspondent aux marquages des places de contrôle que nous voulons déterminer

pour assurer les spécifications de marquage désirées.

Les spécifications de marquage désirées peuvent être représentées par des inéquations linéaires dans l'algèbre Min-Plus, comme démontré dans la section précédente. Dans un premier temps, nous considérons des contraintes de marquage imposées sur des places d'un graphe d'événements temporisé. Par la suite, nous considérons le cas où les contraintes de marquage sont imposées sur certains chemins (succession de places) d'un graphe d'événements temporisé. Dans les deux cas, nous entamerons tout d'abord le cas simple, où le GET est soumis à une seule contrainte de marquage, après nous allons généraliser notre étude pour le cas d'un GET soumis à plusieurs contraintes de marquage.

### 3.3.1.1. Contraintes de marquage imposées sur des places

#### a) Une seule contrainte

Nous considérons un graphe d'événements temporisé  $G$  soumis à une seule contrainte de marquage imposée sur une place notée  $p_{ij}$ . Les spécifications de marquage imposées sur ce graphe consiste à assurer que le nombre de jetons dans la place  $p_{ij}$  ne dépasse pas le nombre  $b$ . Cette contrainte peut être exprimée par l'inéquation suivante :

$$x_j(t) \leq (b - M_{ij0}) \cdot x_i(t) \quad (3.5)$$

où,  $x_j(t)$  et  $x_i(t)$  représente respectivement les compteurs associés au deux transitions  $t_j$  et  $t_i$  et indiquant respectivement le nombre de jetons entrant et le nombre de jetons sortant de la  $p_{ij}$ .

Comme indiqué à la propriété 1 de la section 1.3, pour  $\tau = \phi$ , tel que  $\phi$  est un entier avec  $\phi \geq 1$ , l'équation d'état (1.11) peut s'étendre à l'expression suivante :

$$x(t) = A^\phi \cdot x(t - \phi) \oplus \left[ \bigoplus_{k=0}^{\phi-1} (A^k \cdot B) \cdot u(t - k) \right], \quad (3.6)$$

La  $i^{\text{ème}}$  composante de  $x(t)$ , est donnée par :

$$x_i(t) = \left[ \bigoplus_{r=1}^n A_{ir}^\phi \cdot x_r(t - \phi) \right] \oplus \left[ \bigoplus_{k=0}^{\phi-1} (A^k \cdot B)_i \cdot u(t - k) \right], \quad (3.7)$$

tel que,  $\phi \geq 1$ ,  $A \in \bar{\mathbb{R}}_{min}^{n \times n}$ ,  $n$  représente le nombre de transitions internes du graphe  $G$  et  $A_{ir}^\phi$  indique le  $i^{\text{th}}$  ligne de la matrice  $A^\phi$ .

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

Du fait qu'il existe un chemin  $\alpha$  qui relie la transition  $t_u$  à la transition  $t_j$ , nous avons alors l'inégalité suivante :

$$x_j(t) \leq (A^{\tau_\alpha} \cdot \mathbf{B})_j \cdot u(t - \tau_\alpha), \quad (3.8)$$

avec  $\tau_\alpha$  représente la somme des temporisations de toutes les places qui appartiennent au chemin  $\alpha$ .

En prenant en considération l'hypothèse 4(franchissement au plus tôt), et en remplaçant les deux fonctions,  $x_i(t)$  et  $x_j(t)$  par leurs expressions données respectivement dans les deux équations (3.7) et (3.8) dans l'inéquation de contrainte de marquage (3.5), on obtient cette inéquation :

$$(A^{\tau_\alpha} \cdot \mathbf{B})_j \cdot u(t - \tau_\alpha) \leq (b - M_{ij0}) \cdot \left[ \bigoplus_{r=1}^n A_{ir}^\phi \cdot x_r(t - \phi) \right] \oplus \left[ \bigoplus_{k=0}^{\phi-1} (A^k \cdot \mathbf{B})_i \cdot u(t - k) \right] \quad (3.9)$$

La satisfaction de l'inéquation (3.9) revient à vérifier les deux inéquations suivantes :

$$(A^{\tau_\alpha} \cdot B)_j \cdot u(t - \tau_\alpha) \leq (b - M_{ij0}) \left[ \bigoplus_{k=0}^{\phi-1} (A^k \cdot B)_i \cdot u(t - k) \right] \quad (3.10)$$

et

$$(A^{\tau_\alpha} \cdot B)_j \cdot u(t - \tau_\alpha) \leq (b - M_{ij0}) [A_{ir}^\phi \cdot x_r(t - \phi)] \quad (3.11)$$

Ces deux inéquations peuvent s'écrire sous les formes suivantes :

$$(A^{\tau_\alpha} \cdot B)_j \cdot u(t - \tau_\alpha) \leq (b - M_{ij0}) \left[ \bigoplus_{k=0}^{\phi-1} (A^k \cdot B)_i \cdot u(t - k) \right] \quad (3.12)$$

et

$$u(t - \tau_\alpha) \leq (b - M_{ij0} - (A^{\tau_\alpha} \cdot B)_j) \cdot [A_{ir}^\phi \cdot x_r(t - \phi)] \quad (3.13)$$

Dans le Théorème qui suit, le contrôleur  $u(t)$  qui assure la satisfaction de la contrainte de marquage (3.5) est explicitement donné.

**Théorème 7.** *Nous considérons un graphe d'événements temporisé  $\mathcal{G}$  dont l'évolution est représenté par l'équation d'état (1.3). Ce graphe est soumis à une seule contrainte de marquage de la forme(3.5).*

*Pour  $\phi = \tau_\alpha + 1$  la loi de commande donnée par :*

$$u(t) = \bigoplus_{r=1}^N F_r \cdot x_r(t - 1) \quad (3.14)$$

tel que  $F_r = \max(0, (b - M_{ij0} - (A^{\tau_\alpha} \cdot \mathbf{B})_j) \cdot A_{ir}^{\tau_\alpha+1})$ ,

assure la satisfaction de la contrainte (3.5) si la condition suivante est respectée :

$$(b - M_{ij0}) \cdot (A^k \cdot \mathbf{B})_i \geq (A^{\tau_\alpha} \cdot \mathbf{B})_j, \quad k = 0 \cdots \tau_\alpha, \quad (3.15)$$

**Démonstaration.** Nous cherchons des lois de commande sous la forme de  $u(t) = F \cdot x(t - 1)$  garantissant la contrainte de marquage (3.5).

De l'inéquation (3.12), nous tirons  $u(t)$  en fonction de  $x(t - 1)$  si seulement si  $\phi = \tau_\alpha + 1$ .

En remplaçant  $\phi$  par  $\tau_\alpha + 1$  dans les deux inéquations (3.12) et (3.13), nous obtenons les deux nouvelles inéquations :

$$(A^{\tau_\alpha} \cdot \mathbf{B})_j \cdot u(t - \tau_\alpha) \leq (b - M_{ij0}) \left[ \bigoplus_{k=0}^{\tau_\alpha} (A^k \cdot \mathbf{B})_i \cdot u(t - k) \right] \quad (3.16)$$

et

$$u(t) \leq (b - M_{ij0} - (A^{\tau_\alpha} \cdot \mathbf{B})_j) \cdot [A_{ir}^\phi \cdot x_r(t - 1)] \quad (3.17)$$

La fonction  $u(t)$  qui représente le nombre de franchissement de la transition  $t_u$  à l'instant  $t$  est une fonction croissante, c'est-à-dire :  $u(t) \geq u(t - 1) \geq u(t - 2) \cdots \geq u(t - n)$ . En prenant en considération cette remarque, l'inégalité (3.16) peut s'exprimer sous la forme qui suit :

$$(A^{\tau_\alpha} \cdot \mathbf{B})_j \leq (b - M_{ij0}) (A^k \cdot \mathbf{B})_i \quad (3.18)$$

Pour  $k = 0$  à  $\tau_\alpha$

L'inéquation (3.17) traduit explicitement les lois de commande données par le Théorème 7, garantissant la satisfaction de la contrainte de marquage (3.5) dans le cas où condition (3.18) est vérifiée. □

Les feedbacks donnés par le Théorème 7, assurent la satisfaction de toutes les contraintes de marquage imposées sur un graphe d'événements temporisé et qui peuvent être formulés sous la forme de l'inéquation (3.5). Ces feedbacks peuvent être représentés par des places marquées et temporisées connectées au graphe initial.

### b) Plusieurs contraintes

Nous considérons maintenant un graphe d'événements temporisé  $G$ , comportant une seule transition de commande  $t_u$ , et  $S$  contraintes de marquage. Les places sous contraintes sont

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

notées  $p_s$ , pour  $s = 1$  à  $s = S$ . Pour chaque place  $p_s$ , nous notons par  $M_{s0}$  et  $\tau_s$  respectivement le marquage initial et la temporisation de la place  $p_s$  et par  $b_s$  le nombre maximal de jetons à ne pas dépasser dans chaque place  $p_s$ . Les transitions  $t_j$  et  $t_i$  sont respectivement la transitions d'entrée et la transition de sortie de la place  $p_s$ . Les fonctions  $x_j(t)$  et  $x_i(t)$  définissent les compteurs correspondants.

Les contraintes de marquage sont représentées dans l'algèbre Min-Plus par l'inéquation suivante :

$$x_j(t) \leq (b_s - M_{s0}) \cdot x_i(t) \quad (3.19)$$

avec  $s$  allant de 1 à  $S$ .

Nous supposons qu'il existe au moins un chemin, noté  $\alpha_s$ , de la transition de commande  $t_u$  vers la transition  $t_j$ . Toutes les transitions  $t_j$  sont alors contrôlables. Pour chaque contrainte  $p_s$ , nous pouvons appliquer les résultats du Théorème 7.

Si la condition de Théorème 7 est satisfaite pour chaque contrainte  $s$ , alors la loi de commande  $u_s(t)$  satisfait la sème contrainte.

D'une manière générale, les lois de commande sont données par l'équation suivante :

$$u_s(t) = F_{r_s} \cdot x_r(t - 1)$$

Avec  $F_{r_s} = \bigoplus_{r=1}^N ((b_s - M_{s0} - (A^{\tau_{\alpha_s}} \cdot B)_j) \cdot A_{ir}^{\tau_{\alpha_s} + 1})$ .

Le contrôleur qui garantit la satisfaction de toutes les contraintes de marquage imposées sur le système correspond au minimal des contrôleurs calculé. Ces résultats ont été résumés dans le Théorème 8 suivant :

**Théoreme 8.** *Soit  $\mathcal{G}$  un graphe d'événements temporisé. L'évolution de ce graphe est donnée par l'équation (1.3). Ce graphe est soumis à  $S$  contraintes de marquage sous la forme (3.19).*

*La loi de commande exprimée par :*

$$u(t) = \bigoplus_{s=1}^S u_s(t),$$

*avec*

$$u_s(t) = F_{r_s} \cdot x_r(t - 1)$$

et  $F_{r_s} = \max(0, \oplus_{r=1}^N ((b_s - M_{s_0} - (A^{\tau_{\alpha_s}} \cdot B)_j) \cdot A_{ir}^{\tau_{\alpha_s}+1}))$  pour  $s = 1, \dots, S$ ,

assure la satisfaction des spécifications (3.19) si la condition suivante est bien respectée :

$$(b_s - M_{s_0}) \cdot (A^k \cdot B)_i \geq (A^{\tau_{\alpha}} \cdot B)_j, \quad k = 0 \quad \tau_{\alpha}, \quad (3.20)$$

**Démonstaration.** Comme nous l'avons indiqué dans le Théorème 7, le contrôleur (3.14) garantit le respect de la contrainte de marquage (3.19). Il est donc clair que pour  $S$  contraintes de marquage, la loi de commande  $u_s(t)$  garantit la  $S$ ème contrainte si la condition (3.20) est satisfaite.

Finalement, le feedback qui garantit les  $s$  contraintes (3.19) est donné par :

$$u(t) = \oplus_{s=1}^S u_s(t) \text{ pour } s=1 \text{ à } S.$$

Ce feedback correspond au minimum des contrôleurs calculés pour les  $s$  contraintes.  $\square$

Nous illustrons maintenant les résultats par un exemple d'un système manufacturier qui comporte un stock borné.

**Exemple 8.** Nous considérons une ligne de production modélisée par le graphe d'événements temporisé représenté dans la Fig.3.3. Ce système est composé de deux machines  $M_1$  et  $M_2$  avec un stock intermédiaire  $B$ . La transition  $t_1$  (resp.  $t_3$ ) indique le début du traitement sur la machine  $M_1$  (resp.  $M_2$ ). L'événement de fin de traitement dans  $M_1$  (resp.  $M_2$ ) est représenté par la transition  $t_2$  (resp.  $t_4$ ). Le franchissement de la transition  $t_2$  (resp.  $t_3$ ) signifie qu'une pièce est entrée (sortie) au/de stock  $B$ . La transition de commande  $t_u$  est la seule transition contrôlable dans ce graphe. Le stock  $B$  est borné à 2 pièces.

Le graphe de marquage de ce réseau de Petri est donné par la Fig.3.4. Nous supposons que la place  $p_{1u}$  comporte toujours le nombre de jetons nécessaire pour le franchissement de la transition  $t_1$ . Comme montré par le graphe de marquage, le nombre de jetons dans le stock  $B$  dépasse le nombre autorisé  $(\mu_4, \mu_5, \mu_6, \dots)$ .

On veut synthétiser une loi de commande  $u(t)$  pour que le nombre de jetons dans la place  $p_{32}$  qui modélise le stock  $B$ , ne dépasse pas deux jetons. Pour cela, nous appliquons les résultats du Théorème 7. Pour construire le modèle d'état du graphe de la Fig.3.3 il faut étendre le graphe de cette figure pour avoir un GET équivalent, avec  $\tau_{ij} = 0$  ou 1. Nous décomposons ainsi, la place  $p_{43}$  en deux places, notées respectivement par  $p_{53}$  et  $p_{45}$

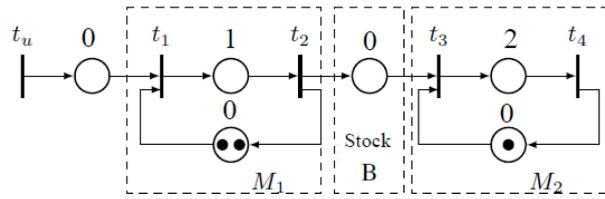


FIG. 3.3: modèle d'une ligne de production

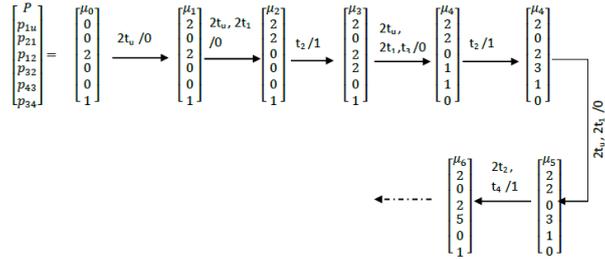


FIG. 3.4: Graphe de marquage de la Fig.3.3

et temporisées à 1 unité de temps. Le graphe étendu est représenté dans la Fig.3.5.

L'équation d'état du graphe de la Fig.3.5 est exprimée comme suite :

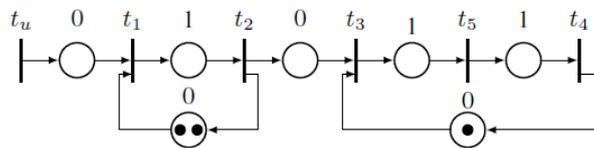


FIG. 3.5: Graphe étendu de la Fig.3.3

$$\mathbf{x}(t) = \begin{bmatrix} 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix} \cdot \mathbf{x}(t-1) \oplus \begin{bmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \cdot u(t)$$

Nous appliquons le Théorème 7 pour calculer la loi de commande qui garantira les spécifications de marquage imposées sur le stock B. Le marquage initial de la place  $p_{43}$  est nul ( $M_{430} = 0$ ). Il existe un chemin  $\alpha$  qui relie la transition  $t_u$  à la transition  $t_2$ , avec une temporisation de  $\tau_\alpha = 1$ , ( $\phi = \tau_\alpha + 1 = 2$ ). Nous avons aussi,  $(A^k \cdot \mathbf{B})_4 = \varepsilon, e$  pour  $k = 0$  à 1,  $(A^{\tau_\alpha} \cdot \mathbf{B})_j = 2$ , et  $(A_{3r}^2 = [2, \varepsilon, 1, \varepsilon, \varepsilon])$ .

La condition (3.20) du Théorème 7 est donnée par :

$$2 \cdot (A^k \cdot B)_3 \geq (A^2 \cdot \mathbf{B})_2$$

pour  $k = 0$  à  $1$ .

On remarque que la condition du Théorème 7 est bien vérifiée. Il existe donc des lois de commande qui permettent de garantir la contrainte de marquage imposée sur le stock  $B$ .

Ce contrôleur est donné par l'équation suivante :

$$u(t) = (2 - 0 - (A^1 \cdot \mathbf{B})_2) \cdot \left[ \bigoplus_{r=1}^5 A_{3r}^2 \cdot x_r(t-1) \right]$$

qui peut être exprimé par :

$$u(t) = 4 \cdot x_1(t-1) \oplus 3 \cdot x_3(t-1)$$

On peut simplifier cette expression en comparant le nombre de franchissement de la transition  $t_1$  à l'instant  $t$  est plus grand que le nombre de franchissement de la transition  $t_3$ , alors nous pouvons écrire :

$$x_1(t) \geq x_3(t) \implies x_1(t-1) \geq x_3(t-1).$$

Ce qui donne

$$4 \cdot x_1(t-1) \geq 3 \cdot x_3(t-1)$$

En conclusion, la loi de commande  $u(t)$  qui assure le respect de la contrainte est :

$$u(t) = 3 \cdot x_3(t-1)$$

Cette loi peut être représentée par une place de contrôle marquée de 3 jetons et temporisée par 1 unité de temps. Cette place de contrôle relie la transition  $t_3$  du GET à la transition de commande  $t_u$  (place en double cercle Fig.3.6).

La Fig.3.7 représente l'évolution du graphe de marquage de modèle de la ligne de production après l'intégration de la place de contrôle au réseau de Petri initial. Nous pouvons facilement vérifier que dans le système bouclé, le nombre de jetons dans le stock  $B$  est bien respecté, et que le graphe est vivant et sans blocage.

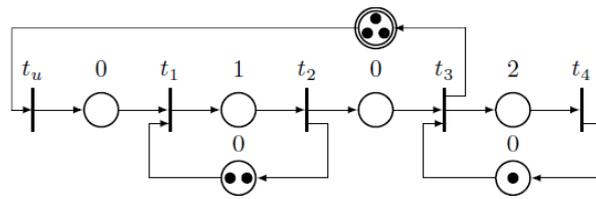


FIG. 3.6: Modèle de ligne de production avec la place de contrôle

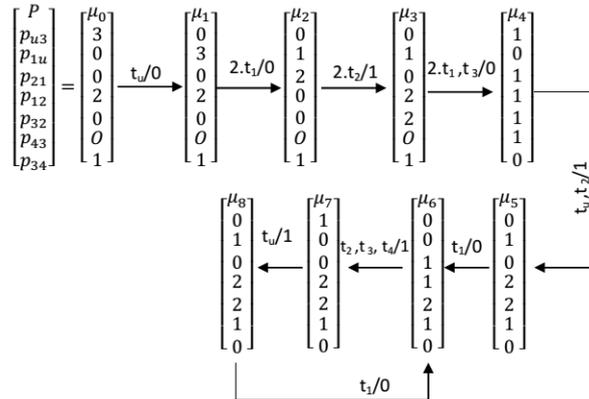


FIG. 3.7: Graphe de marquage du modèle de la Fig.3.6

### 3.3.1.2. Contraintes imposées sur des chemins d'un GET

Dans cette section, nous considérons le problème de contraintes de marquage imposées sur un ensemble de places dépendantes. Dans ce cas, le problème consiste à limiter le nombre de jetons dans certains chemins (succession de places) d'un graphe d'événements temporisé. L'objectif étant de synthétiser une loi de commande sous la forme de  $u(t) = F \cdot x(t - 1)$ , afin de satisfaire ces contraintes. Le même principe serait utilisé comme précédemment, la contrainte imposée sur un chemin d'un GET peut être représentée par une simple inéquation linéaire dans l'algèbre Min-Plus. Cette inéquation est donnée en fonction du marquage initial de ce chemin ainsi que les compteurs associés à sa transition d'entrée et sa transition de sortie.

Nous considérons par exemple, le GET représenté à la Fig.3.8.

Nous supposons que ce graphe est soumis à une contrainte de marquage imposé sur le chemin  $\rho$  qui relie la transition  $t_j$  à la transition  $t_i$ . La contrainte imposée sur ce chemin est donnée par l'inéquation suivante :

$$M_{qj}(t) + M_{kq}(t) + M_{ik}(t) \leq b, \quad (3.21)$$

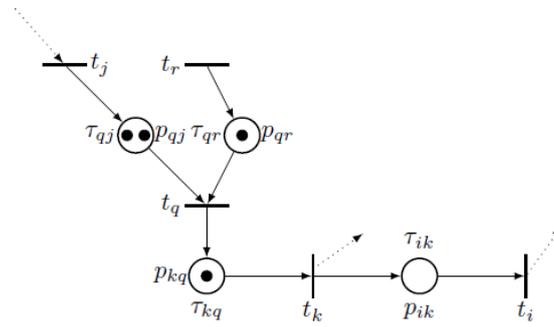


FIG. 3.8: Succession de places d'un GET

avec  $M_{qj}(t)$ ,  $M_{kq}(t)$ ,  $M_{ik}(t)$  représentent respectivement les marquages des places  $p_{qj}$ ,  $p_{kq}, p_{ik}$  à l'instant  $t$  et  $b$  étant un entier non négatif représente le nombre de jetons maximal à ne pas dépasser dans le chemin  $\rho$ .

Nous associons à chaque transition de ce graphe une fonction compteur. Cette contrainte peut être exprimée par cette inéquation :

$$(x_j(t) - x_q(t) + 2) + (x_q(t) - x_k(t) + 1) + (x_k(t) - x_i(t)) \leq b.$$

Après simplification, cette inéquation devient :

$$x_j(t) - x_i(t) \leq (b - 3),$$

D'une manière générale, l'inéquation de contrainte peut être décrite comme suit :

$$x_j(t) \leq (b - M_{\rho_0}) \cdot x_i(t) \tag{3.22}$$

avec  $M_{\rho_0} = 3$  est le marquage initial du chemin sous contrainte  $\rho$ .

### a) Un seul chemin sous contrainte

Nous considérons un graphe d'événement temporisé avec une transition d'entrée  $t_u$  qui est la transition de commande. Ce graphe est soumis à une contrainte de marquage imposée sur un de ces chemins noté  $\rho$ . Les transitions  $t_j$  et  $t_i$  représentent respectivement la transition d'entrée et la transition de sortie du chemin. Les fonctions  $x_j(t)$  et  $x_i(t)$  définissent les compteurs correspondants.

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

La contrainte de marquage imposée sur le chemin  $\rho$  est représentée par une inéquation de la forme (3.22) .

Nous supposons qu'il existe au moins un chemin  $\alpha$  reliant la transition de commande  $t_u$  à la transition  $t_j$  (Hypothèse 3),

La même démarche appliquée dans le cas d'un graphe d'événements temporisé sous contrainte de marquage imposée sur une place sera suivie pour calculer les lois de commande qui garantissent la satisfaction de la contrainte (3.22).

En remplaçant le marquage initial de la place sous contrainte par le marquage initial de chemin  $\rho$ , dans la section 3.3.1.1, nous remarquons que la formule des deux contraintes est la même. Nous pouvons alors appliquer les résultats obtenus pour la contrainte de marquage imposée sur une place pour résoudre le problème de contrainte de marquage imposée sur un chemin d'un GET. Avec quelques modifications dans certains paramètres qui seront indiqués par la suite.

L'objectif du Théorème 9 est de définir les lois de commande qui assurent le respect des spécifications de marquage imposées sur un chemin d'un GET.

**Théorème 9.** *L'équation :*

$$u(t) = F_r \cdot x_r(t - 1) \quad (3.23)$$

avec  $F_r = \max(0, (b - M_{\rho_0} - (A^{\tau_\alpha} \cdot B)_j) \cdot A_{ir}^\phi)$  et  $\phi = \tau_\alpha + 1$ , assure la satisfaction de la contrainte 3.21 si la condition suivante :

$$(A^{\tau_\alpha} \cdot B)_j \leq (b - M_{\rho_0}) \cdot (A^k \cdot B)_i, \quad k = 0, \dots, \phi - 1. \quad (3.24)$$

*est satisfaite.*

**Démonstration.** Comme nous l'avons démontré dans la section 3.3.1.1, le contrôleur donné par la formule (3.14) satisfait la contrainte de marquage imposée sur une place d'un GET, dans le cas où la condition (3.20) est satisfaite. Après simplification de la contrainte de marquage imposée sur un chemin d'un GET donnée par l'inéquation (3.21). Nous pouvons facilement remarquer que l'inéquation résultante (3.22) est la même que celle de la contrainte imposée sur une place (3.5) en remplaçant le marquage initial de

la place  $p_{ij}$  noté  $M_{ij0}$  par le marquage initial du chemin  $\rho$  donné par  $M_{\rho 0}$ . En suivant la même procédure de calcul des lois de commande dans le cas de contrainte imposée sur une place, nous trouvons toujours les mêmes résultats en remplaçant le marquage de la place  $p_{ij}$  par le marquage du chemin  $\rho$ .

Finalement, les lois de commande données par l'équation (3.23) garantissent la satisfaction de la contrainte de marquage (3.21) si la condition (3.24) est satisfaite.  $\square$

**Exemple 9.** Soit un système manufacturier constitué de deux machines  $M_1$  et  $M_2$  qui travaillent en série et un robot comme un moyen de transport (voir Fig.3.9). Les deux machines  $M_1$  et  $M_2$  ont une capacité d'usinage d'une seule pièce à la fois. La machine  $M_1$  est décomposée en deux parties ; une partie thermique et une zone de refroidissement. Ces deux parties sont modélisées respectivement par les deux places  $p_{21}$  et  $p_{32}$  dans la Fig.3.10. Fig.3.10 représente le modèle graphe d'événements temporisé du système précédent. La machine  $M_2$  est modélisé par la place  $p_{75}$  et le robot par la place  $p_{43}$ , le robot peut transporter une pièce à la fois. Les temps d'exécutions des tâches par les deux machines et le temps de transport de la pièce par le robot sont associés à chaque place modélisant l'une de ces entités.

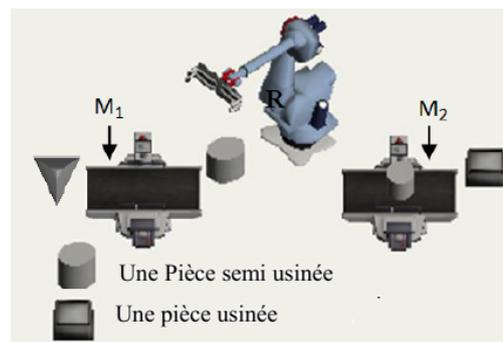


FIG. 3.9: Système manufacturier (deux machines et un robot)

Le cahier des charges de ce système indique que le robot ne doit pas être en attente pour poser la pièce dans la machine  $M_2$ . Ce qui signifie que les deux places  $p_{43}$  et  $p_{54}$  ne doivent pas être toutes les deux marquée à la fois. Cet état est atteignable au cours de fonctionnement de système, ce qui est indiqué par le graphe de marquage du RDP illustré

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

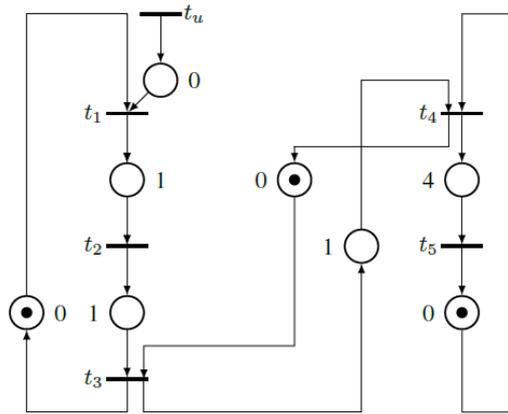


FIG. 3.10: Modèle GET de système de la Fig.3.9

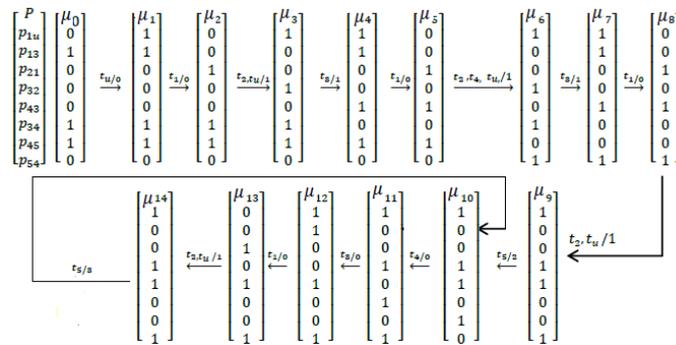


FIG. 3.11: Graphe de marquage de Modèle GET de la Fig.3.10

par la Fig.3.11 (état  $\mu_7, \mu_8, \mu_9, \mu_{12}, \mu_{13}, \mu_{14}$ ). Nous pouvons écrire ces spécifications par l'inéquation suivante :

$$M_{43} + M_{54} \leq 1.$$

Comme indiqué précédemment, cette contrainte peut être représentée par une inéquation linéaire dans l'algèbre Min-Plus, sous la forme :

$$(x_3(t) - x_4(t)) + (x_4(t) - x_5(t)) \leq 1.$$

En simplifiant cette inéquation, l'inéquation de contrainte devient :

$$(x_3(t) - x_5(t)) \leq 1 \tag{3.25}$$

On veut synthétiser des lois de commande qui garantissent la spécification de marquage imposées à ce système. Pour cela, nous commençons tout d'abord par la représentation

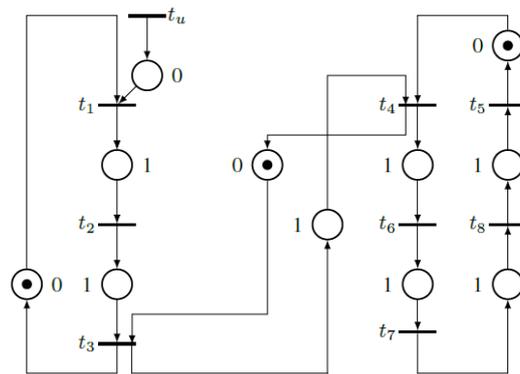


FIG. 3.12: Modèle GET de système de la figure

de la dynamique du système par une équation d'état. La temporisation de la place  $p_{54}$  est supérieure à 1, alors on décompose cette place en 4 places temporisées à 1. Nous ajoutons 3 transitions intermédiaires, comme montré dans la Fig.3.12 L'équation d'état associé au graphe d'événements temporisé de la Fig.3.12 est donnée par l'équation :

$$x(t) = \begin{bmatrix} \varepsilon & 1 & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ e & \varepsilon \\ \varepsilon & e & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix} \cdot x(t-1) \oplus \begin{bmatrix} e \\ \varepsilon \end{bmatrix} \cdot u(t)$$

Le marquage initial du chemin  $\rho$  est nul ( $M_{\rho 0} = 0$ ), la temporisation du chemin  $\alpha$  reliant la transition de commande  $t_u$  à la transition  $t_2$  est donnée par  $\tau_\alpha = 2$ , alors on déduit que  $\phi = \tau_\alpha + 1 = 3$ .

Afin de vérifier la condition du Théorème 9, nous calculons, tout d'abord  $A^2$ .

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

$$A^2 = \begin{bmatrix} \varepsilon & 1 & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & 2 & 3 \\ \varepsilon & e & 1 & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & e & \varepsilon \end{bmatrix}$$

On peut facilement vérifier que la condition du Théorème 9 est bien vérifiée :

$$(A \cdot B)_3 \leq 1 \cdot (A^k \cdot B)_5, \text{ pour tout } k \text{ allant de } 0 \text{ à } 2.$$

Alors il existe un contrôleur qui garantit le respect des spécifications de marquage imposées sur le système précédent. D'après le Théorème 9, ce contrôleur est donné par la formule suivante :

$$\begin{aligned} u(t) &\leq \bigoplus_{r=1}^8 (1 - 0) \cdot A_{5r}^1 - (B)_3 \cdot x_r(t - 1), \\ &= 1 \cdot x_6(t - 1) = 1 \cdot x_4(t - 2) \end{aligned}$$

Le feedback calculé peut être représenté par une place de contrôle marquée, temporisée et connectée au modèle de graphe d'événements temporisé du système manufacturier (place représentée par un double cercle dans la Fig.3.13). Ce contrôleur assure la satisfaction de la contrainte de marquage imposée sur le système, cela est bien vérifiée par le graphe de marquage illustré sur la Fig.3.14

#### b) Plusieurs chemins sous contraintes de marquage

Nous considérons maintenant un graphe d'événements temporisé  $G$  toujours avec une seule transition de commande  $t_u$ . Ce graphe est soumis à  $S$  contraintes de marquage imposées sur  $S$  chemins notés  $\rho_s$ , avec  $s = 1$  à  $S$ . Pour chaque chemin  $\rho_s$ , on note par  $M_{\rho_s 0}$  et  $\tau_{\rho_s}$  respectivement le marquage initial et la temporisation du chemin  $\rho_s$ . Les transitions  $t_j$  et  $t_i$  représentent respectivement la transition d'entrée et la transition de sortie du chemin

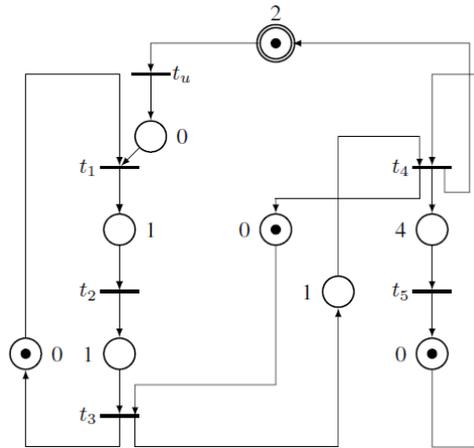


FIG. 3.13: Modèle GET de système manufacturier avec la place de control

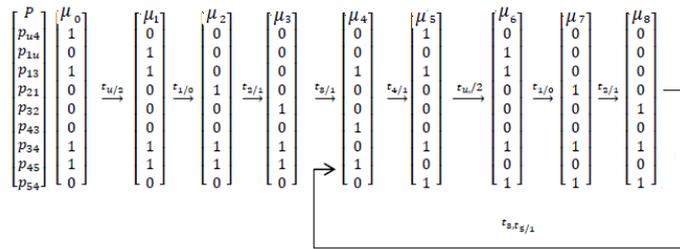


FIG. 3.14: Graphe de marquage de Modèle GET de la Fig.3.13

$s$ . Les fonctions  $x_j(t)$  et  $x_i(t)$  définissent les compteurs correspondants.

Chaque chemin  $\rho_s$  est soumis à une contrainte de marquage, qui exige que le nombre de jetons dans le chemin  $\rho_s$  ne dépasse pas  $b_s$  avec  $s$  allant de 1 à  $S$ . Nous représentons ces spécifications de marquage imposées sur le graphe  $G$  par cette inéquation linéaire :

$$x_j(t) \leq (b_s - M_{\rho_{s_0}}) \cdot x_i(t), \quad (3.26)$$

**Théoreme 10.** *Nous considérons un graphe d'événements temporisé  $\mathcal{G}$ . Son évolution est représentée par l'équation (1.3) et soumise à plusieurs contraintes sous la forme de (3.26).*

*Le contrôleur donné par :*

$$u(t) = \bigoplus_{s \geq 1} u_s(t),$$

avec

$$u_s(t) = (b_s - M_{\rho_{s_0}} - (A^{\tau_{\alpha_s}} \cdot \mathbf{B})_j) \cdot A^{\tau_{\alpha_s} + 1ir} \cdot x(t - 1),$$

assure la satisfaction des spécifications de marquage (3.26) si la condition suivante est

bien vérifiée :

$$(A^{\tau_{\alpha_s}} \cdot B)_j \leq (b_s - M_{\rho_{s0}}) \cdot (A^k \cdot B)_i, \quad k = 0, \dots, \phi_s - 1 \quad (3.27)$$

avec  $\phi_s = \tau_{\alpha_s} + 1$ ,

**Démonstaration.** D'après le Théorème 8, le contrôleur 3.23 garantit le respect de la contrainte de marquage (3.21). Donc pour  $S$  contraintes de marquage, le contrôleur  $u_s(t)$  garantira la  $S^{\text{ème}}$  contrainte si la condition (3.22) est satisfaite.

Le feedback qui garantit les  $S$  contraintes (3.26) est alors donné par :

$$u(t) = \oplus_{s=1}^S u_s(t) \text{ pour } s=1 \text{ à } S.$$

Ce feedback correspond au contrôleur minimum calculé pour les  $S$  contraintes. □

### 3.3.2. Commande multivariables

Nous considérons dans cette section, un graphe d'événements temporisé avec plusieurs transitions de commande. Le comportement de ce graphe est donné par l'équation d'état (1.4), avec un vecteur  $u(t) \in \bar{\mathbb{R}}_{min}^m$  et  $B \in \bar{\mathbb{R}}_{min}^{N \times m}$ , pour  $m \geq 1$ . Nous supposons que ce graphe est soumis à des contraintes de marquage imposées sur des places ou sur certains chemins. Ces contraintes peuvent être représentées par des inéquations linéaires dans l'algèbre Min-Plus. Nous cherchons le vecteur de commande  $u(t)$  qui garantit ces contraintes. Nous considérons en premier lieu un GET soumis à une seule contrainte, puis en second lieu un graphe d'événements temporisé soumis à plusieurs contraintes de marquage.

#### 3.3.2.1. Une seule contrainte

On propose dans cette partie une démarche de synthèse de lois de commande pour satisfaire une seule contrainte de marquage imposée à un GET. Nous supposons que cette contrainte est imposée sur une place ou sur un chemin. Nous associons à chaque transition source un compteur noté  $u_l(t)$  pour  $l = 1$  à  $m$ . La contrainte de marquage imposée à ce graphe est donnée par :

$$M_\rho(t) \leq b \quad (3.28)$$

tel que,  $\rho$  représente le chemin sous contrainte.

Nous notons par  $t_j$  et  $t_i$  respectivement la transition d'entrée et la transition de sortie du chemin  $\rho$ .

Comme dans le cas de la commande avec une seule transition de commande, nous supposons qu'il existe au moins un chemin qui relie la transition de commande  $u_l$  et la transition  $t_j$  représentant la transition d'entrée du chemin sous contrainte. (Hypothèse 3). La relation entre ces deux transitions peut s'exprimer par cette inéquation :

$$x_j(t) \leq \bigoplus_{l=1}^m (A^{\tau_l} \cdot B)_{jl} \cdot u_l(t - \tau_l)$$

Dans ce cas, les deux inéquations (3.12) et (3.13) sont remplacées par :

$$\bigoplus_{l=1}^m (A^{\tau_\alpha} \cdot B)_{jl} \cdot u_l(t - \tau_\alpha) \leq (b - M_{\rho 0}) \left[ \bigoplus_{k=0}^{\phi-1} \bigoplus_{l=1}^m (A^k \cdot B)_{il} \cdot u_l(t - k) \right] \quad (3.29)$$

et

$$\bigoplus_{r=1}^N \left[ \bigoplus_{l=1}^m u_l(t - \tau_\alpha) \leq [(b - M_{\rho 0} - (A^{\tau_\alpha} \cdot B)_{jl}) \cdot A_{ir}^\phi] \cdot x_r(t - \phi) \right] \quad (3.30)$$

Nous cherchons à définir la relation entre les deux variables  $\phi$  et  $\tau_\alpha$  qui qui permet de satisfaire les deux inéquations précédentes pour que la contrainte (3.28) soit satisfaite.

Comme dans le cas de la commande des GET avec une seule transition de commande, nous supposons qu'il existe au moins un chemin  $\alpha$  avec une temporisation  $\tau_\alpha$  qui relie une transition de commande  $u_c$  à la transition  $t_j$ , ce qui implique que  $(A^{\tau_\alpha} \cdot B)_{jc} \neq \varepsilon$ .

Pour  $\phi = \tau_\alpha + 1$  le contrôleur  $u_c(t)$  donnée par l'inéquation 3.30 peut s'écrire comme suite :

$$u_c(t) \leq [(b - M_{\rho 0} - (A^{\tau_\alpha} \cdot B)_{jc}) \cdot A_{ir}^\phi] \cdot x_r(t - 1) \quad (3.31)$$

L'inéquation 3.29 est satisfaite si les deux conditions suivantes sont vérifiées :

$$(A^{\tau_\alpha} \cdot B)_{jc} \leq (b - M_{\rho 0}) (A^k \cdot B)_{il} \quad (3.32)$$

$$u(t - \tau_\alpha) \leq u_c(t - k)$$

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

**Théoreme 11.** Soit  $G$  un GET avec  $m$  transitions d'entrées et soumis à  $s$  contraintes de marquage sous la forme (3.28). Le contrôleur :

$$u_c(t) = \begin{cases} (b - M_{\rho_0} - (A^{\tau_\alpha} \cdot B)_{jc}) \cdot A_{ir}^{\phi_c} \cdot \mathbf{x}(t-1) & \text{si } l = c, \\ \varepsilon, & \text{sinon.} \end{cases} \quad (3.34)$$

garantit la validation des  $s$  contraintes (3.28) si la condition suivante :

$$(b - M_\rho) \cdot (A^k \cdot B)_{il} \geq (A^{\tau_\alpha} \cdot B)_{jc} \quad (3.35)$$

est vérifiée.

Avec  $\phi = \tau_\alpha + 1$ ,  $l = 1$  à  $m$ ,  $k = 0$  à  $\phi - 1$ ,

**Démonstaration.** Pour  $\phi = \tau_\alpha + 1$ , grâce à l'inéquation (3.30) la formule de contrôleur qui garantit la satisfaction de la contrainte (3.28) est exprimé par l'inéquation (3.31). Comme notre objectif est de trouver un contrôleur qui garantit la satisfaction de la contrainte (3.28) , il suffit alors de trouver une seule composante de vecteur  $u(t)$  qui assure l'accomplissement de cette objective. Nous avons par la suite fixé toutes les autres composantes à  $\varepsilon$ .

Finalement, le contrôleur (3.34) garantit la satisfaction de la contrainte de marquage (3.28), si la condition (3.35) est bien vérifiée.  $\square$

#### 3.3.2.2.Plusieurs contraintes

Soit  $G$  un graphe d'événements temporisé avec  $m$  transitions d'entrée. Ce graphe est soumis à  $S$  contraintes de marquage. Ces contraintes sont imposées sur certains places ou chemins notés  $\rho_s$  pour  $s = 1$  à  $S$ . Nous notons par  $t_j$  et  $t_i$  respectivement la transition d'entrée et la transition de sortie de chemin  $\rho_s$ . Nous notons par  $b_s$  le nombre de jetons maximum qui ne doit pas être dépassé dans le chemin  $\rho_s$ . Ces contraintes sont exprimées par les inéquations suivantes :

$$M_{\rho_s}(t) \leq b_s \quad (3.36)$$

Nous supposons que pour chaque contrainte  $s$ , il existe au moins un chemin noté,  $\alpha_s$  reliant la transition de commande  $tu_{c_s}$  à la transition d'entrée de chemin sous contrainte  $\rho_s$ ,

où  $c_s$  désigne l'indice associé à la transition de commande qui agit sur la contrainte  $s$ . On note par  $\tau_s$  la temporisation de ce chemin. Cette condition peut se traduire comme suit :

$$x_s(t) \leq (A^{\tau_s} \cdot B)_{sc_s} \cdot u_{c_s}(t - \tau_s). \quad (3.37)$$

L'équation explicite de la transition  $t_i$  est donnée par :

$$x_i(t) = \bigoplus_{r=1}^N (A^{\phi_s})_{ir} \cdot x_r(t - \phi_s) \oplus \bigoplus_{k=0}^{\phi_s-1} ((A^k \cdot B)_i \cdot u(t-k)), \quad (3.38)$$

avec  $\phi_s \geq 1$ .

Nous utilisons les résultats pour le cas de la commande d'une seule contrainte, obtenus précédemment, et nous traitons chacune des contraintes séparément. Pour  $s=1$  à  $S$ , la contrainte  $s$  est équivalente aux deux inégalités (3.29), (3.30) sont données par les deux inéquations suivantes :

$$(A^{\tau_{\alpha_s}} \cdot B)_{jc_s} \cdot u_{c_s}(t - \tau_{\alpha_s}) \leq (b - M_{\rho_0}) \left[ \bigoplus_{k=0}^{\phi_s-1} \bigoplus_{l=1}^m (A^k \cdot B)_{il} \cdot u_l(t - k) \right] \quad (3.39)$$

et

$$u_{c_s}(t - \tau_{\alpha_s}) \leq [(b - M_{\rho_{s0}} - (A^{\tau_{\alpha_s}} \cdot B)_{jc_s}) \cdot A_{ir}^{\phi_s}] \cdot x_r(t - \phi_s) \quad (3.40)$$

avec  $\phi_s = \tau_{\alpha_s} + 1$  pour  $s= 1$  à  $S$ .

Si pour tout  $s$ , les conditions :

$$(b_s - M_{\rho_s}) \cdot (A^k \cdot B)_{il} \geq (A^{\tau_{\alpha_s}} \cdot B)_{jc_s} \quad (3.41)$$

et

$$u_l(t) \geq u_{c_s}(t) \quad (3.42)$$

sont vérifiées, le contrôleur est alors donné par l'équation suivante :

$$u_{c_s}(t) \leq [(b - M_{\rho_{s0}} - (A^{\tau_{\alpha_s}} \cdot B)_{jc_s}) \cdot A_{ir}^{\phi_s}] \cdot x_r(t - 1)$$

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

**Théoreme 12.** *Le contrôleur donnée par :*

$$u_l(t) = \bigoplus_{s=1}^S \mathbf{u}_{c_s}(t) \quad (3.43)$$

avec  $u_{c_s}(t) = (b - M_{\rho_{s0}}) - (A^{\tau_{\alpha_s}} \cdot B)_{j_{c_s}} \cdot A_{i_r}^{\phi_s} \cdot x_r(t - 1)$

garantit la satisfaction de  $s$  contrainte imposées à un GET  $G$  si la condition 3.35 est vérifiée pour les  $s$  contrainte.

Tel que  $u_c(t)$  est le contrôleur 3.34 calculé dans le Théorème 10.

**Démonstaration.** Comme nous l'avons démontré dans le Théorème 11, le contrôleur 3.34 garantit le respect de la contrainte de marquage (3.28). Donc pour  $S$  contraintes de marquage le contrôleur  $u_{c_s}(t)$  garantit la satisfaction de la Sème contrainte si la condition (3.35) est satisfaite.

Le feedback qui garantit les  $S$  contraintes (3.36) est alors donné par :

$$u_l(t) = \bigoplus_{s=1}^S u_{c_s}(t) \text{ pour } s=1 \text{ à } S.$$

Ce feedback correspond au contrôleur minimum calculé pour les  $S$  contraintes.

□

**Exemple 10.** *Nous considérons dans cet exemple un système d'assemblage avec deux lignes de production, modélisé par le graphe d'événements temporisé schématisé dans la Fig.3.15. Ce graphe contient deux transitions de commande ( $m = 2$ ) notées respectivement  $t_{u1}$  et  $t_{u2}$ . Ce système comporte deux stocks limités modélisés respectivement par les deux places  $p_{32}$  et  $p_{76}$ . De ce fait, deux contraintes de marquage imposées sur ces deux places. Ces deux contraintes sont définit par ces deux inéquations :*

$$M_{32} \leq 2$$

$$M_{76} \leq 3$$

La Fig.3.16 représente le garphe de marquage de GET de la Fig.3.15, nous pouvons remarqué qu'à partir de l'état  $\mu_7$  le nombre de jetons dans les deux place  $p_{32}$ ,  $p_{76}$  dépasse le nombre imposé par le cahier de charge ( $_{32} \leq 2$ ,  $M_{76} \leq 3$ ).

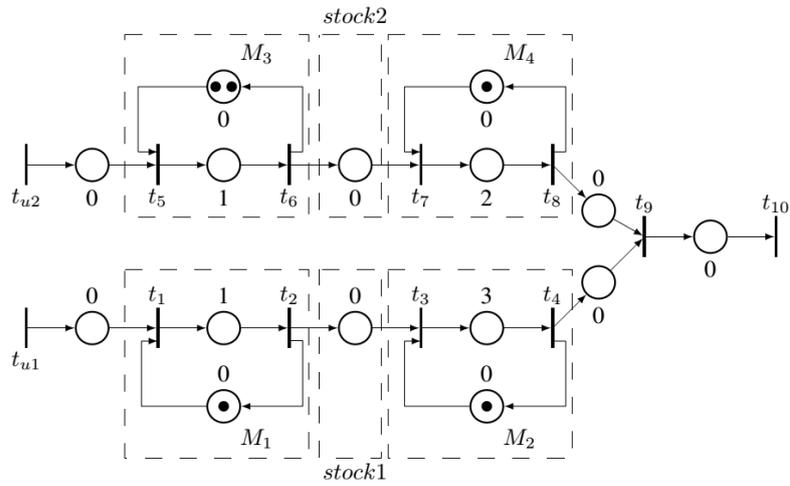


FIG. 3.15: Modèle GET de système d'assemblage

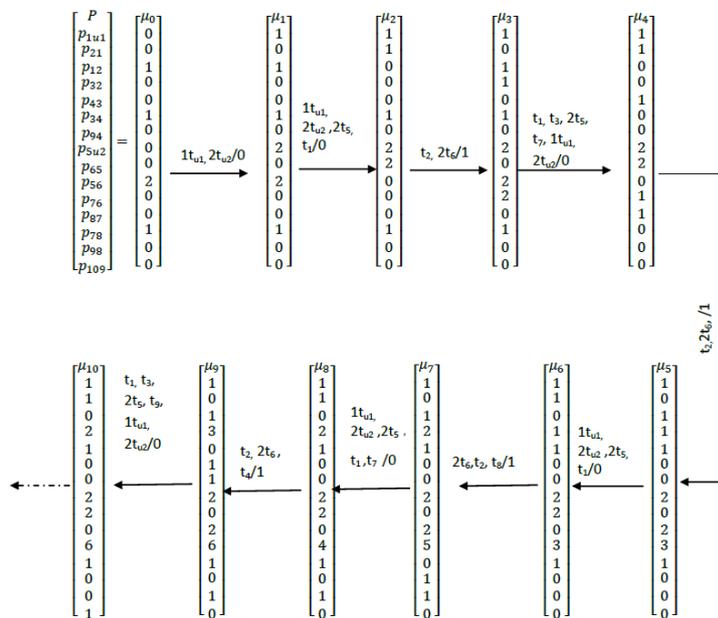


FIG. 3.16: Graphe de marquage de Modèle GET de la Fig.3.15

### 3.3 Commande des graphes d'événements temporisés sous contraintes de marquage

Nous souhaitons calculer le vecteur de contrôle  $u(t)$ , afin de limiter le nombre de jetons dans le stock 1 et le stock 2 à  $(b_1, b_2)$ .

Avant d'appliquer le Théorème 12, nous décrivons d'abord la dynamique de modèle de système d'assemblage par une équation d'état. Pour obtenir cette équation, nous avons décomposé la place  $(p_{34})$  en trois places temporisées à 1 unité de temps et deux transitions sont ajoutées  $(t_{11}, t_{12})$ . Nous avons ainsi décomposé la place  $(p_{87})$  en deux places temporisées à 1 unité de temps, et une transition est ajoutée  $(t_{13})$ . L'équation d'état du système est donnée par :

$$x(t) = \begin{bmatrix} 1 & \varepsilon \\ e & \varepsilon \\ e & \varepsilon & 1 & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & e \\ \varepsilon & e \\ \varepsilon & e & e \\ \varepsilon & e & e \\ \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix} \oplus \begin{bmatrix} e & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \end{bmatrix} \cdot u(t).$$

Nous appliquons les résultats du Théorème 12 pour déterminer les lois de commande qui assurent le respect des deux contraintes de capacité imposées sur ce système d'assemblage. Nous avons  $M_{320} = 0$ , on remarque bien qu'il existe un chemin qui relie la transition de contrôle  $t_{u1}$  à la transition  $t_2$ , on note ce chemin  $\alpha_1$ . La temporisation de ce chemin est  $\tau_{\alpha_1} = 1$ . La condition du Théorème 12 donnée par :

$$2.(A^k \cdot B)_{3l} \geq (A \cdot B)_{21}$$

est bien vérifiée pour  $k = 0$   $\phi - 1 = 1$  et  $l = 1, 2$ .

D'après le Théorème 12, puisque la condition est bien vérifiée alors il existe un contrôleur

qui garantit la satisfaction de la contrainte  $M_{32} \leq 2$ , les lois de commande qui représente le premier élément du vecteur de contrôle est donnée par :

$$\begin{aligned}
 u_1(t) &= 2 - (A^1 \cdot B)_{21} \cdot \bigoplus_{r=1}^{13} A_{3r}^2 \\
 u_1(t) &= 2(1 \cdot x_1(t-1) \oplus 1 \cdot x_{11}(t-1)) \\
 &= 3 \cdot x_1(t-1) \oplus 3 \cdot x_{11}(t-1) \\
 &= 3 \cdot x_1(t-1) \oplus 3 \cdot x_3(t-2) \\
 &= 3 \cdot x_3(t-2)
 \end{aligned}$$

Nous passons maintenant au calcul de la deuxième composante du vecteur du contrôle  $u(t)$ , les lois de commande qui garantissent la contrainte  $M_{76} \leq 3$ , est la deuxième contrainte imposée sur le système d'assemblage.

Nous avons  $M_{760} = 0$ , on remarque bien qu'il existe un chemin qui relie la transition de contrôle  $t_{\alpha 2}$  à la transition  $t_2$ , le délai du chemin qui relie la transition de contrôle  $t_{\alpha 2}$  à la transition  $t_j = t_6$  est on a  $\tau_{\alpha 2} = 1$ . En remplaçant  $\tau_{\alpha 2}$  dans le Théorème, la condition de ce Théorème est donnée par :

$$3 \cdot (A^k \cdot B)_{7l} \geq (A \cdot B)_{62}$$

on remarque bien que la condition est bien satisfaite pour  $k = 0$   $\phi - 1 = 1$  et pour  $l = 1, 2$ . On prend toujours le Théorème 12 en considération, l'équation de la loi de commande qui représente le deuxième élément du vecteur de contrôle qui est donnée par :

$$\begin{aligned}
 u_2(t) &= 3 - (A^1 \cdot B)_{62} \cdot \bigoplus_{r=1}^{13} A_{7r}^2 \\
 u_2(t) &= 3(2 \cdot x_5(t-1) \oplus e \cdot x_7(t-1)) \\
 &= 5 \cdot x_5(t-1) \oplus 3 \cdot x_7(t-1)
 \end{aligned}$$

On prend en considération les chemins de GET de la Fig.3.15, on a :

$$x_5(t-1) \geq x_7(t-1),$$

alors

$$5 \cdot x_5(t-1) \geq 3 \cdot x_7(t-1)$$

l'équation  $u_2(t)$  peut se réduire à cette équation :

$$u_2(t) = 3 \cdot x_7(t-1)$$

Finalement, le contrôleur qui garantit les deux contraintes imposées sur le système d'assemblage est donné par le vecteur suivant :

$$\mathbf{u}(t) = \begin{bmatrix} 3 \cdot x_3(t-2) \\ 3 \cdot x_7(t-1) \end{bmatrix}$$

Ces lois de commande sont représentées par des places marquées et temporisés connectées au modèle du système d'assemblage (places représentées par des doubles cercles dans la Fig.3.17, et grâce à la Fig.3.18 qui représente l'évolution de système après l'intégration des lois de commande calculées, nous pouvons conclure que les contraintes sont bien respectées.

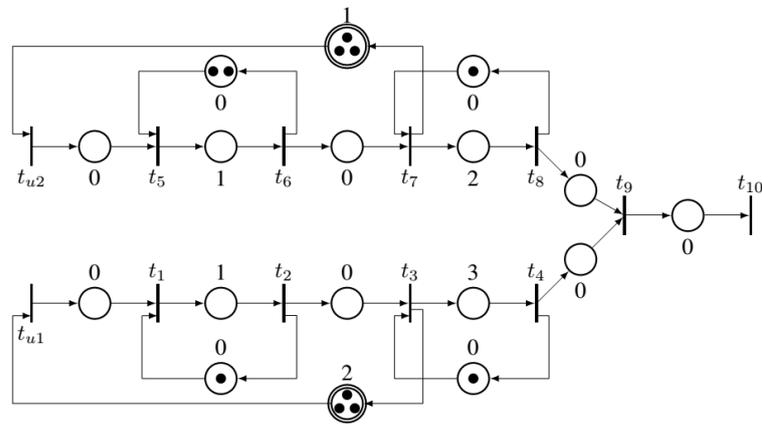


FIG. 3.17: Modèle GET de système d'assemblage avec les places de contrôle

### 3.4 Conclusion

Dans cette partie, nous avons proposé une approche de commande des systèmes à événements discrets, modélisés par des graphes d'événements temporisés sous contraintes de

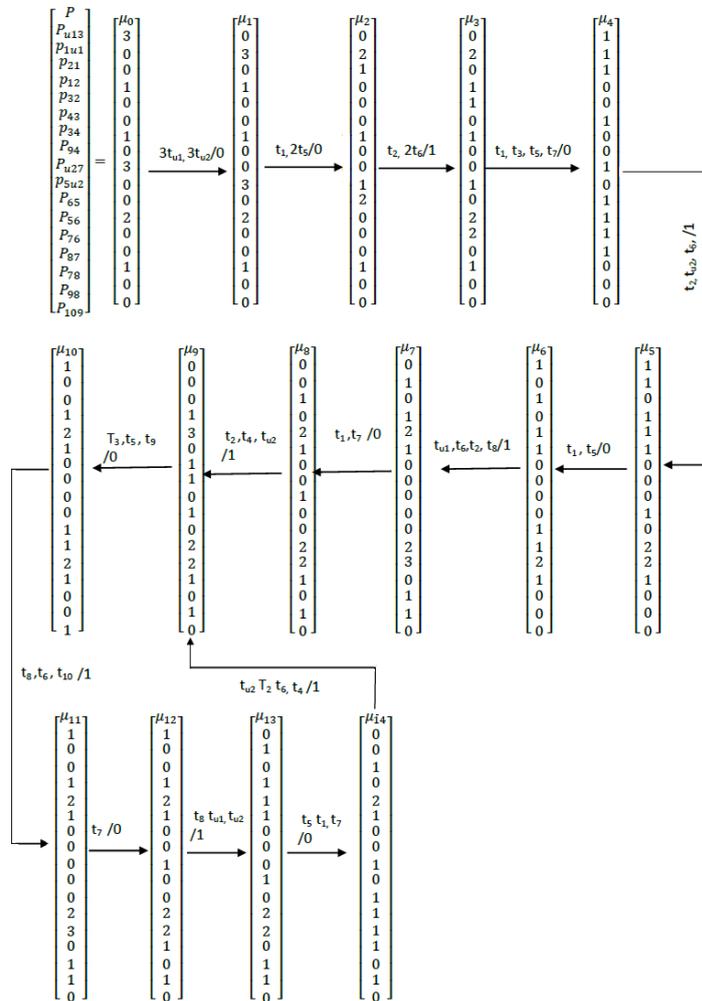


FIG. 3.18: Graphe de marquage de Modèle GET de la Fig.3.17

marquage. Notre approche est basée sur l'utilisation de l'algèbre Min-Plus. Nous avons utilisé l'algèbre Min-Plus pour représenter le comportement de GET par une équation linéaire et les contraintes de marquage par des inéquations linéaires dans l'algèbre Min-Plus. Les lois de commande calculées sont des feedbacks qui sont sous la forme de  $u(t) = F \cdot x(t-1)$ . Dans la première partie, nous avons considéré le problème de commande des GETs avec une seule transition de commande soumis à une seule ou plusieurs contraintes de marquage. Dans la deuxième partie, nous avons généralisé notre étude pour le cas des GETs avec plusieurs transitions de commande. Dans les deux cas, nous avons considéré le problème de commande sous des contraintes de marquage imposée sur des places indépen-

---

dantes d'un GET et des contraintes imposées sur des places dépendantes (chemins) d'un GET.



# Conclusion générale

Nous avons présenté dans ce manuscrit deux méthodologies de synthèse de lois de commande pour les systèmes à événements discrets modélisés par des graphes d'événements temporisés. Nous avons développé une approche de commande des graphes d'événements temporisés soumis à des spécifications (contraintes) de temps ou de marquage imposées aux places ou bien aux chemins de ces graphes.

Nous avons présenté dans la première partie de cette thèse une méthode formelle pour la synthèse des lois de commande en boucle fermée afin de satisfaire un ensemble de contraintes temporelles imposées aux systèmes à événements discrets modélisés par des graphes d'événements temporisés. L'approche développée est basée sur l'utilisation de l'algèbre Max-Plus. Le comportement dynamique du graphe d'événements temporisé est exprimé par une équation linéaire Max-Plus. Grâce aux dateurs associés à chaque transition du GET, les contraintes temporelles sont représentées par des inéquations linéaires dans l'algèbre Max-Plus. Des conditions suffisantes pour l'existence de lois de commande qui satisfont ces contraintes temporelles sont définies. Les lois de commande calculées peuvent être représentées par des places de contrôle connectées au graphe initial afin de garantir le respect des contraintes. Par la suite, nous avons appliqué les résultats de notre approche pour la commande d'une architecture d'automatisation en réseau soumise à une contrainte temporelle imposée sur leur temps de réponse. Les lois de commandes calculées sont des feedbacks causaux qui sont représentés par des places de contrôle connecté au modèle GET du l'architecture d'automatisation en réseau.

Dans la deuxième partie de ce travail, nous avons abordé le problème de commande de systèmes à événements discrets soumis à des contraintes de capacités de ressources. Nous

avons présenté une nouvelle méthode pour la synthèse de contrôleurs pour des graphes d'événements temporisés soumis à des spécifications sur les marquages. L'approche que nous avons proposée est basée sur l'algèbre Min-Plus, où le comportement dynamique du GET est exprimé par des équations linéaires et les contraintes de marquage par des inéquations linéaires dans l'algèbre Min-Plus. Nous avons défini une condition suffisante pour l'existence de lois de commande qui satisfont ces contraintes. Dans un premier temps, nous avons considéré le cas où le problème posé est de limiter le nombre de jetons dans une seule place. Ensuite, nous avons étendu l'approche pour le cas où les spécifications imposées sur le système exigent de borner le nombre de jetons dans une séquence de places (chemin). Pour les deux types de contraintes, nous avons tout d'abord considéré le cas de GETs avec une seule transition de commande, par la suite nous avons généralisé notre contribution pour le cas de GETs avec plusieurs transitions d'entrée.

Plusieurs perspectives peuvent être envisagées pour ce travail : sur l'aspect théorique, il serait intéressant de considérer des systèmes où les transitions du GET ne sont pas toutes observables. Dans nos travaux nous avons considéré le cas où le chemin  $\alpha$  qui relie la transition de commande et la transition d'entrée de la place ou de chemin sous contrainte est vide, il serait intéressant d'envisager une amélioration de la technique proposée ou bien une nouvelle méthode pour le cas où le chemin  $\alpha$  est non vide.

Pour le cas de commande des GETs sous contraintes de marquage imposées sur plusieurs places, nous avons limité notre étude au cas où les places sont situées sur le même chemin. Il serait intéressant d'envisager une commande basée sur l'algèbre Min-Plus pour le cas de problème d'exclusion mutuelle (places ne sont pas sur le même chemin).

Sur le plan pratique, pour la commande des architectures d'automatisation en réseaux, il reste à étendre la validité des résultats de cette étude et leur généralisation en mettant l'accent sur des AARs plus complexes (plusieurs contrôleurs et plusieurs modules d'entrées/sorties) et d'étudier d'autres AAR qui fonctionnent avec d'autres protocoles de communication (par exemple, client/serveur). En outre, il est important d'étendre la validité de notre contribution dans différents contextes, notamment pour la vérification et la validation des systèmes de production automatisés et les réseaux de transport.

# Bibliographie

- [1] Amari S., I. Demongodin and J-J. Loiseau,. Sizing, cycle time and plant control using dioid algebra. Chapter 6 in Supply Chain Optimization, Series Applied Optimization, A. DolguiJ.Soldek and O. Zaikin (Eds), Springer, pp. 71-85,(2004).
- [2] Amari S., Demongodin I., Loiseau J.-J. and Martinez C. : Max-Plus Control Design for Temporal Constraints Meeting in Timed Event Graphs, IEEE Transactions. on Automatic Control, 57 (2), pp.462-467, (2012).
- [3] Amari. S., Feedback control for a class of discrete event systems with critical time, International Journal of Control, vol.219(66), pp.957-984, (2015).
- [4] Amari Said "Commande des graphes d'événements temporises sous contraintes temporelles" Thèse de doctorat, Université de Nantes, (2005).
- [5] Abderrahim Benfekir "Contribution à lanalyse de performances des Systèmes à Evénements Discrets non linéaires dans lalgèbre (min, +)" Thèse de doctorat, Université de Cergy, (2013).
- [6] Achour, Z., Rezg, N., Time Floating General Mutual Exclusion Constraints (TFG-MEC), Comutational Engineering in Systems Applications, IMACS Multiconference on. vol.1, pp.893-897, 2006.
- [7] Addad. B., Amari,S., and J-J. Lesage, Analytic calculus of response time in networked automation, IEEE Transactions on Automation Science and Engineering, vol.7(4), pp. 858 869, 2010.
- [8] Atto. A. M., Martinez C. and Amari, S., Control of Discrete Event Systems with Respect to Strict Duration : Supervision of an Industrial Manufacturing Plant, Computers and Industrial Engineering, Elsevier, 61(4), pp.1149-1159,(2011).

- 
- [9] Atli, M., Adjallah, K., Achour, Z., Sava, A., Rezg, N., Supervisory control of timed-place marked graph based on Marking Exclusion Constraint, International Conference on Communications, Computing and Control Applications (CCCA), pp.1-8, 2011.
- [10] Badouel. E., Darondeau. P. Bernardinello. L , "Polynomial algorithms for the synthesis of bounded nets", in Proceedings Caap 95, Lecture Notes in Computer Science 915, Springer, pp. 364378, (1994).
- [11] Baccelli. B., Cohen. G., & Gaujal. B, "Recursive equations and basic properties of timed Petri nets", Discrete Event Dynamic Systems : Theory and Applications, vol.1(4), pp.145-439, (1992).
- [12] Baccelli. F., Cohen. G., Olsder. G., & Quadrat. J, "Synchronization and linearity : an algebra for discrete event systems", Wiley,(199).
- [13] Bonhomme, P, "Constraints Graph Based Approach for the Analysis and the Control of Time Critical Systems", International Journal of Advanced Manufacturing Technology 57,(1), pp. 353-365, (2011).
- [14] Brandin. B., & Wonham. W, "Supervisory control of timed discrete event systems", IEEE Trans. Autom. Control. vol.2(39), pp.329-342, (1994).
- [15] Cottenceau, B.L., Hardouin, Boimond J.L. and Ferrier J.L, "Synthesis of Greatest linear feedback for timed event graphs in dioid", IEEE Transactions on Automatic Control, 44(6), pp.1258-1262, (1999).
- [16] Cena G., Cibrario Bertolotti I., Valenzano A., Zunino C. : Evaluation of Response Times in Industrial WLANs, IEEE Transactions on Industrial Informatics, 3(3), pp. 191-201,(2007).
- [17] Cohen, G., Dubois, D., Quadrat, J.-P., and Viot, M. "Analyse du comportement périodique des systèmes de production par la théorie des dioïdes".Rapport de recherche 191, INRIA, Le Chesnay, France, (1983).

- [18] Cohen. G., Dubois. D., Quadrat. J.-P., and Viot. M, " A linear system theoretic view of discrete event processes and its use for performance evaluation in manufacturing". IEEE Trans. on Automatic Control, 30(3) :210-220,(1985).
- [19] Cohen. G., Gaubert. S and Quadrat., J. P., " Max plus algebra and systems theory : wherewe are and where to go now", Annual Reviews in Control, vol. 23, pp.207-219,(1999).
- [20] David.R et . Alla.H, "Du grafcet au réseaux de petri". Editions Hermès, Paris, (1992).
- [21] Diouri. I, Georges. J-P, Rondeau. E, "Accommodation of delays for networked control systems using classification of service". IEEE Int Conf Netw Sensing and Control London :410 415, (2007).
- [22] Dideban, A. A., Alla, H, "Controller synthesis with highly simplified linear constraints, Asian Journal of Control", vol.15(1), pp.80-94, 2013.
- [23] Georges. J-P., Thierry Divoux, and Eric Rondeau, Validation of the network calculus approach for the performance evaluation of switched ethernet based industrial communications, In Proceedings of the 16th IFAC World Congress, 2005, Prague, Czech Republic.
- [24] Giua. A., DiCesare. F.M., & Silva. M, "Generalized Mutual Exclusion Constraints on Petri Nets with Uncontrollable Transitions", In Proc. IEEE int. conf. SMC, vol.2, pp.974-979, 1992.
- [25] Gianluca.C., Cibrario Bertolotti. I, and A.Valenzano, Experimental analysis of latencies in Etherne, In Proc. of WFCS06.
- [26] Ghaffari. A., Rezg. N., Xie. X, "Design of a Live and Maximally Permissive Petri Net Controller Using the Theory of Regions", IEEE Transactions on Robotics and Automation. vol.(19)1, pp.137-142, 2003.
- [27] Greifeneder J., Liu L., Frey G. : Comparing Simulative and Formal Methods for the Analysis of Response Times in Networked Automation Systems, IFAC World Congress, Korea South, 17 (1), pp.5113-5118, (2008).

- [28] Houssin. L., Lahaye. S., & Boimond. J.L, "Control of (max,+)-linear systems minimizing delays", Discrete Event Dynamic Systems, Springer Verlag, 23(3), pp.261-276, (2013).
- [29] Houssin. L, Lahaye. S., Boimond. J.L., "Just in time control of constrained (max,+)-linear systems", Discrete Event Dynamic Systems, vol 17, (2007).
- [30] Houssin. L., "Contribution à l'étude des systèmes à événements discrets dans l'algèbre des diodes, Applications aux systèmes de transport". Rapport de stage de DEA, DEA Automatique et Informatique Appliquée, Nantes-Angers, France,(2003).
- [31] Jacob. R., Amari. S., "Output feedback control of discrete processes under time constraint : application to cluster tools". International Journal of Computer Integrated Manufacturing, p.1-15, (2016).
- [32] Kumar.R and Garg. V.K," Modeling and control of logical discrete event systems". Kluwer Academic Publishers, USA, (1995).
- [33] Kim. J-H., Tae-Eog Lee, Hwan-Yong. Lee and Doo- Byeong Park, Scheduling Analysis of Time- Constrained Dual-Armed Cluster Tools, IEEE Transactions on Semiconductor Manufacturing.Vol. 16, No. 3, pp. 521-534, (2003).
- [34] Lahaye. S., Cottenceau. B., Correia. A., "Commande de graphes d'événements temporisés avec contraintes de temps critique", Conférence internationale d'Automatique francophone CIFA, Tunisie (2004).
- [35] Lee. K. C., and Lee. S, Performance evaluation of switched Ethernet for real-time industrial communications Computer standards interfaces, (24) :411423, 2002.
- [36] Le Boudec, J.-Y. and Thiran, P. "Nework Calculus". Springer- Verlag. *http* : *//ica1www.epfl.ch/PSfiles/NetCal.htm*.(2001).
- [37] Marsal. G., Bruno Denis, Jean-Marc Faure, and Georg Frey, Evaluation of Response Time in Ethernet-based Automation Systems. In 11th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2006, Prague, Czech Republic.

- [38] Maia. C.A., Andrade. C.R., Hardouin. L, " On the control of max- plus linear system subject to state restriction *Automatica*, 47(5), pp.988-992, (2011).
- [39] Mehdi Lhommeau, " Étude de systèmes à événements discrets dans l'algèbre  $(\max; +)$ ", Thèse de doctorat, l'ISTIA - Université d'Angers, 2003.
- [40] Menguy. E., " Contribution à la commande des systèmes linéaires dans les dioïdes". Thèse, LISA - Université d'Angers,(1997).
- [41] Moody J. O., Antsaklis P., Petri net supervisors for DES with uncontrollable and unobservable transition, *IEEE Trans. Automatic Control*, 45(3) :462-476, Mars (2000).
- [42] Murata.T, "Petri Nets, "Properties, Analysis and Applications". In *IEEE Proceedings : Special issue on Discrete Event Systems*, volume 77, pages 541-581, (1989).
- [43] Proth. J.M., et X. Xie, "Les réseaux de petri pour la conception et la gestion des systèmes de production". Edition MASSON, (1995).
- [44] Petri.C.-A, *Kommunikation mit Automaten*. Thèse, Institut für Instrumentelle Mathematik, Bonn, Germany, (1962).
- [45] Ramadge. P. J., & Wonham. W. M, "Modular feedback logic for discrete event systems, *SIAM Journal of Control and Optimization*", vol.25(5), pp.1202-1218, (1987).
- [46] Ramadge. P. J. Wonham, W.M, "The control of discrete event systems. Dynamics of discrete event systems". Special issue. *Proceedings of the IEEE*, vol.77(1), pp.81-98, (1989).
- [47] Ran. N., Wang. S., Su. H., & Wang, C, "Supervisor synthesis for enforcing linear constraints on a class of Petri nets with unobservable transitions", *IMA Journal of Mathematical Control and Information*, doi : 10.1093/imamci/dnv059,(2015).
- [48] Tebani Karima, " Commande des systèmes à événements discrets à temps critiques, application aux systèmes de commande en réseau", Thèse de Magister, Université Mouloud Mammeri de Tizi-Ouzou, 2012.

- 
- [49] Tebani. K., Amari. S., Kara. R, "Closed-loop control of constrained discrete event systems : application to a networked automation system". *The International Journal of Advanced Manufacturing Technology*,90(5), 1295-1306,(2016).
- [50] Uzam. M., An Optimal Deadlock Prevention Policy for Flexible Manufacturing Systems Using Petri Net Models with resources and the Theory of Regions *Int J Adv Manuf Tech*, 19 : 192-208, (2002).
- [51] Vasiliu, A. and Alla, H., "Structural optimal control for safe petri nets". *International Journal of Control*, 83(9), 1810-1822, (2010).
- [52] Wang. Y, De Schutter. B, van den Boom. TJJ, Ning. B., "Optimal trajectory planning for trainsa pseudospectral method and a mixed integer linear programming approach". *Transp Res C* 29 :97 114, (2013).
- [53] Yamalidou. K., Moody. J., Lemmon. M., & Antsaklis. P," Feedback control Petri net based on place invariant". *Automatica*, vol. 32(1), pp.15-28, (1996).
- [54] Yamalidou. E.C., Kator. J.C, "Modeling and optimal control of discrete-event chemical processes using petri nets", *Computers & Chemical Engineering*, vol.15(7), pp.503–519, (1991).
- [55] Zareiee. M., Dideban. A., Orouji. A. A., Soltanizadeh, H. "Solving the problem of forbidden states in discrete event systems : a novel systematic method for reducing the number of control places". *Asian Journal of Control*, 17(3), 1006-1015.(2015)