

# Commande de Graphes d'Événements Temporisés partiellement commandables avec contraintes temporelles

Aurélien CORRÉIA<sup>1</sup>, Sébastien LAHAYE<sup>2</sup> et Bertrand COTTENCEAU<sup>2</sup>

<sup>1</sup>SeT - EA3317

Université de Technologie de Belfort-Montbéliard  
90010 Belfort Cedex, France

<sup>2</sup>LISA - EA4014

Université d'Angers  
62, avenue Notre Dame du Lac  
49000 Angers, France

aurelien.correia@utbm.fr, lahaye@istia.univ-angers.fr, cottence@istia.univ-angers.fr

*Résumé*— Il existe une classe de Systèmes (dynamiques) à Événements Discrets représentables par des Graphes d'Événements Temporisés. Il a été montré que ces systèmes peuvent être modélisés par des équations linéaires dans les dioïdes. On présente une démarche de synthèse d'un correcteur qui, placé en amont des entrées du système, permet de poursuivre au mieux un modèle de référence en tenant compte de contraintes temporelles dans certains chemins d'un Graphes d'Événements Temporisés. La méthode généralise celle proposée dans [1] en intégrant les cas où plusieurs contraintes temporelles interviennent au sein du système et/ou certaines entrées sont non commandables. La détermination de ce correcteur repose sur des outils de recherche du plus grand point fixe d'applications définies sur des dioïdes.

*Mots-clés*— Graphes d'Événements Temporisés, algèbre ( $Max, +$ ), résiduation, commande, contraintes temporelles.

## I. INTRODUCTION

Les *Systèmes (dynamiques) à Événements Discrets (SED)* sont une classe de systèmes dynamiques qui, par opposition aux systèmes dynamiques continus, satisfont les deux propriétés suivantes:

1. l'*espace d'état* est décrit par un *ensemble discret*,
2. les *transitions d'état* sont observées seulement en des *instants discrets* du temps; ces transitions d'état sont associées à des *événements*.

La représentation en *Réseaux de Petri (RdP)* de ces systèmes permet une bonne compréhension du fonctionnement des *SED*, mais à cause de certaines caractéristiques complexes (concurrency, parallélisme,...) il est difficile d'en déduire un modèle mathématique exploitable.

Cependant, on peut considérer une sous-classe des *RdP* temporisés, où chaque place a exactement une transition en amont et une transition en aval: les *Graphes d'Événements Temporisés (GET)*<sup>1</sup>. Cette sous-classe admet une modélisation mathématique linéaire dans l'*algèbre des dioïdes*, qui

permet d'écrire des équations d'état faisant penser à celles représentant l'évolution des systèmes continus linéaires.

Grâce aux avancées dans ce domaine, il existe aujourd'hui divers outils permettant la modélisation, l'évaluation de performances [3], [4] et l'élaboration de lois de commande pour les systèmes linéaires sur les dioïdes [5], [6].

Nous reprenons ici le problème de synthèse de commande de *GET* abordé dans [1]. Dans cette communication, on a proposé la synthèse d'une loi de commande prenant pour objectif d'approcher au mieux la dynamique d'un modèle de référence tout en satisfaisant une contrainte temporelle. On rencontre ce type de contrainte au sein de certains systèmes manufacturiers [7]. Considérons, par exemple, le cas d'une chaîne de production au milieu de laquelle se trouve un four (voir l'exemple de la section IV). On souhaite limiter le temps de séjour des pièces à la stricte durée de cuisson recommandée, afin de ne pas les détériorer. Ceci revient à limiter le temps de séjour des jetons dans un chemin du *GET* modélisant le système de production.

Dans le présent travail, nous améliorons significativement l'applicabilité de cette méthode en la généralisant selon deux axes:

- on rend possible la prise en compte de plusieurs contraintes temporelles au sein du système,
- des entrées non commandables peuvent de plus être considérées.

Dans la section II, nous ferons d'abord de brefs rappels sur la résolution de certains types d'équations dans les dioïdes. La section III présente une méthode de synthèse de loi de commande, de type précompensateur, en tenant compte de contraintes temporelles sur certains chemins du *GET*. Enfin, la section IV présente un exemple d'application de cette synthèse.

<sup>1</sup>*GET*: voir [2, p. 554] pour une définition plus détaillée.

Avant de présenter le problème, nous rappelons quelques éléments de la théorie des dioïdes et de la résolution d'équations dans cette structure algébrique. Puis nous abordons la mise en équation des **GET** dans le dioïde  $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$ , et nous donnons quelques notions sur la commande des **GET**.

### A. Dioïdes et résolution d'équations

*Définition 1* (Dioïde [8], [4]) Un *dioïde* est un semi-anneau idempotent noté  $(\mathcal{D}, \oplus, \otimes)$ .

- On note  $\mathbf{e}$  l'élément neutre pour la multiplication:  $\forall a \in \mathcal{D}, a \otimes \mathbf{e} = \mathbf{e} \otimes a = a$ ;
- On note  $\varepsilon$  l'élément neutre pour l'addition et absorbant pour la multiplication:  $\forall a \in \mathcal{D}, a \oplus \varepsilon = a$  et  $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$ ;
- La relation d'ordre est définie par:  $a \oplus b = b \Leftrightarrow a \preceq b$ .

*Remarque 1:* Par la suite, le signe ' $\otimes$ ' sera parfois omis ou remplacé par ' $\cdot$ '.

*Définition 2* (Dioïde complet) Un dioïde est dit *complet* si il est fermé pour les sommes infinies et que la multiplication distribue sur les sommes infinies.

*Définition 3* (Dioïde de séries formelles [4]) Une *série formelle* à deux variables  $z_1$  et  $z_2$  commutatives à coefficients dans un dioïde  $\mathcal{D}$  est une application  $f$  de  $\mathbb{Z}^2$  dans  $\mathcal{D}$ : pour  $\{k_1, k_2\} \in \mathbb{Z}^2$ ,  $f(k_1, k_2)$  représente le coefficient de  $\{z_1^{k_1}, z_2^{k_2}\}$ . Une représentation équivalente est:

$$f \stackrel{\text{déf}}{=} \bigoplus_{\{k_1, k_2\} \in \mathbb{Z}^2} f(k_1, k_2) z_1^{k_1} z_2^{k_2} .$$

L'ensemble des séries formelles à deux variables commutatives à coefficients dans un dioïde  $\mathcal{D}$  muni des opérations suivantes:

$$\begin{aligned} f \oplus g &: (f \oplus g)(k_1, k_2) \stackrel{\text{déf}}{=} f(k_1, k_2) \oplus g(k_1, k_2) , \\ f \otimes g &: (f \otimes g)(k_1, k_2) \stackrel{\text{déf}}{=} \bigoplus_{i_\lambda + j_\lambda = k_\lambda} f(i_1, i_2) \otimes g(j_1, j_2) , \end{aligned}$$

$\lambda \in \{1, 2\}$ , est un dioïde noté  $\mathcal{D}[[z_1, z_2]]$ . Si  $\mathcal{D}$  est complet, alors  $\mathcal{D}[[z_1, z_2]]$  l'est également.

De par la structure ordonnée des dioïdes, les résultats concernant les points fixes énoncés sur les treillis s'appliquent. En particulier, le théorème de Tarski [9] permet d'affirmer l'existence d'une plus grande solution à  $f(x) = x$ , où  $f$  est isotone. Le théorème suivant spécifie une méthode classique pour son calcul.

*Théorème 1* (Plus grand point fixe) Soit  $h$  une application isotone d'un dioïde complet  $\mathcal{D}$  dans lui-même, et soit  $w \in \mathcal{D}$ . On définit l'application  $f$  isotone par:  $\forall x \in \mathcal{D}, f(x) = h(x) \wedge w$  (borne inf entre  $h(x)$  et  $w$ ).

Si la suite suivante

$$\begin{aligned} y_0 &= w \\ y_{k+1} &= h(y_k) \wedge w \end{aligned}$$

converge en un nombre fini  $k_f$  d'itérations, alors  $y_{k_f}$  est le plus grand point fixe de  $f$ .

Il n'est pas toujours possible de résoudre  $f(x) = b$  dans un dioïde, car  $f$  n'est pas nécessairement inversible. La

théorie de la résiduation permet de définir une «pseudo-inverse» quand cela est possible. Une théorie complète a été développée, et une spécification aux dioïdes est réalisée dans [4].

*Définitions 4* (Résiduée, résiduée duale)

Soient  $(A, \preceq_A)$  et  $(B, \preceq_B)$  des ensembles ordonnés et  $f : A \rightarrow B$  une application.

4.1  $f$  est *résiduable* si elle est isotone et si l'ensemble  $\{x \in A \mid f(x) \preceq_B b\}$  admet une borne sup pour tout  $b \in B$ , que l'on note  $f^\sharp(b)$ .

4.2  $f$  est *dualement résiduable* si elle est isotone et si l'ensemble  $\{x \in A \mid f(x) \succeq_B b\}$  admet une borne inf pour tout  $b \in B$ , que l'on note  $f^\flat(b)$ .

*Théorème 2:* [4] L'application  $L_a : x \mapsto a \otimes x$  est résiduable. On note  $L_a^\sharp(y) = a \oslash y = \frac{y}{a} = \bigoplus \{x \in \mathcal{D} \mid a \otimes x \preceq y\}$  la résiduée de  $L_a$ .

Le tableau suivant présente un ensemble de propriétés, prouvées notamment dans [4, pp. 180 à 182]. D'autres propriétés sont présentées dans cet ouvrage.

$$a \overline{\frac{x}{a}} \preceq x \quad (2a)$$

$$\overline{\frac{ax}{a}} \succeq x \quad (2b)$$

$$\overline{\frac{x}{ab}} = \frac{a \oslash x}{b} \quad (2c)$$

### B. Modélisation et commande des **GET** dans les dioïdes

Le comportement d'un **GET** peut être mis sous forme d'équations d'état linéaires dans un dioïde de séries formelles à deux variables commutatives  $\gamma, \delta$  à exposants dans  $\mathbb{Z}$  et à coefficients booléens [3]:  $\gamma$  est l'opérateur de *décalage événementiel* tandis que  $\delta$  est l'opérateur de *décalage temporel*. Ce dioïde est noté  $\mathcal{M}_{in}^{ax}[[\gamma, \delta]]$  [4]. On regroupe les séries associées aux tirs des

- $m$  transitions sources dans le *vecteur d'entrée*  $U$ ,
- $p$  transitions puits dans le *vecteur de sortie*  $Y$ ,
- $n$  transitions internes dans le *vecteur d'état*  $X$ .

Les équations d'état se présentent alors sous la forme suivante:

$$\begin{cases} X(\gamma, \delta) &= A \otimes X(\gamma, \delta) \oplus B \otimes U(\gamma, \delta) , \\ Y(\gamma, \delta) &= C \otimes X(\gamma, \delta) . \end{cases}$$

On peut déduire de ces équations d'état une *matrice de transfert*  $H = CA^*B$ , avec  $a^* = \bigoplus_{i \in \mathbb{N}} a^i = \mathbf{e} \oplus \mathbf{a} \oplus \mathbf{a}^2 \oplus \dots$  [4]. Elle représente la relation entrée-sortie du système en fonctionnement «au plus tôt». Plus précisément, les éléments  $H_{ij}$  de la matrice de transfert sont des éléments causaux (voir [4, Déf. 5.35 p. 254]). Le dioïde complet des éléments causaux sera noté  $\mathcal{M}_{in}^{ax+}[[\gamma, \delta]]$  par la suite.

*Remarque 2:* Une définition informelle de la causalité est que les éléments  $H_{ij}$  ne représentent pas d'anticipation événementielle ou temporelle, autrement dit les monômes des séries formelles de  $H$  sont tous à exposants positifs.

*Théorème 3* (Projection causale [5])

L'injection canonique  $I : \mathcal{M}_{in}^{ax+}[[\gamma, \delta]] \rightarrow \mathcal{M}_{in}^{ax}[[\gamma, \delta]]$  est résiduable. Sa résiduée, notée  $\text{Pr}_+$ , est une *projection*, c'est-à-dire:  $\text{Pr}_+ \circ \text{Pr}_+ = \text{Pr}_+$ .

*Corollaire 1:* L'application  $L_a^+ : \mathcal{M}_{in}^{ax+}[\gamma, \delta] \rightarrow \mathcal{M}_{in}^{ax+}[\gamma, \delta], x \mapsto a \otimes x$  est résiduable. Sa résiduée est  $L_a^{+\sharp} : x \mapsto \text{Pr}_+(a \otimes x)$ .

*Proof:*  $a \otimes b$  est la plus grande solution (non nécessairement causale) de  $ax \preceq b$ . Donc  $\text{Pr}_+(a \otimes b)$  est la plus grande solution causale. ■

*Définition 5* (Série réalisable) Une série  $s \in \mathcal{M}_{in}^{ax}[\gamma, \delta]$  est réalisable s'il existe un **GET** dont le transfert est  $s$ . Une définition moins naïve est donnée dans [4, Déf. 5.37 p. 255].

Selon [4, p. 256], une série réalisable est nécessairement causale.

Si on positionne en amont d'un système de transfert  $H$  un système de transfert  $P$ , la dynamique de l'ensemble ainsi configuré s'en trouve modifiée: le transfert global est donné par  $H \otimes P$ . Le but de la *commande* ou de la *correction* d'un **GET** par *précompensation* est de trouver le transfert  $P$  tel que l'ensemble système-précompensateur ait un comportement le plus proche possible d'un *modèle de référence* de transfert  $G_{ref}$ :  $HP \preceq G_{ref}$  [10], [11].

*Théorème 4* (Poursuite de modèle [5]) Soit un système  $H \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{p \times m}$  réalisable et soit un modèle de référence  $G_{ref} \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{p \times m}$  réalisable. Le plus grand précompensateur  $P$  réalisable tel que  $H \otimes P \preceq G_{ref}$  est  $P = \text{Pr}_+(H \otimes G_{ref})$ .

*Proof:* Ce théorème est une application directe du corollaire 1. ■

### III. SYNTHÈSE D'UNE COMMANDE SATISFAISANT DES CONTRAINTES TEMPORELLES

#### A. Présentation du problème

Dans un **GET**, le temps de séjour d'un jeton dans une place peut être très supérieur au délai de cette place (voir l'exemple de la figure 1). En ce sens, le modèle de **GET** n'intègre pas les contraintes temporelles inhérentes aux systèmes dits «à temps critique» ou «à temps contraint». On pense par exemple à un four pour lequel la contrainte est que, sous peine d'endommager les pièces, celles-ci n'y résident pas trop longtemps [7]. Dans la sous-section suivante, on se propose de résoudre ce problème par une loi de commande.

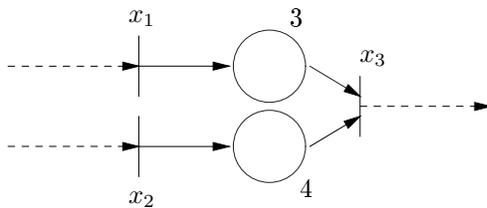


Fig. 1

SI ON NE TIRE PAS  $x_1$  EXACTEMENT 1 UNITÉ DE TEMPS APRÈS  $x_2$ , ALORS LE TEMPS DE SÉJOUR DANS UNE DES PLACES SERA SUPÉRIEUR À LA TEMPORISATION DE CETTE PLACE.

Grâce à la loi de commande, on souhaite limiter le temps de séjour dans certains chemins d'un **GET**. Considérons un chemin  $x_j \rightarrow x_i$  (de la transition  $x_j$  vers la transition  $x_i$ ), contenant initialement  $\alpha_{ij}$  jetons. Le respect d'une contrainte temporelle, notée  $\tau_{ij}$ , sur ce chemin s'exprime

dans l'algèbre classique:

$$\underbrace{x_i(k + \alpha_{ij})}_{\text{date de sortie du jeton}} - \underbrace{x_j(k)}_{\text{date du } k^{\text{ième}} \text{ tir de } x_j} \leq \tau_{ij},$$

introduit par le  $k^{\text{ième}}$  tir de  $x_j$

ou encore:

$$(-\tau_{ij}) + x_i(k + \alpha_{ij}) \leq x_j(k).$$

Soit la contrainte suivante sur les séries de  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$  associées aux transitions:

$$\gamma^{-\alpha_{ij}} \delta^{-\tau_{ij}} x_i \preceq x_j. \quad (3)$$

Dans [1], il a été proposé une approche permettant de synthétiser le plus grand précompensateur tel qu'une unique contrainte temporelle soit respectée et tel que le système corrigé ne soit pas plus lent que le système nominal. Dans ce qui suit, nous étendons ces travaux en deux points:

- d'une part, pour permettre la prise en compte de plusieurs contraintes temporelles,
- d'autre part, le précompensateur proposé ici peut prendre en compte le fait que certaines entrées soient *non commandables* (i.e. leurs dates de tir ne peuvent pas être modifiées).

#### B. Prise en compte des contraintes temporelles

Selon l'inégalité (3), les contraintes temporelles  $\tau_{ij}$  imposées à certains chemins d'un **GET** peuvent se traduire par une inégalité sur le vecteur d'état du système corrigé  $X_c$ :

$$\Delta X_c \preceq X_c, \forall V, \quad (4)$$

où  $\Delta \in \mathcal{M}_{in}^{ax}[\gamma, \delta]^{n \times n}$  est tel que:

$$[\Delta]_{ji} = \begin{cases} \gamma^{-\alpha_{ij}} \delta^{-\tau_{ij}} & \text{s'il existe une contrainte} \\ & \text{sur le chemin } x_j \rightarrow x_i, \\ \varepsilon & \text{sinon.} \end{cases}$$

*Proposition 1:* Le plus grand précompensateur  $P$  tel que le système corrigé respecte la ou les contrainte(s) temporelle(s) spécifiée(s) par  $\Delta$  et tel que son transfert soit inférieur à  $G_{ref}$ , est la plus grande solution de:

$$P \preceq \frac{G_{ref}}{H} \wedge \frac{A^*BP}{\Delta A^*B} \wedge \text{Pr}_+(P). \quad (5)$$

*Proof:* Développons l'inéquation (4):

$$\begin{aligned} \Delta X_c &\preceq X_c, \\ \iff \Delta A^*BPV &\preceq A^*BPV \quad \forall V, \\ \iff \Delta A^*BP &\preceq A^*BP. \end{aligned}$$

Or d'après la théorie de la résiduation,  $ax \preceq b \iff x \preceq a \otimes b$ . Ainsi l'inéquation précédente est équivalente à

$$P \preceq \frac{A^*BP}{\Delta A^*B}. \quad (6)$$

Le précompensateur doit être causal pour être réalisable:  $P = \text{Pr}_+(P)$ . Or on vérifie aisément que  $\text{Pr}_+(P) \preceq P$ , donc on peut considérer la relation suivante comme une contrainte sur  $P$ :

$$P \preceq \text{Pr}_+(P) . \quad (7)$$

En résumé, en considérant (6) et (7) et que  $HP \preceq G_{ref} \Leftrightarrow P \preceq H \oslash G_{ref}$ , on obtient (5). ■

### C. Prise en compte d'entrées non commandables

On qualifie de *commandable* une transition d'entrée d'un GET dont on peut modifier les dates de tir; elle est dite *non commandable* dans le cas contraire.

On partitionne l'ensemble des entrées  $\{u_i\}$  en  $M$  (sous-ensemble des entrées commandables) et  $\overline{M}$  (sous-ensemble des entrées non commandables).

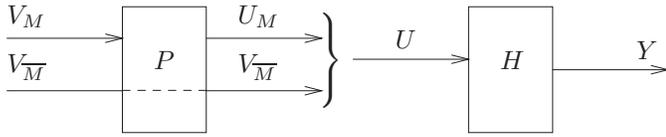


Fig. 2

ENTRÉES COMMANDABLES ET NON COMMANDABLES

La figure 2 schématise le principe de fonctionnement du système corrigé. L'objectif est de synthétiser un précompensateur qui n'agit pas sur les entrées non commandables. Ainsi, les consignes  $v_j$  en entrée du précompensateur  $P$  avec  $j \in \overline{M}$ , sont telles que  $u_j = v_j$  (la consigne est appliquée sans être modifiée pour les entrées non commandables).

Soit  $D \in \mathcal{M}_{in}^{ax}[[\gamma, \delta]]^{m \times m}$  tel que  $D_{ii} = \mathbf{e}$  si  $i \in \overline{M}$ ,  $D_{ij} = \varepsilon$  sinon. Pour respecter le caractère non commandable de certaines entrées, on cherche  $P$  tel que  $DU = DV \forall V$ , i.e.  $DPV = DV \forall V$ , autrement dit on cherche  $P$  tel que  $DP = D$ . Un tel correcteur sera dit *applicable* vis-à-vis de la matrice  $D$ .

*Proposition 2:* Un correcteur applicable vis-à-vis d'une matrice  $D$  vérifie:

$$P \preceq \frac{D}{D} , \quad (8)$$

$$P \succeq D . \quad (9)$$

*Proof:* On remarque que  $D \preceq I_d$  et  $D^2 = D$ . On montre l'équivalence:  $DP = D \Leftrightarrow D \preceq P \preceq \frac{D}{D}$ .

$$\begin{aligned} DP = D &\Leftrightarrow \begin{cases} DP \preceq D, \\ D \preceq DP, \end{cases} \\ &\Rightarrow \begin{cases} P \preceq \frac{D}{D}, \\ D \preceq DP \preceq P \quad \text{car } D \preceq I_d. \end{cases} \end{aligned}$$

⇐

$$\begin{aligned} D \preceq P \preceq \frac{D}{D} &\Rightarrow \underbrace{D^2}_{=D} \preceq DP \preceq \underbrace{D \frac{D}{D}}_{=D} , \\ &\Rightarrow DP = P . \end{aligned}$$

### D. Synthèse de la commande

On cherche le plus grand précompensateur  $P$  réalisable, applicable (voir proposition 2) vis-à-vis d'une matrice  $D$ , tel que le système corrigé respecte la ou les contrainte(s) temporelle(s) spécifiée(s) par  $\Delta$  et tel que son transfert soit inférieur à  $G_{ref}$  (voir proposition 1). Plus formellement, d'après (5), (8) et (9), le précompensateur  $P$  attendu doit vérifier le système suivant:

$$\begin{cases} P \preceq \frac{G_{ref}}{H} \wedge \frac{A^*BP}{\Delta A^*B} \wedge \text{Pr}_+(P) \wedge \frac{D}{D} \\ P \succeq D . \end{cases} , \quad (10)$$

Si on calcule un précompensateur  $P$  respectant (10), alors le système corrigé obtenu est au moins aussi rapide que le modèle de référence de transfert  $G_{ref}$ , mais respectera la ou les contrainte(s) temporelle(s) définies dans  $\Delta$ , sans intervenir sur les éventuelles entrées non commandables définies dans  $D$ .

*Proposition 3:* Si la suite  $y_0, y_1, \dots$ , définie par

$$\begin{aligned} y_0 &= \frac{G_{ref}}{H} \wedge \frac{D}{D} \\ y_{k+1} &= \frac{A^*By_k}{\Delta A^*B} \wedge \text{Pr}_+(y_k) \wedge y_k \end{aligned}$$

converge en un nombre fini  $k_f$  d'itérations et que  $y_{k_f} \succeq D$ , alors  $y_{k_f}$  est le plus grand précompensateur solution de (10).

*Proof:* On peut décomposer le système (10) de la manière suivante:  $x \preceq f(x)$ , où  $x$  est l'inconnue et correspond ici à  $P$ , et  $f : f(x) = h(x) \wedge w$  avec:

$$\begin{aligned} w &= \frac{G_{ref}}{H} \wedge \frac{D}{D} , \\ h(x) &= \frac{A^*Bx}{\Delta A^*B} \wedge \text{Pr}_+(x) . \end{aligned}$$

Alors le théorème 1 nous fournit, si l'algorithme converge et que la solution trouvée est supérieure à  $D$ , le plus grand  $P$  répondant à tous les critères. ■

*Remarques 3:*

3.1 À notre connaissance, l'application  $h$  considérée ne possède pas de propriété suffisante pour garantir la convergence de la suite  $y$ .

3.2 la suite précédente peut être utilisée pour les systèmes totalement commandables en prenant simplement  $D = \varepsilon^{m \times m}$  (les contraintes (8) et (9) sont alors toujours vérifiées).

## IV. EXEMPLE

Nous illustrons notre démarche au travers d'un exemple considéré dans [1] et inspiré du problème réel étudié dans [7], en s'astreignant ici à des hypothèses plus réalistes. Le GET de la figure 3 représente un îlot au sein d'un système de production contenant un four (place  $x_1 \rightarrow x_2$ ) ayant une capacité de 3 pièces et un temps de cuisson de 3 *unités de temps (u.t.)*, suivi d'un tampon de déchargement de 3 places muni d'un convoyeur (temps de transport 4 *u.t.*). ■

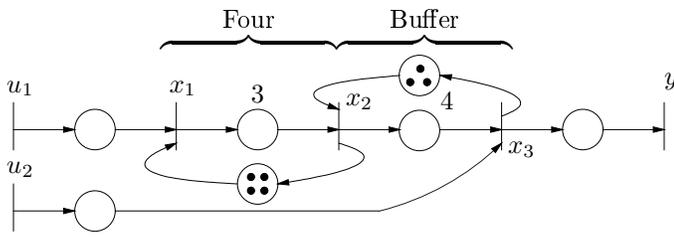


Fig. 3

EXEMPLE DE GET AVEC CONTRAINTE TEMPORELLE.

L'entrée  $u_1$  symbolise l'entrée de pièces dans l'ilôt. L'entrée  $u_2$  représente l'évacuation d'une pièce prise dans le tampon de déchargement.

Pour ce système, on souhaite que les pièces ne restent pas dans le four plus que le temps de cuisson (soit 3 u.t.). De plus, on considère que l'évacuation des pièces du buffer n'est pas commandable: on ne peut pas influencer sur la trajectoire de tir de l'entrée  $u_2$ . On choisit d'accepter un retard de 3 u.t. sur les sorties par rapport au système nominal: on prend comme modèle de référence  $G_{ref} = \delta^3 H$ .

Tout d'abord, mettons en équation ce GET:

$$X = \begin{pmatrix} \varepsilon & \gamma^4 & \varepsilon \\ \delta^3 & \varepsilon & \gamma^3 \\ \varepsilon & \delta^4 & \varepsilon \end{pmatrix} X \oplus \begin{pmatrix} \mathbf{e} & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} \end{pmatrix} U ,$$

$$Y = (\varepsilon \ \varepsilon \ \mathbf{e}) X .$$

D'où

$$H = \begin{pmatrix} \delta^7[\gamma^3\delta^4]^* & [\gamma^3\delta^4]^* \\ \delta^{10}[\gamma^3\delta^4]^* & \delta^3[\gamma^3\delta^4]^* \end{pmatrix} ,$$

$$G_{ref} = \begin{pmatrix} \delta^7[\gamma^3\delta^4]^* & [\gamma^3\delta^4]^* \\ \delta^{10}[\gamma^3\delta^4]^* & \delta^3[\gamma^3\delta^4]^* \end{pmatrix} .$$

Compte tenu de la contrainte temporelle de cuisson et du fait que l'entrée  $u_2$  n'est pas commandable, les matrices  $\Delta$  et  $D$  sont définies par

$$\Delta = \begin{pmatrix} \varepsilon & \delta^{-3} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} ,$$

$$D = \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & \mathbf{e} \end{pmatrix} .$$

L'algorithme converge en une itération:

$$y_0 = \frac{G_{ref}}{H} \wedge \frac{D}{D} = \begin{pmatrix} \delta^3[\gamma^3\delta^4]^* & \delta^{-4}[\gamma^3\delta^4]^* \\ \varepsilon & \mathbf{e} \end{pmatrix} ,$$

$$y_1 = \frac{A^*B}{\Delta A^*B y_0} \wedge \text{Pr}_+(y_0) \wedge y_0 = \begin{pmatrix} \delta^3[\gamma^3\delta^4]^* & \gamma^3[\gamma^3\delta^4]^* \\ \varepsilon & \mathbf{e} \end{pmatrix} ,$$

$$y_2 = \frac{A^*B}{\Delta A^*B y_1} \wedge \text{Pr}_+(y_1) \wedge y_1 = \begin{pmatrix} \delta^3[\gamma^3\delta^4]^* & \gamma^3[\gamma^3\delta^4]^* \\ \varepsilon & \mathbf{e} \end{pmatrix} = y_1 .$$

D'où

$$P = \begin{pmatrix} \delta^3[\gamma^3\delta^4]^* & \gamma^3[\gamma^3\delta^4]^* \\ \varepsilon & \mathbf{e} \end{pmatrix} .$$

$P$  est bien solution de notre problème, puisque  $P \succeq D$ . Une réalisation de ce précompensateur est donnée par la figure 4.

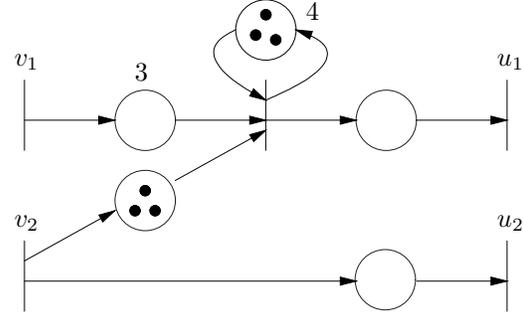


Fig. 4

PRÉCOMPENSATEUR POUR LE GET DE LA FIGURE 3

On remarque que le précompensateur n'influence pas l'entrée  $u_2$ . Le système précompensé est au moins aussi rapide que le modèle de référence:

$$HP = (\delta^{10}[\gamma^3\delta^4]^* \ \mathbf{e} \oplus \gamma^3\delta^7[\gamma^3\delta^4]^*) \preceq G_{ref} .$$

## V. CONCLUSION

La démarche proposée dans ce papier permet de calculer une loi de commande de type précompensation pour un SED pouvant être représenté par un GET. Ensemble, le système et son correcteur ainsi calculé sont au moins aussi rapides que le modèle de référence choisi au préalable. De plus, on s'assure de limiter les temps de séjour des jetons dans certains chemins du GET. D'autre part, la loi de commande prend en compte le cas d'éventuelles entrées non commandables.

L'exemple numérique proposé a été mis en oeuvre à l'aide de la boîte à outils *MinMaxGD*<sup>2</sup> [6, Annexes] pour le logiciel de calcul numérique *Scilab*<sup>3</sup>, permettant de manipuler les séries réalisables dans le dioïde  $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ . Les scripts sont disponibles à l'adresse <http://aurelien.correia.free.fr>. Ces outils rendent aisée l'expérimentation de la commande proposée sur tout système modélisé par un GET. Jusque-là, les expérimentations ont été réalisées seulement pour des systèmes de taille raisonnable (GET contenant moins d'une vingtaine de transitions), et la suite de calcul de la commande (proposition 3) a toujours convergé rapidement (moins de cinq itérations). Le présent papier ouvre plusieurs perspectives de travail. Parmi celles-ci, l'expérimentation de la méthode proposée sur des systèmes réels de taille importante devra être envisagée. Sur un plan théorique, il serait intéressant de mettre à jour des conditions suffisantes pour garantir la convergence de la suite de calcul de la commande.

## RÉFÉRENCES

- [1] Sébastien Lahaye, Bertrand Cottenceau, et Aurélien Corrèia. Commande de graphes d'événements temporisés avec contrainte

<sup>2</sup><http://www.istia.univ-angers.fr/~hardouin/outils.html>.

<sup>3</sup><http://www.scilab.org>.

- de temps critique. *Conférence Internationale Francophone d'Automatique (CIFA)*, 2004.
- [2] Tadao Murata. Petri nets: properties, analysis and applications. *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, 77(4):541–580, April 1989.
  - [3] Guy Cohen, Pierre Moller, Jean-Pierre Quadrat, et Michel Viot. Algebraic tools for the performance evaluation of discrete event systems. *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, 77(1):39–58, 1989.
  - [4] François Baccelli, Guy Cohen, Geert-Jan Olsder, et Jean-Pierre Quadrat. *Synchronization and linearity, an algebra for discrete event systems*. Wiley, 1992.
  - [5] Bertrand Cottenceau, Laurent Hardouin, Jean-Louis Boimond, et Jean-Louis Ferrier. Model reference control for timed event graphs in dioids. *Automatica*, 37:1451–1458, March 2001.
  - [6] Mehdi Lhommeau. *Étude de systèmes à événements discrets dans l'algèbre  $(\max, +)$ : Synthèse de correcteurs robustes dans un dioïde d'intervalles. Synthèse de correcteurs en présence de perturbations*. PhD thesis, Institut des Sciences et Techniques pour l'Ingénieur d'Angers (ISTIA), Université d'Angers, France, 2003.
  - [7] Saïd Amari, Isabel Demongodin, et Jean-Jacques Loiseau. *Sizing, cycle time and control of an industrial plant using dioid algebra*, chapter 6. Kluwer Academic Publishers, 2004.
  - [8] R Cunninghame-Green. Minimax algebra. *Lecture notes in economics and mathematical systems*, volume 166. Springer-Verlag, 1979.
  - [9] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of mathematics*, 5(2):285–309, 1955.
  - [10] Laurent Libeaut et Jean-Jacques Loiseau. On the control of timed event graphs. *International Workshop On Discrete Event Systems (WODES)*, Edimbourg, Scotland, 1996.
  - [11] Laurent Hardouin, Eric Menguy, Jean-Louis Boimond, et Jean-Louis Ferrier. S.i.s.o. discrete event systems control in dioid algebra. *Journal Européen des Systèmes Automatisés (JESA)*, 31:433–452, 1997.