Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri, Tizi-Ouzou

Faculté du Génie Electrique et Informatique

Département d'Automatique

MEMOIRE DE MAGISTER

Spécialité : Automatique

Option: Automatique des Systèmes Continus et Productiques

Présenté par :

M^{elle} TEBANI Karima

Thème :

Commande des systèmes à événements discrets à temps critiques, application aux systèmes de commande en réseau

Soutenu le 10 /10 /2012, devant le jury d'examen composé de :

M. DJENNOUNE Saïd	**Professeur à l'UMMTO**	**Président**
M. KARA Redouane	**M.C, Classe A à l'UMMTO**	**Rapporteur**
M. AMARI Saïd	**M.C à l'université Paris XIII**	**Examinateur**
M. MAIDI Ahmed	**M.C, Classe A à l'UMMTO**	**Examinateur**
M. OUKACHA Brahim	**M.C, Classe A à l'UMMTO**	**Examinateur**
M. COLLART-DUTILLEUL Simon	**HDR à l'école centrale de Lille**	**Invité**

Remerciements

Ce travail a été effectué au sein du Laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), de l'université Mouloud Mammeri de Tizi-Ouzou.

Je tiens tout d'abord à remercier Monsieur *KARA Redouane*, Maitre de conférences Classe A 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.

Je tiens à exprimer ma reconnaissance à Monsieur *AMARI Saïd*, Maitre de conférences à l'université Paris XIII, pour sa patience, ses conseils et remarques pertinentes, et pour son aide tout au long de ma thèse, et pour avoir accepté d'examiner ce travail et de faire partie du jury de ce mémoire.

Je remercie Monsieur DJENNOUNE Saïd, Professeur à l'université Mouloud Mammeri de Tizi Ouzou, pour avoir accepté d'examiner ce travail et de présider le jury de soutenance de ce mémoire.

Je tiens également à remercier Monsieur *MAIDI Ahmed* Maitre de conférences Classe A à l'université Mouloud Mammeri de Tizi Ouzou, et Monsieur *OUKACHA Brahim* Maitre de conférences Classe B à l'université Mouloud Mammeri de Tizi Ouzou, qui ont accepté d'être membres du jury de soutenance.

Mes remerciements s'adressent également à tous les membres du laboratoire de Conception et Conduite des Systèmes de Production (L2CSP), et tous mes collègues pour l'excellente ambiance de travail qu'ils ont créé.

Je remercie tous mes amis, qui de près ou de loin m'ont supportée, soutenue et encouragée tout au long de ces années.

Mes derniers remerciements, et non des moindres, je les destine à mes parents, sans qui rien ne serait, et à mes deux frères et ma sœur, pour leur disponibilité et leurs encouragements.

Sommaire

Table des figures

1 INTRODUCTION

Sous l'appellation *Systèmes à Evénements Discrets* (SED) sont regroupés certains systèmes, généralement de conception humaine. On peut par exemple citer les systèmes de production (ateliers flexibles, lignes d'assemblage) [11], [17] les réseaux de communication (réseaux informatiques) [18] et les systèmes de transport (routier, ferroviaire ou aérien), [19]. Le comportement dynamique de cette classe de système ne peut pas être décrit par des équations différentielles ou aux différences comme dans les systèmes continu classique. L'importance prise par ces systèmes dans notre société a conduit de nombreux chercheurs à proposer des modèles mathématiques permettant de décrire leur comportement afin d'en évaluer les performances et d'optimiser leur conception ou leur pilotage. On cite par exemple les modèles à base d'automates à états finis, et l'algèbre des dioïdes. Ces systèmes sont plus fréquents dans le domaine industriel. En outre les applications dans ce domaine posent un autre problème qui réside dans l'existence des contraintes temporelles dont il faut tenir compte. Ces contraintes peuvent prendre diverses formes (échéance, intervalle de temps, durée de validité, etc.) et s'appliquer à des objets variés.

Ce problème de contraintes temporelles est très fréquent dans les systèmes de production incluant des traitements thermique ou chimique [20], dans les systèmes embarqués [21], dans les réseaux de transport urbain ou ferroviaire [22,23] et dans les systèmes de commande en réseau [24]. Dans ce travail, nous nous intéressons à cette dernière catégorie de systèmes à temps critique.

Les Architectures d'Automatisation distribuées sur des Réseaux (AAR) sont aujourd'hui présentes dans de nombreux secteurs d'activités : dans l'industrie manufacturière par exemple, au travers des systèmes de contrôle de procédés (usines, centrales nucléaires), dans l'aéronautique au travers des processus de pilotage d'avions, et de satellites et dans les applications de supervision ou d'e-maintenance sur des longues distances. Il est à noter que le facteur temps dans ces AAR est une composante primordiale. Le respect des contraintes temporelles est aussi important que l'exactitude du résultat. Autrement dit, le système ne doit pas simplement délivrer des résultats exacts, mais il doit les délivrer dans des délais imposés. Chaque performance temporelle des AAR, en particulier le temps de réponse, est caractérisée non pas par une valeur unique mais par une distribution de valeurs [6], [25-27]. Ceci provient des mécanismes de consommation de temps et des retards (traitement de données, synchronisation entre processus) au sein de l'architecture qui induisent des délais variables.

Nous considérons dans notre cas une AAR qui comporte une contrainte temporelle (borne maximale) sur son temps de réponse à ne pas dépasser. Il est donc nécessaire, de disposer de techniques et d'outils qui permettent d'assurer ces spécifications temporelles dans ces systèmes de commande en réseau.

Nous formulons la question en terme d'un problème de commande qui consiste à calculer en temps réel des lois de commande en fonction des paramètres de l'AAR pour garantir cette contrainte temporelle stricte.

Plusieurs travaux sur la commande des systèmes à événements discrets temporisés ont été développés. On peut citer l'approche publiée dans [28], qui est basée sur les automates temporisés, et d'autres méthodes formelles qui utilisent les graphes d'événements temporisés et les algèbres Max-Plus et Min-Plus, qui sont présentées dans [35],[29],[30-33]. Les auteurs de [9], [34], s'intéressent plus précisément à des problèmes de rejet de perturbations et de poursuite de modèle, et la contrainte temporelle est apparue comme une condition supplémentaire. Par contre, ceux de [31-33] ont travaillé sur la commande des systèmes à événements discrets sous contraintes temporelles strictes. Ils calculent en ligne des lois de commande pour satisfaire ces contraintes. C'est pour cette raison que nous choisissons la méthode développée et détaillée dans [31,32] pour résoudre le problème de commande de l'AAR sous contrainte temporelle.

Ce mémoire est décomposé en quatre 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 graphes d'événements temporisés. Et dans la dernière partie de ce chapitre nous montrons comment décrire le comportement d'un graphe d'événement temporisé par des équations linéaire dans l'algèbre Max-Pus et l'algèbre Min-Plus.

Dans le deuxième chapitre, le fonctionnement et la modélisation de l'AAR sont présentés, nous représentons tout d'abord de manière détaillée les composantes et le fonctionnement de l'architecture d'automatisation en réseau dans laquelle les contrôleurs logique communiquent avec les modules d'entrées/sorties via un réseau de communication TCP/IP. La deuxième partie de ce chapitre est consacrée à la modélisation de cette AAR avec un graphe d'événement temporisé et la représentation de la dynamique de cette AAR dans l'algèbre Max-Plus et Min-Plus.

Dans le troisième chapitre nous calculons les lois de commande qui vont satisfaire les contraintes temporelles associe a l'AAR après avoir présenté de manière plus détaillé la

méthode de calcule de ces lois de commande dans la première partie de ce chapitre. Et après avoir présenté dans la dernière partie de chapitre trois les inconvénients de la première modélisation ainsi que la non satisfaction de résultats obtenus pour le calcul des lois de commande, nous représentons une nouvelle modélisation qui fonctionne suivant un réseau de communication producteur/consommateur ensuite on calcule les lois de commande qui vont satisfaire les contraintes temporelles représentées sur cette nouvelle modélisation.

Enfin, une synthèse des résultats obtenus ainsi que quelques perspectives envisagées pour les travaux futurs sont données pour conclure ce travail.

Chapitre 1

Modélisation et analyse des systèmes à événements discrets dans les dioïdes

Dans la première partie decette section, nous présentons des outils algébriques permettant de traiter une classe particulière de réseau de Petri (Rdp). Nous rappelons également la manière avec laquelle ces outils algébriques sont utilisés. Plusieurs propriétés utiles pour faire de l'analyse et de la commande à travers un modèle graphique ou algébrique. Dans la deuxième section nous présentons les modèles de graphe d'événements temporisé ainsi que la

représentation de leur dynamique dans les dioïdes. La plupart de notions de cette partie sont issues de [1-5].

2.1Théorie des dioïdes

Dans cette section un rappel sur les structures algébriques est considéré. Le but de cette section est d'introduire les concepts et notations qui seront utiles pour l'étude de la commande de graphes d'événements temporisés.

2.1.1 Rappels algébrique

Définition 2.1(Monoïde) Un monoïde notée (D,$\oplus$) est un ensemble D muni d'une loi de composition interne notée $\oplus$ associative : $\forall a, b, c \in D, (a \oplus b) \oplus c = a \oplus (b \oplus c)$, et d'un élément neutre ε tel que, $\forall a \in D, a \oplus \varepsilon = \varepsilon \oplus a = a$. Le monoïde est dit commutatif si la loi $\oplus$ est commutative c'est-à-dire :$\forall a, b \in D, a \oplus b = b \oplus a$.

Exemple2.1 L'ensemble des entiers naturels N muni de l'addition est un monoïde. Il s'agit d'un monoïde commutatif : $\forall\, a, b \in N, a + b = b + a$. L'élément neutre *de N* est 0.

Définition 2.2 (Semi-anneau, dioïde). On appelle semi-anneau un ensemble D, muni de deux lois internes$\oplus$ et$\otimes$, tel que :

• (D,$\oplus$) est un monoïde commutatif dont l'élément neutre ε est appelé élément nul.

• (D, $\otimes$) est un monoïde. Son élément neutre est appelé unité et est noté *e*.

• La loi multiplicative $\otimes$ est distributive à droite et à gauche par rapport à la loi additive $\oplus$, c'est-à-dire $\forall a, b, c \in D, a \otimes (b \oplus d) = (a \otimes b) \oplus (a \otimes d)$

$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c).$$

• L'élément neutre ε est absorbant pour la loi $\otimes$ ($\forall a \in D, a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$). Si en outre la loi $\oplus$est idempotente, alors (D, $\oplus$,$\otimes$) est qualifié de semi-anneau idempotent ou dioïde.

Remarque 2.1 La loi $\oplus$ est idempotent si $\forall\, a \in D$,a $\oplus$ a = a.

Exemples 2.2

- On peut vérifier aisément que :$(\mathbb{R} \cup \{-\infty, +\infty\}, max, +)$ est un dioïde commutatif pour lequel $\varepsilon = -\infty$ et e = 0. Ce dioïde est noté R_{max}, et traditionnellement appelé "algèbre

(max,+)". Dans ce dioïde, la loi $\oplus$ correspond à l'application maximum et la loi $\otimes$ est la somme usuelle.

Notons toutefois par hypothèse que : $\varepsilon \otimes (+\infty) = (-\infty) + (+\infty) = \varepsilon = (-\infty)$ dans le dioïde $\mathbb{R}_{\max}$

- $(\mathbb{R} \cup \{-\infty, +\infty\}, min, +)$ est un dioïde commutatif pour lequel $\varepsilon = +\infty$; et e = 0. Ce dioïde est noté $\mathbb{R}_{\min}$, et traditionnellement appelé "algèbre (min; +)".

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

2.1.2Propriétés des dioïdes

2.1.2.1 Dioïde complet Un dioïde$(D, \oplus, \otimes)$est dit complet s'il est fermé pour les sommes infinies et si 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$A \subset D$,

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

Il résulte de cette définition que, pour tout $A \subset Det B \subset D$:

$(\oplus_{a \in A}\, \boldsymbol{a}) \otimes (\oplus_{\boldsymbol{b} \in \boldsymbol{B}}\, \boldsymbol{b}) = \oplus_{(\boldsymbol{a},\boldsymbol{b}) \in \boldsymbol{A} \times \boldsymbol{B}}\, (\boldsymbol{a} \otimes \boldsymbol{b})$**.**

Exemple 2.3 (dioïde complet de $\mathbb{R}_{\max}$).$\mathbb{R} \cup \{\pm\infty\}$, muni du max et du +, avec la convention: $(+\infty) + (-\infty) = -\infty$, est un dioïde complet que l'on notera $\overline{\mathbb{R}}_{max}$.

2.1.2.2 Relation d'ordre dans un dioïde

Définition 2.3 (Relation d'ordre). Une relation R sur un ensemble 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 $xRy\ et\ yRz,\ alors\ xRz$

-R est antisymétrique : si $xRy\ et\ yRx,\ alors\ x = y$.

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

De plus, cette relation d'ordre est compatible avec les lois de structure de D, c'est-à-dire, $\forall (a, b, c) \in D^3, a \leq b \Rightarrow a \oplus c \leq b \oplus c$

$$a \leq b \Rightarrow a \otimes c \leq b \otimes c\ et\ c \otimes a \leq c \otimes b$$

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

Exemple 2.4 La relation $\leq$, associée à l'application **max** est une relation d'ordre qui correspond à l'ordre usuel $\leq$, $a \leq b \Leftrightarrow b = \max(a, b)$.

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

2.1.2.3 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 : $(A \oplus B)_{ij} = A_{ij} \oplus \mathrm{B_{ij}}$.

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.B, avec des coefficients qui sont donnés comme suit:$(A \otimes B)_{ij} = \bigoplus_{k=1}^{n} (A_{ik} \otimes \mathrm{B_{kj}})$.

Exemple 2.5

Soit A ,B deux matrices dans l'algèbre (max,+) tel que

$A = \begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix}$ 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

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

2.1. 3 Résolution d'équation dans les dioïdes :

Dans cette partie on s'intéresse à la résolution d'équation de type

$f(x) = b,\ b \in D$ (2.1)

Les lois $\oplus$ et $\otimes$ n'étant pas inversibles, en particulier les applications matricielles. Il n'est pas possible, en général d'inverser les applications définies sous forme analytique dans un dioïde, alors la résolution de l'équation (2.1) pose un problème d'inversion d'application. La théorie de la résiduation permet cependant de définir des pseudo-inverses pour des applications définies sur des dioïdes. Par conséquent, elle permet d'établir, lorsqu'elles

existent des solutions extrêmes de l'équation précédente. On s'intéresse donc à la plus grande solution de l'inéquation $f(x) \leq b$ ou la plus petite solution de l'inéquation$f(x) \geq b$.

Définition 2.4 [3] (Etoile de Kleene)[1],[2]. Soit D un dioïde complet et *a* un élément de D, l'étoile de Kleene de a, noté a^*, est définie comme suit :$a^* = \bigoplus_{k \in N} a^k$ 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 : $A^* = \bigoplus_{i \in N} A^i$ avec $A^0 = I_n$.

Théorème ([1,2]) Dans un dioïde complet, la quantité(A^*B) est la plus petite solution de l'équation $x = Ax \oplus B$ et de l'inéquation $x \geq Ax \oplus B$.

2.2 Propriété spectrales des matrices définies dans un dioïde :

Les matrices carrées à coefficients dans un dioïde présentent certaines propriétés spectrales intéressantes qui permettent notamment d'étudier le comportement asymptotique de certains systèmes dynamiques. Plus particulièrement, pour chaque matrice carrée, on peut lui associer un graphe valué orienté, appelé graphe de précédence. Si un graphe de précédence est fortement connexe, alors la matrice qui lui correspond est dite irréductible.

2.2.1Rappel sur les graphes

Un graphe orienté est défini par un ensemble de nœuds interconnectés par des arcs orientés. Un graphe est dit valués, si des poids positifs sont associés aux arcs qui reliant les nœuds *j* et les nœuds *i*. Ces poids correspondent aux termes notés A_{ij}, de la matrice A. Un chemin orienté est une succession de nœuds que l'on peut parcourir en empruntant les arcs orientés du graphe. Un chemin dont le nœud de départ coïncide avec le nœud d'arrivée est appelé circuit. Ce circuit est dit élémentaire s'il ne contient pas d'autre circuit.

Définition 2.4(Graphe de précédence)[4]. On appelle graphe de précédence d'une matrice $A \in D^{n \times n}$ noté G(A), le graphe qui est composé de n nœuds et des arcs, notés (j, i), qui sont pondérés par le coefficient A_{ij}. Si $A_{ij} \neq \varepsilon$, alors il existe un arc qui relie le nœud *j* au nœud *i*, sinon l'arc (j, i), n'existe pas. Dualement, pour tous graphes orientés valués composés de *n* nœud, on peut associer une matrice carrée de dimension $n \times n$, telle que les coefficients de cette matrice correspondent aux poids des arcs du graphe.

Définition 2.5 (graphe fortement connexe)[5]. Un graphe G(A) est dit fortement connexe si pour toute paire de nœuds$(i,j) \in G(A)$, il existe un chemin orienté allant du nœud i au nœud j.

Définition 2.6 (matrice irréductible)[5]. Une matrice $A \in D^{n\times n}$ est dite irréductible si pour toute paire de nœuds (j, i), il existe un entier k tel que$(A^k)_{ij} \neq \varepsilon$. Cette matrice admet une unique valeur propre notée $\lambda \in D$, et donnée par :$\lambda = \bigoplus_{k=1}^{n} (traceA^k)^{1/k}$.

Avec $traceA^k = \bigoplus_{i=1}^{n} (A^k)_{ii}$ et n désigne l'ordre de la matrice A.

Dans l'algèbre usuelle, l'expression de la valeur propre λ s'écrit comme suit :

$$\lambda = max_{k=1}^{n}[\frac{1}{k}(max_{i=1}^{n}(A^k)_{ii})].$$

Le graphe de précédence associé à une matrice réductible n'est pas fortement connexe. Il est décomposable, dans ce cas, en plusieurs sous graphes fortement connexe. Cette matrice réductible, peut avoir plusieurs valeurs propres, en la décomposant en blocs irréductibles.

La détermination des vecteurs propres d'une matrice A définie sur un dioïde, revient à résoudre l'équation suivante :$Au = \lambda u$.

On appelle u le vecteur propre associé à la valeur propre λ. Cette valeur propre correspond au maximum des poids moyens des circuits élémentaires du graphe de précédence G(A). Le vecteur propre associé à cette unique valeur propre, ne contient pas de ε. Si une composante du vecteur u est égale à ε, il faut que la matrice A contiennent au moins une ligne avec des ε, ce qui correspondra à un graphe G(A) non fortement connexe.

Exemple 2.6. Nous considérons le graphe orienté et valué de la Figure 2.1suivante:

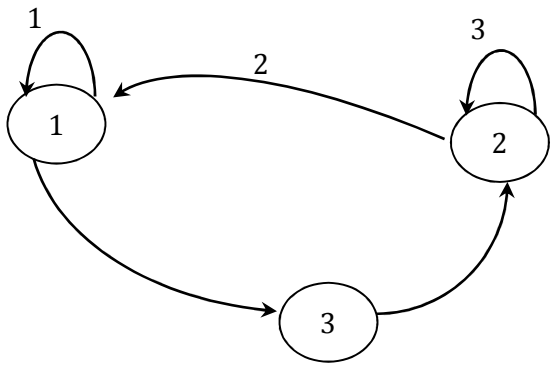


Figure2.1 Graphe orienté et valué

$A = \begin{pmatrix} 1 & 2 & \varepsilon \\ \varepsilon & 3 & e \\ \varepsilon & \varepsilon & e \end{pmatrix}$ est la matrice de précédence associée au graphe de la Figure 2.1

Le graphe de la Figure 2.1 est composé de trois nœuds et pour chaque paire de nœuds, il existe toujours un chemin orienté qui relie ses noeuds. Le graphe est donc fortement connexe, par conséquent sa matrice de précédence associée A est irréductible. La valeur propre de cette matrice est unique, elle est notée λ et est donnée par : $\lambda = \bigoplus_{k=1}^{3} (traceA^k)^{1/k}$
Dans l'algèbre (max,+) on a :

$$A^2 = \begin{pmatrix} \max(1+1, 2+\varepsilon, \varepsilon+e) & \max(1+2, 2+3, \varepsilon+\varepsilon) & \max(1+\varepsilon, 2+e, \varepsilon+\varepsilon) \\ \max(\varepsilon+1, 3+\varepsilon, e+e) & \max(\varepsilon+2, 3+3, e+\varepsilon) & \max(\varepsilon+\varepsilon, 3+e, e+\varepsilon) \\ \max(e+1, \varepsilon+\varepsilon, \varepsilon+e) & \max(e+2, \varepsilon+3, \varepsilon+\varepsilon) & \max(e+\varepsilon, \varepsilon+e, \varepsilon+\varepsilon) \end{pmatrix}$$

$A^2 = \begin{pmatrix} 2 & 5 & 2 \\ 0 & 6 & 3 \\ 1 & 2 & \varepsilon \end{pmatrix}, A^3 = \begin{pmatrix} 3 & 8 & 5 \\ 3 & 9 & 6 \\ 2 & 5 & 2 \end{pmatrix}.$

$$\lambda = (\bigoplus_{i=1}^{3} A_{ii}) \oplus \frac{1}{2}(\bigoplus_{i=1}^{3} (A^2)_{ii}) \oplus \frac{1}{3}(\bigoplus_{i=1}^{3} (A^3)_{ii})$$

$$\lambda = (1 \oplus 3 \oplus \varepsilon) \oplus \frac{1}{2}(2 \oplus 6 \oplus \varepsilon) \oplus \frac{1}{3}(3 \oplus 9 \oplus 2) = (3 \oplus \frac{6}{2} \oplus \frac{9}{3}) = 3$$

Le vecteur propre, noté *u*, associé à la valeur propre λ vérifie l'équation suivante :
$Au = \lambda u$ le calcul du vecteur propre ce fait comme suit :

$$\begin{pmatrix} 1 & 2 & \varepsilon \\ \varepsilon & 3 & 0 \\ \varepsilon & \varepsilon & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = 3 \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \Longrightarrow \begin{pmatrix} \max(1+u_1, 2+u_2, \varepsilon+u_3) \\ \max(\varepsilon+u_1, 3+u_2, 0+u_3) \\ \max(\varepsilon+u_1, \varepsilon+u_2, 0+u_3) \end{pmatrix} = \begin{pmatrix} 3+u_1 \\ 3+u_2 \\ 3+u_3 \end{pmatrix}$$

On trouve le vecteur propre suivant :

$u = \begin{pmatrix} 3 \\ 4 \\ 0 \end{pmatrix}.$

2.3 Graphes d'événements temporisés

Les réseaux de Petri sont un langage graphique permettant de modéliser, d'analyser et de commander les systèmes à événements discrets. Ils ont été introduits en 1962 par le mathématicien Carl Adam Petri, dans le cadre de la théorie des automates.

Les réseaux de Petri comportent deux sortes de nœuds : les transitions et les places. Des arcs relient certaines transitions à certaines places ou certaines places à certaines transitions. Le sens de

parcours des arcs définit les places amont et aval d'une transition. Les jetons présents dans les places constituent le marquage du réseau qui caractérise l'état du système représenté; ce marquage évolue en franchissant les transitions du réseau. Lorsque chacune des places en amont d'une transition contient au moins un jeton, celle-ci est activée. Cette activation entraîne le retrait d'un jeton dans les places amonts et l'ajout d'un jeton dans l'ensemble des places avals. L'étude de la dynamique des systèmes synchronisés nécessite l'introduction du temps dans les réseaux de Petri dit alors temporisés.

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). Les graphes d'événements temporisés admettent une représentation linéaire sur une structure algébrique de dioïde, les résultats sont bien connus [13-15] et [5]. Concernant les graphes d'événements temporels, une modélisation dans l'algèbre des dioïdes a été faite par [16].

2.3.1 Définition d'un graphe d'événements temporisés

Un graphe d'événements 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 p_{ij}, la place qui relie la transition t_j à t_i. si cette place existe, la temporisation correspondante est notée τ_{ij} et son marquage est noté m_{ij}. Si quelque soit l'évolution du graphe considéré le plus grand marquage des places est 1, on dit que le graphe est binaire ou sauf.

2.3.2 Propriétés des graphes d'événements :

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

Proposition 2.1. [5] Dans un graphe d'événements, le nombre de jetons d'un circuit élémentaire est constant.

Démonstration 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.

Proposition 2.2 [5] Soit R un graphe d'événements et M_0 son marquage initial, alors (R, M_0) est vivant si et seulement si tout circuit élémentaire contient au moins une place initialement marquée.

Démonstration Supposons qu'un circuit élémentaire d'un graphe d'événements ne soit pas initialement marqué. En référence à la proposition 1, ce circuit ne contiendra jamais de jeton et donc toutes ses transitions sont en permanence non franchissables : le graphe d'événements n'est donc pas vivant. Inversement, dans un graphe d'événements non vivant, une transition morte (qui n'est jamais tirée) possède obligatoirement en amont une transition également morte. En remontant ainsi d'une transition morte à une autre située en amont, on aboutit inéluctablement à une transition appartenant à un circuit, circuit qui est donc nécessairement sans jeton.

Exemple 2.7

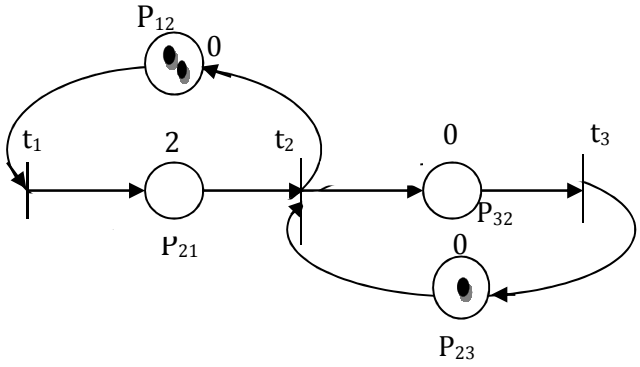


Figure 2.2 Exemple d'un graphe d'événements temporisé

Dans ce graphe il existe deux circuits élémentaire $\gamma_1 = (t_1, P_{21}, t_2, P_{12}, t_1)$, $\gamma_2 = (t_2, P_{32}, t_3, P_{23}, t_2)$.

2.3.3 Le temps de cycle d'un graphe d'événements temporisé

Le temps de cycle dans un système de production correspond à la durée moyenne nécessaire pour produire une seule pièce. Le temps de cycle λ d'un graphe modélisant un processus de production est caractérisé par l'équation suivante :

$$\lambda = \max_{\gamma \in \delta} \frac{T(\gamma)}{M(\gamma)}$$

Où : δ est l'ensemble des circuits élémentaire de GET considéré, $T(\gamma)$ est la somme des temporisations les long de circuit γ et $M(\gamma)$ est le nombre de jetons dans les places du circuit γ.

Nous calculons le temps de cycle du graphe d'événements temporisé de la Figure 2.2, en utilisant l'expression donnée précédemment. Nous décomposons le graphe d'événement temporisé de la Figure 2.2 en circuits élémentaires. On obtient 2 circuits, qu'on note γ_i pour i=1 à 2. Ces circuits sont donnés comme suit :

γ_1=(t_1, P_{21}, t_2, P_{12}, t_1),

γ_2=(t_2, P_{32}, t_3, P_{23}, t_2).

En appliquant la formule du temps du cycle donnée précédemment, le temps de cycle λ du graphe est donné par :

$$\lambda = \max_{\gamma \in \delta} = \frac{T(\gamma)}{M(\gamma)} = \max\left(\frac{2}{2}, \frac{0}{1}\right) = \max(1,0) = 1$$

Le temps de cycle du circuit élémentaire γ_1 est égal au temps de cycle λ.

2.3.4 Représentation de la dynamique

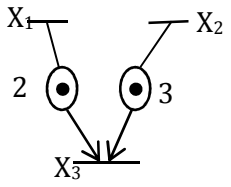
Les graphes d'événements temporisés sont utilisés comme outil de modélisation intermédiaire. Si cette modélisation graphique constitue une première étape, la seconde étape est la mise en équation du modèle graphique, c'est-à-dire la définition d'une représentation analytique du système. Le comportement d'un graphe d'événements temporisés peut s'exprimer sous forme d'équations linéaires dans les dioïdes $\overline{\mathrm{R}}_{\min}$ ou $\overline{\mathbb{R}}_{max}$. Ces équations représentent l'évolution du graphe dans les domaines temporel et évènementiel. Si on considère τ_{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 marquage initial noté m_{ij}. La plus grande temporisation du graphe d'événements considéré est notée τ^{max}, définie par : $\tau^{max} = \max(\tau_{ij})$. Le plus grand marquage initial des places du graphe d'événements considérés est noté m^{max} et est défini par : $m^{max} = \max(m_{ij})$.

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.

2.3.4.1 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 $\theta(t) \in \overline{\mathbb{R}}_{min}^{n}$ cette fonction est appelée compteur. Les compteurs correspondants aux transitions source sont les composantes du vecteur $u(t) \in \overline{\mathbb{R}}_{min}^{m}$.

Exemple



$$x_3(t) \leq \min(1 + x_1(t-2), 1 + x_2(t-3)).$$

Plus généralement, le comportement d'un graphe d'événements temporisé est représenté par l'inéquation suivante :

$$\theta(t) \leq \oplus_{\tau=0}^{\tau\,\max}\left(A_\tau.\theta(t-\tau) \oplus \mathrm{B}_\tau.u(t-\tau)\right).$$

où $A_\tau \in \overline{\mathbb{R}}_{min}^{n\times n}$ 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 ε sinon. Les termes des matrices $B_\tau(k) \in \overline{\mathbb{R}}_{min}^{n\times m}$ 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 dés qu'elle est franchissable. Cette évolution correspond à la solution maximale de l'inéquation précédente dans $\overline{\mathbb{R}}_{min}$. Cette solution satisfait l'équation linéaire suivante :

$$\theta(t) = \oplus_{\tau=0}^{\tau\,\max}\left(A_\tau.\theta(t-\tau) \oplus \mathrm{B}_\tau.u(t-\tau)\right). \qquad (2.2)$$

L'équation (2.2) est implicite en général. Elle est souvent remplacée par la solution explicite suivante :

$$\theta(t) = \oplus_{\tau=1}^{\tau\,\max}(A_0^*.A_\tau\theta(t-\tau) \oplus A_0^* B_\tau u(t-\tau)) \qquad (2.3)$$

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

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

Par analogie avec la théorie des systèmes linéaires classiques, l'équation explicite (2.3) 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 τ places temporisées à 1. Nous ajoutons donc (τ-1) transitions intermédiaires. On associe des compteurs à ces transitions intermédiaires, au nombre de *n'* qui sont les composantes du vecteur $\bar{\theta}(t) \in \overline{\mathbb{R}}_{min}^{n'}$. Le vecteur d'état résultant, noté *x(t),* appartient à $\overline{\mathbb{R}}_{min}^{N}$, avec *N= n +n'*, et est défini par :

$$x(t) = \begin{pmatrix} \theta(t) \\ \bar{\theta}\,(t) \end{pmatrix}.$$

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

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

Qui peut s'écrire sous la forme explicite :

$$x(t) = A.x(t-1) \oplus B.u(t), \qquad (2.4)$$

avec $A = \hat{A}_0^*.\hat{A}_1$ $etB = \hat{A}_0^*.\hat{B}$

Propriété 2.1[4] pour un graphe d'événements temporisé avec des temporisations commensurables, l'équation d'état (2.4) est équivalente à la formulation suivante :

$$x(t) = A^{\tau}.x(t-\tau) \oplus [\oplus_{k=0}^{\tau-1} A^k.B.u(t-k)], \qquad (2.5)$$

Pour tout τ 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 (2.4). Il est claire que la propriété est vérifiée pour $\tau = 1$. Supposons qu'elle est aussi vérifiée pour $\tau = K$, c'est-à-dire :

$$x(t) = A^k.x(t-k) \oplus [\oplus_{k=0}^{K-1} A^k.B.u(t-k)]. \qquad (2.6)$$

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

$$x(t-k) = A.x(t-k-1) \oplus B.u(t-k).$$

En remplaçant x(t-k) par son expression dans (2.5), nous obtenons alors

$$x(t) = A^{k+1}.x(t-(k+1)) \oplus [\oplus_{k=0}^{K-1} A^k.B.u(t-k)].$$

2.3.4.2 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 $\theta(k) \in \overline{\mathbb{R}}_{max}^{n}$ cette fonction est appelée dateur. Les dateurs correspondants aux transitions source sont les composantes du vecteur $u(k) \in \overline{\mathbb{R}}_{max}^{m}$.

La dynamique d'un graphe d'événements temporisé est représentée par l'inéquation suivante :

$$\theta(k) \geq \oplus_{\mathrm{l}=0}^{\mathrm{m\ max}}\left(A_l.\theta(k-l) \oplus \mathrm{B_l}.u(k-l)\right).$$

où $A_l \in \overline{\mathbb{R}}_{max}^{n\times n}$ est la matrice dont le terme $A_{l,ij}$ est égal à τ_{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 ε. Les termes des matrices $B_l \in \overline{\mathbb{R}}_{max}^{n\times m}$ 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 dés qu'elle est franchissable. Cette évolution correspond à la solution minimale de l'inéquation précédente dans $\overline{\mathbb{R}}_{max}$. Cette solution satisfait l'équation linéaire suivante :

$$\theta(k) = \oplus_{\mathrm{l}=0}^{\mathrm{m\ max}}\left(A_l.\theta(k-l) \oplus \mathrm{B_l}.u(k-l)\right) \qquad (2.7)$$

L'équation (2.7) est implicite en général. Elle est souvent remplacée par sa solution explicite suivante :

$$\theta(k) = \oplus_{l=1}^{m\ \max}(A_0^*.A_l\theta(m-l) \oplus A_0^* B_l u(m-l)) \qquad (2.8)$$

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

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

Par analogie avec la théorie des systèmes linéaires classiques, l'équation explicite (2.8) 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 $\bar{\theta}(k) \in \overline{\mathbb{R}}_{max}^{n''}$. Le vecteur d'état résultant, noté *x(k)*, appartient à $\overline{\mathbb{R}}_{max}^{N}$, avec *N= n +n''*, et est défini par :

$$x(k) = \begin{pmatrix} \theta(k) \\ \bar{\theta}\ (k) \end{pmatrix}.$$

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

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

Qui peut s'écrire sous la forme explicite suivante :

$$x(k) = A.x(k-1) \oplus B.u(k), \qquad (2.9)$$

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

Propriété 2.2 on considère un graphe d'événements temporisé dont le comportement est représenté par l'équation (2.9). Cette représentation d'état est équivalente à la formulation suivante :

$$\tilde{x}(k) = \tilde{A}^{\varphi}x(k-\varphi) \oplus [\oplus_{l=0}^{\varphi-1} \tilde{A}^{l}.\tilde{B}.\tilde{u}(k-l)]$$

Pour tout φ entier tel que $\varphi \geq 1$.

2.3.5Exemple

Nous considérons le graphe d'événements P-temporisé de la Figure 2.3. Ce graphe comporte une transition source, nous avons aussi, $\tau^{max} = 4,\ et\ m^{max} = 2$.

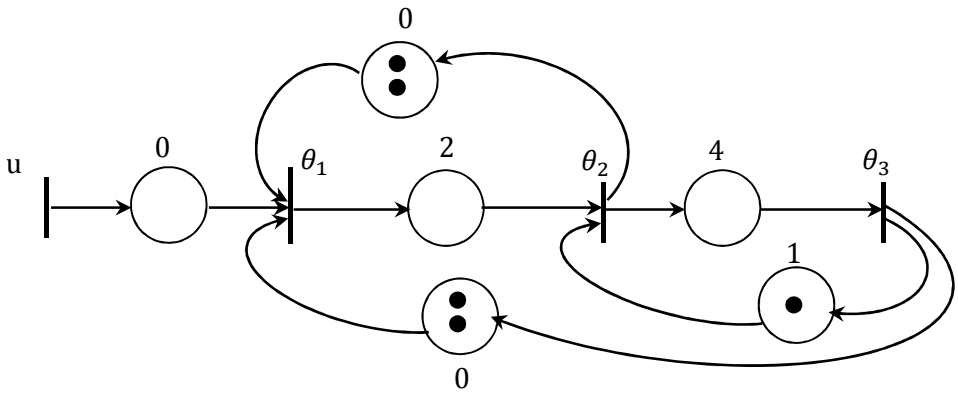


Figure2.3 Exemple de graphe d'évènements

2.3.5.1 Equations compteurs :

On considère un franchissement au plus tôt, c'est-à-dire que toute transition validée est immédiatement franchie. Les fonctions compteurs vérifient alors les équations suivantes :

$$\begin{cases} \theta_1(t) = \min(2 + \theta_2(t), 2 + \theta_3(t), u(t)) \\ \theta_2(t) = \min\big(\theta_1(t-2), 1 + \theta_3(t-1)\big) \\ \theta_3(t) = \theta_2(t-4) \end{cases}$$

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

$$\begin{cases} \theta_1(t) = 2\theta_2(t), \oplus\, 2\theta_3(t) \oplus u(t) \\ \theta_2(t) = \theta_1(t-2) \oplus 1\theta_3(t-1) \\ \theta_3(t) = \theta_2(t-4) \end{cases}$$

Ce qui donne :

$$\theta(t) = \begin{pmatrix} \varepsilon & 2 & 2 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(t) \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(t-1) \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(t-2)$$

$$\oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} \theta(t-4) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} u(t).$$

Cette équation est implicite et peut être remplacée par sa solution explicite. Tout d'abord, on calcule A_0^*, tel que $A_0 = \begin{pmatrix} \varepsilon & 2 & 2 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$.

$$A_0^* = \oplus_{i \in \mathbb{N}} = I \oplus A_0 \oplus A_0^2 = \begin{pmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & 2 & 2 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} = \begin{pmatrix} e & 2 & 2 \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{pmatrix}$$

L'équation explicite (2.2) est donnée par

$$\theta(t) = \begin{pmatrix} \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(t-1) \oplus \begin{pmatrix} 2 & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(t-2) \oplus \begin{pmatrix} \varepsilon & 2 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{pmatrix} \theta(t-4) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} u(t) \, .$$

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 la Figure 2.4.

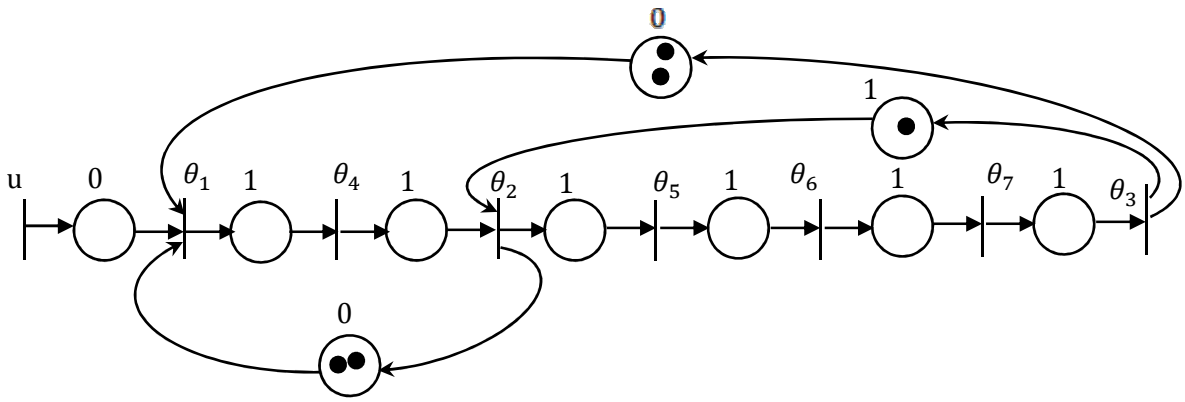


Figure2.4 Graphe d'événements temporisés étendu

Le modèle d'état obtenu est donné par :

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

2.3.5.2 Equations dateurs :

Pour le graphe d'événements de la Figure 2.3, on associe une fonction dateurs $\theta_i(k)$ à chaque transition. Pour un franchissement au plus tôt, les fonctions dateurs vérifient les équations suivantes :

$$\begin{cases} \theta_1(k) = \max(\theta_2(k-2), \theta_3(k-2), u(k)) \\ \theta_2(k) = \max\big(2 + \theta_1(k), 1 + \theta_3(k-1)\big) \\ \theta_3(k) = 4 + \theta_2(k) \end{cases}$$

Dans l'algèbre (max,+) , ces équations sont écrites comme suit :

$$\begin{cases} \theta_1(k) = \theta_2(k-2) \oplus \theta_3(k-2) \oplus u(k)) \\ \theta_2(k) = 2\theta_1(k) \oplus 1\theta_3(k-1) \\ \theta_3(k) = 4\theta_2(k) \end{cases}$$

Ce qui donne :

$$\theta(k) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon \end{pmatrix} \theta(k) \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(t-1) \oplus \begin{pmatrix} \varepsilon & e & e \\ e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix} \theta(k-2) \oplus \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \end{pmatrix} u(k).$$

Cette équation est implicite et peut être remplacée par sa solution explicite. Tout d'abord, on calcule A_0^*, tel que $A_0 = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon \end{pmatrix}$.

$$A_0^* = \oplus_{i \in \mathbb{N}} = I \oplus A_0 \oplus A_0^2 \oplus A_0^3 = \begin{pmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ 6 & \varepsilon & \varepsilon \end{pmatrix} \oplus \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{pmatrix}$$

$$= \begin{pmatrix} e & \varepsilon & \varepsilon \\ 2 & e & \varepsilon \\ 6 & 4 & e \end{pmatrix}.$$

L'équation explicite (2.2) est donnée par

$$\theta(k) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 5 \end{pmatrix} \theta(k-1) \oplus \begin{pmatrix} 2 & e & e \\ \varepsilon & 2 & 2 \\ e & 6 & 6 \end{pmatrix} \theta(k-2) \oplus \begin{pmatrix} e \\ 2 \\ 6 \end{pmatrix} u(k) .$$

Pour avoir la forme d'état, nous avons étendu le graphe d'événements temporisés initial pour avoir un nouveau graphe équivalent avec des marquages égaux à 1 ou 0. Nous obtenons alors le graphe de la Figure 2.5.

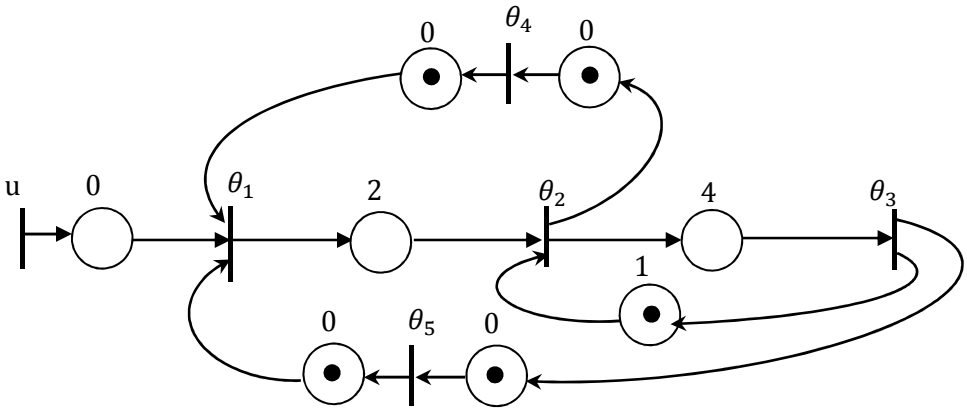


Figure2.5 Graphe d'événements temporisé étendu

Le modèle d'état obtenu est donné par :

$$x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & e & e \\ \varepsilon & \varepsilon & e & 2 & 2 \\ \varepsilon & \varepsilon & 4 & 6 & 6 \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} e \\ 2 \\ 6 \\ \varepsilon \\ \varepsilon \end{bmatrix} u(k),$$

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 t_u.

2.4 Conclusion

Dans ce premier chapitre, nous avons présenté les outils mathématiques utilisés dans ce travail. Nous avons ainsi donné quelques définitions et propriétés relative aux dioïdes. Nous avons enchainé par la théorie spectrale des matrices carrées, définies sur un dioïde. Plus exactement on s'est intéressé à la propriété de périodicité et à la notion de valeur propre et vecteur propre dans ce cas (i.e. dioïde).

Nous avons par la suite introduit les graphes d'événements temporisés après un petit rappel sur les réseaux de Petri. Par la suite, on a montré comment, à partir d'un GET, obtenir des modèles linéaires dans l'algèbre (max,+) et (min,+).

Chapitre 2

Fonctionnement et modélisation de l'architecture

Dans ce deuxième chapitre, nous décrivons la composition détaillée et le fonctionnement des éléments principaux composants les AAR, les contrôleurs (CPU et interface de communication compris), les modules E/S déportés (MES) et le réseau de communication. Après, nous passons à la modélisation des AAR en utilisant les Graphes d'événements temporisés (GET) puis à leur représentation en équations (max,+) et (min,+) linéaires. Les notions énoncées dans cette partie son issus des références [6-8]

3.1 Les architectures d'automatisation en réseau

3.1.1 Constitution d'une AAR

Les architectures d'automatisation en réseau (AAR) se composent généralement de contrôleurs logiques CL ou calculateurs, de modules d'entrées/sorties (MES) connectés à un réseau de communication (aussi appelé réseau de terrain) par lequel ces différents composants échangent des données. Le but de ce système est de permettre les échanges entre les contrôleurs logiques et la partie opérative.

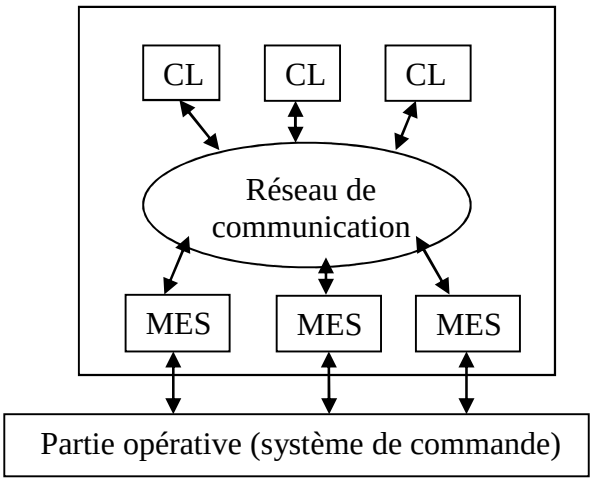


Figure 3.1. Structure générale d'une architecture d'automatisation en réseau

3.1.1.1 Caractéristique du réseau de communication

Les réseaux de terrain sont une réponse directe à l'augmentation de la complexité du câblage dans le niveau inférieur du processus de production. L'idée de base consiste à changer la liaison point à point par un réseau où toutes les informations sont multiplexées temporellement sur le même support (bus). Ce support fournit un lien direct entre les capteurs, les actionneurs et les équipements de contrôle.

Les avantages importants des réseaux de terrain sont :

-Faible coût de câblage avec aussi une influence directe sur la simplification de l'installation et sur la réduction de la complexité de manutention.

-Modularité : L'expansion du système est plus facile et rapide. L'addition d'un nouveau capteur ou groupe de capteurs se fait simplement avec une connexion sur le réseau, et non sur un nouveau "layout", avec de nouvelles interfaces dans le contrôleur central.

-Possibilité de diffusion ("broadcasting" et "multicasting") : Si la même information physique d'un processus est nécessaire en divers endroits du système (contrôles, opérateurs, alarmes,...), avec un réseau, on peut la rendre accessible, d'une façon virtuelle, à tous les autres utilisateurs. Un réseau de terrain avec capacité de "broadcasting" permet l'acquisition simultanée de tous les capteurs mis en relation (mais situés en des endroits différents). La diffusion d'information pour tous les consommateurs sur le réseau peut aussi se faire d'une façon simultanée.

Dans notre étude nous considérons que le réseau de communication est un élément de l'AAR, mais on ne s'intéresse ni à sa composition ni à sa typologie et ni à sa technologie. Nous précisons juste le type de protocole d'échange.

Les principaux protocoles d'échange dans une architecture d'automatisation en réseau sont les suivants :

-*Maître/esclave* : avec ce protocole, un esclave ne peut émettre d'information que lorsqu'il reçoit l'aval du maître. Le maître interroge cycliquement les esclaves et leur donne la main, chacun son tour, pour pouvoir émettre des données.

- *Client/serveur* : avec ce protocole, un client interroge de manière autonome un serveur pour demander une information et le serveur lui répond simplement. Dans notre cas, les contrôleurs sont les clients et les modules d'entrées/sorties sont les serveurs.

- *Producteur/consommateur* : dans ce cas, un nœud producteur n'attend pas la réception d'un aval ou d'une requête pour émettre de l'information, il le fait indépendamment (à chaque fois qu'une variable change d'état par exemple) et l'information est envoyée précisément aux seuls consommateurs intéressés par la variable publiée. Dans notre cas, pour l'AAR, le module d'entrée (producteur) envoie les requêtes vers le CPU (consommateur) sans qu'il reçoive une demande de ce dernier. Dans la phase de retour de l'information vers le module de sortie (consommateur) la CPU devient producteur.

3.1.2 Fonctionnement de l'AAR

3.1.2.1 Fonctionnement de la CPU

Les contrôleurs logiques jouent deux rôles dans l'AAR. Ils exécutent un programme qui permet de déterminer le nouvel état des sorties en fonction de l'état antérieur de ses

variables et de l'état courant des entrées. Ils doivent également gérer les échanges avec les MES pour récupérer les valeurs des entrées et transmettre les valeurs de sorties. Ils sont de ce fait décomposés en deux entités : le processeur de calcul (CPU), qui exécute le programme de commande, et la carte de communication (notée COM), qui communique avec les modules d'E/S.

Le processeur de calcul a un comportement cyclique et la carte de communication un comportement périodique. Ces deux cycles, sont appelés respectivement cycle de calcul et cycle d'IOscanning. Ceci est montré à la Figure 3.2.

- Le processeur de calcul copie toutes les entrées avant d'exécuter le programme de commande (étape 1) ou les prend en compte au cours du calcul lorsqu'il en a besoin (étape6). Il convient de noter que dans le deuxième cas, la phase de lecture n'existe pas pour le processeur de calcul.

Le processeur de calcul émet toutes ses sorties à la fin du programme de commande (étape 2) ou les émet au cours du calcul (étape5). Il convient de noter que dans le deuxième cas, la phase d'écriture n'existe pas pour le processeur de calcul.

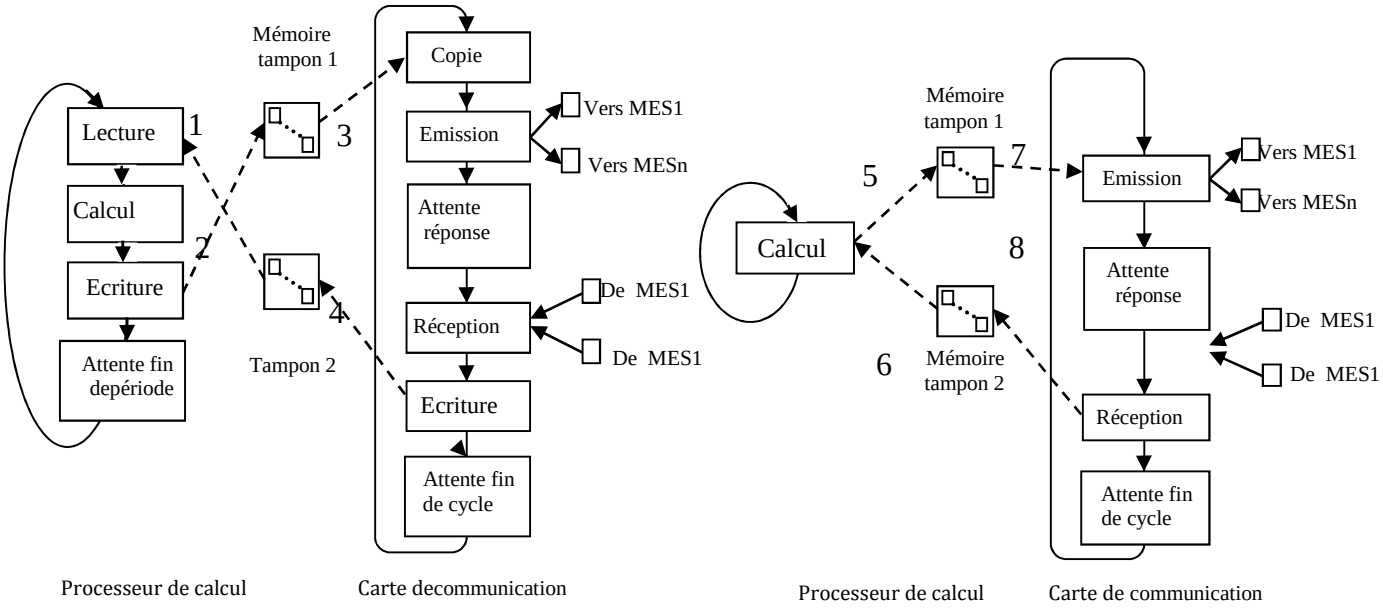


Figure 3.2 Echange de données entre la CPU et la carte de communication

- La carte de communication prend en compte les nouvelles sorties à envoyer aux MES avant la phase d'émission (cas 3) ou au cours de cette phase (cas 7). Il convient de noter que dans le deuxième cas, la phase de copie n'existe pas pour la carte de communication.

La carte de communication met à disposition du processeur de calcul les nouvelles valeurs des entrées à la fin de la phase de réception (une fois qu'elles sont toutes arrivées) (cas 4) ou au fur et à mesure de leur arrivée (cas 8). Il convient de noter que dans le deuxième cas, la phase d'écriture n'existe plus pour la carte de communication.

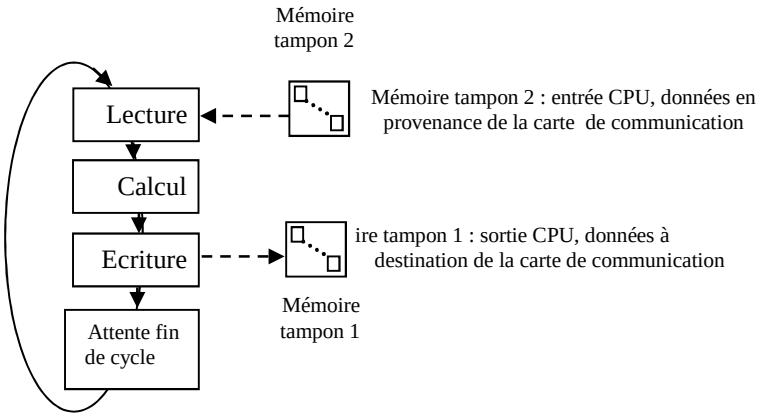


Figure 3.3 Fonctionnement d'un processeur de calcul (CPU)

Le processeur de calcul Figure 3.3 possède un fonctionnement cyclique qui se décompose en trois étapes : lecture des valeurs des entrées dans la mémoire tampon 2 située entre la carte de communication et le processeur de calcul, calcul des nouvelles valeurs des sorties, écriture des nouvelles valeurs des sorties dans la mémoire tampon 1 située entre le processeur de calcul et la carte de communication.

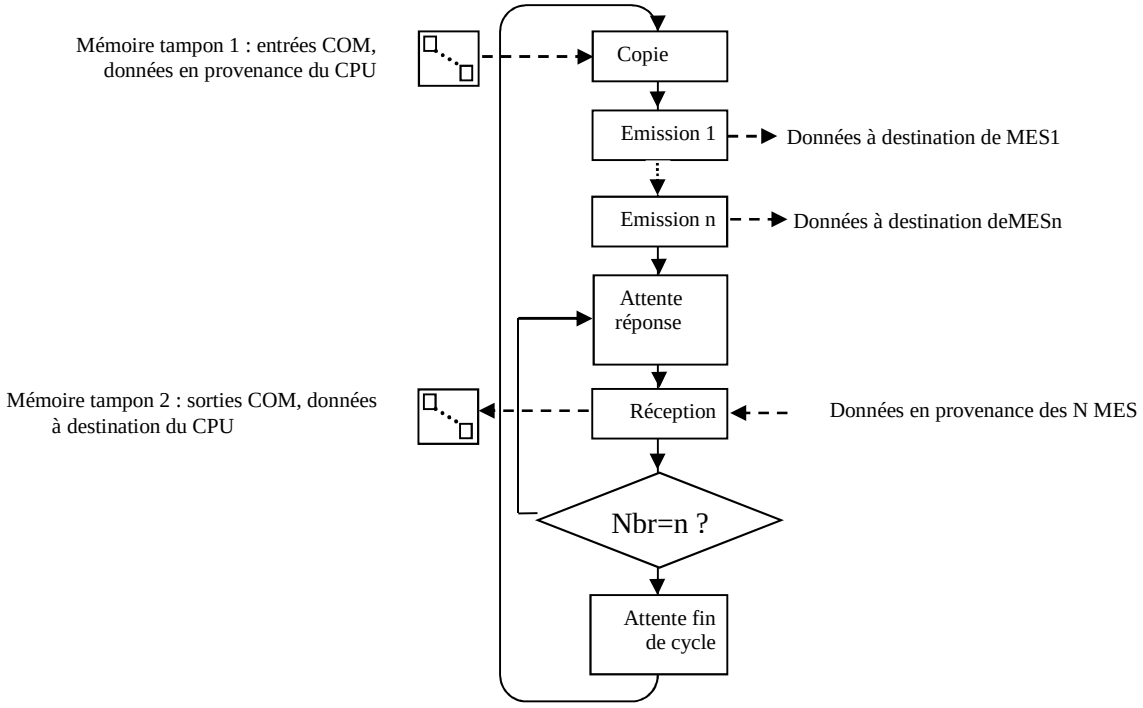


Figure 3.4. Fonctionnement d'une carte de communication

La carte de communication de la Figure3.4 a un comportement périodique. Elle commence par copier toutes les sorties calculées par le processeur de la mémoire tampon entre le processeur de calcul et la carte de communication dans sa mémoire propre puis elle émet les requêtes (une à une) aux n MES auxquels elle est connectée dans un ordre fixé par l'utilisateur. Après ces émissions, elle attend les réponses des MES, les traitent une à une dans leur ordre d'arrivée et écrit les nouvelles valeurs des entrées au fur et à mesure dans la mémoire tampon 2 située entre la carte de communication et le processeur de calcul. Une file d'attente FIFO gère la réception des réponses afin de les traiter dans leur ordre d'arrivée et n'en perd aucune. Une fois toutes les réponses reçues, il y a une période d'attente pour garder un temps de cycle constant. Il est important que la durée du cycle d'IOscanning soit bien configurée pour que toutes les réponses puissent bien être reçues avant la fin du cycle de la carte de communication.

3.1.2.2 Fonctionnement des MES

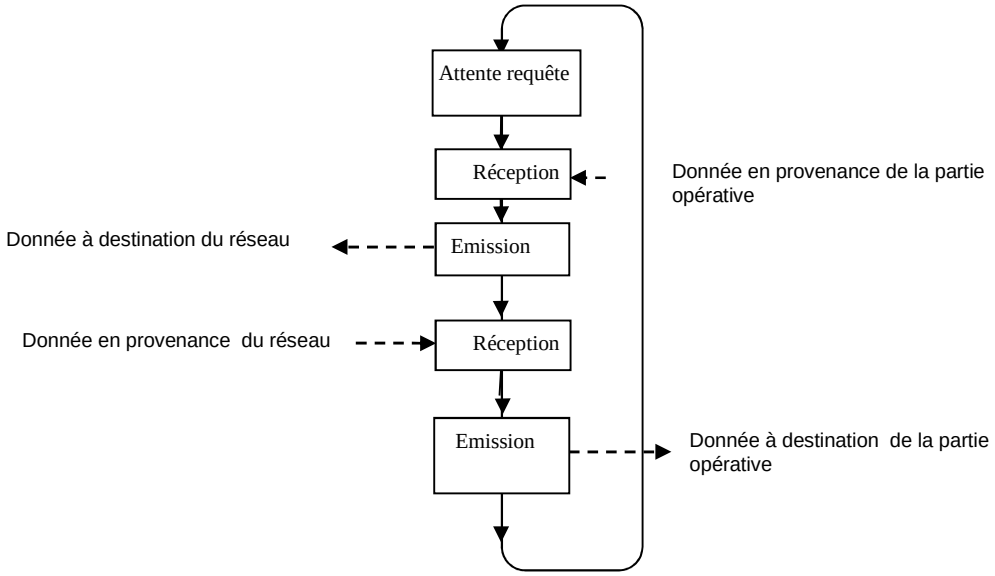


Figure 3.5. Fonctionnement des modules d'entrées/sorties

L'état normal d'un MES est un état d'attente, dont il ne sort que lors de la réception d'une requête en provenance d'une carte de communication.

Le traitement de cette requête consiste :

- à lire les valeurs des entrées physiques connectées au module d'entrée,
- et d'envoyer la réponse contenant les nouvelles valeurs des entrées à la carte de communication à l'origine de la requête.

Après une réponse contenant les nouvelles valeurs de sortie de la partie opérative sera envoyé de la carte de communication vers le module de sortie à travers le réseau de communication.

Et finalement le fonctionnement de tous les éléments de l'AAR est montré à la Figure 3.6.

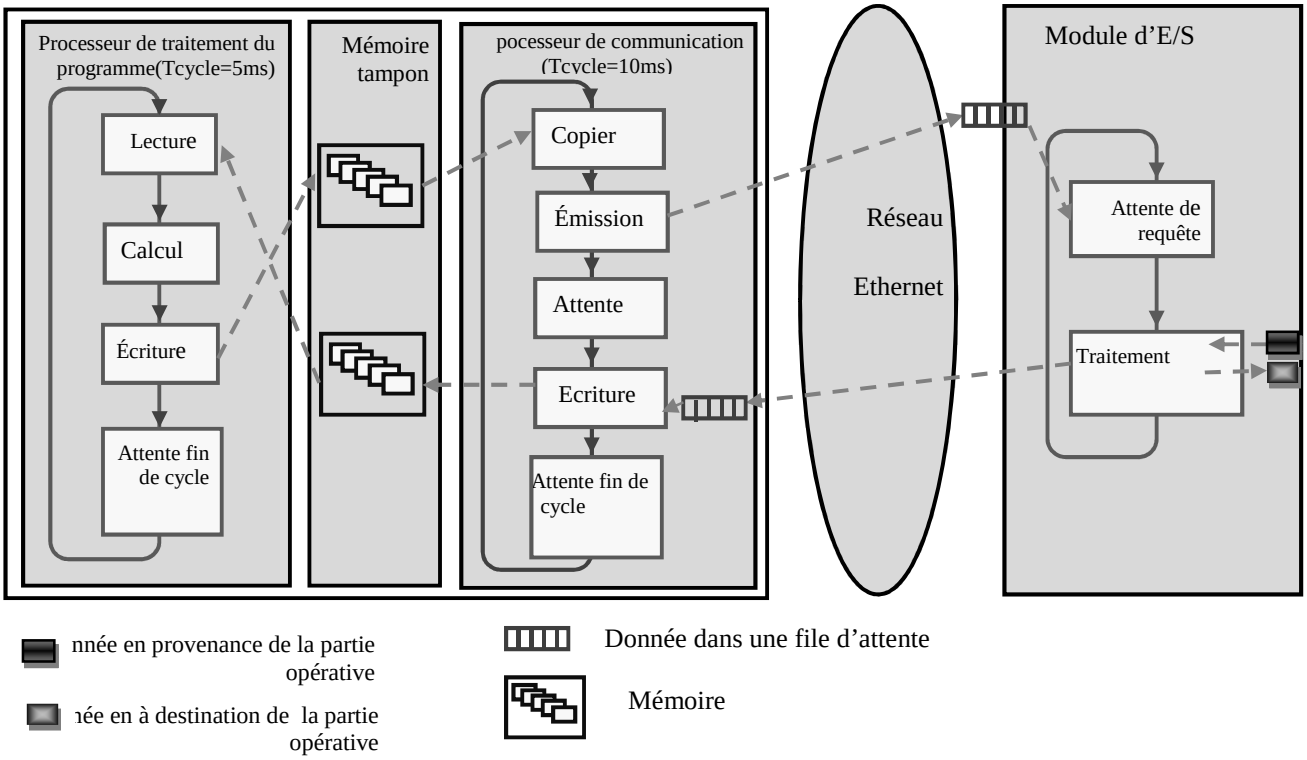


Figure 3.6 : Comportement de l'AAR

Hypothèses sur le fonctionnement des composants des SCR

Les hypothèses suivantes sont considérées dans ce mémoire :

H1 : les temps de lecture/écriture des données dans les mémoires tampons 1 et 2 sont négligés.

H2 : Nous supposons que la CPU est dédiée, c'est-à-dire qu'elle traite juste les informations qui proviennent du module d'E/S.

H3 : Nous supposons que l'AAR fonctionne à sa vitesse maximale, c'est-à-dire chaque requête est traitée dès que le traitement de cette dernière est possible.

3.2 Modélisation en graphes d'événements temporisés (GET) et représentation dans l'algèbre des dioïdes des AAR

Dans cette section de l'étude, nous utilisons les graphes d'événements temporisés pour la modélisation de l'architecture d'automatisation en réseau.

3.2.1. Cas 1 : AAR mono client – mono serveur

Nous commençons la modélisation par le cas le plus simple avec une AAR incluant un seul contrôleur et un seul module E/S.

3.2.1.1Modélisations de la partie Contrôleur : compte tenu du fonctionnement du contrôleur décrit dans la section 3.1, nous obtenons le modèle assez intuitif et naturel de la Figure 3.7.

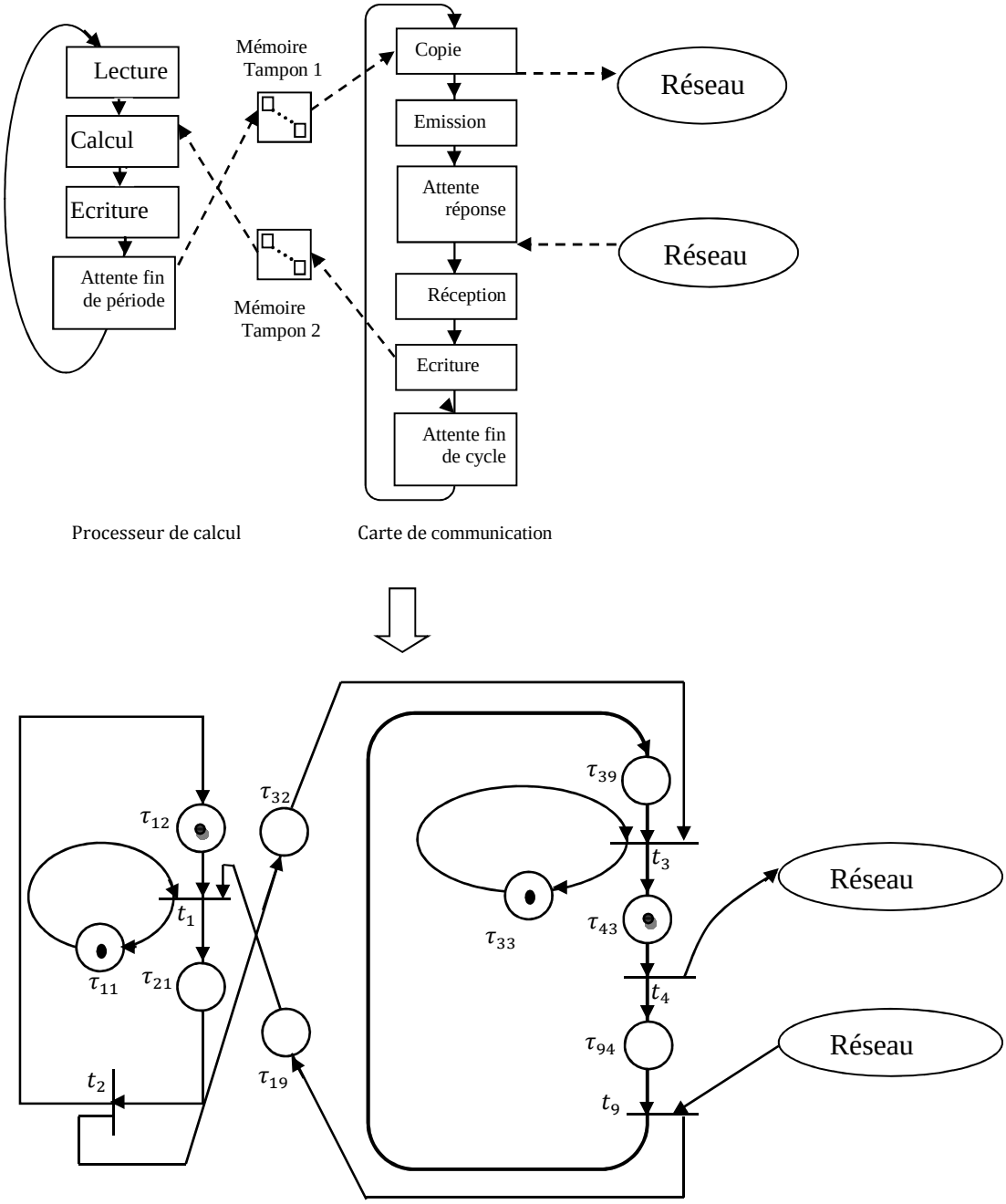


Figure 3.7 Modèle GET de la CPU et la carte de communication

Le franchissement de t_1 modélise le début de cycle CPU. P_{21} représente les phases de lecture, calcul et écriture, qui durent τ_{21} unités de temps et se terminent par le franchissement de la transition t_2. En parallèle, un jeton entre dans la place p_{11}, du fait du franchissement de la transition t_1, et y reste durant une durée égale à τ_{11} qui représente la période de la CPU. Une fois cette durée écoulée. La CPU est alors prête à entamer un nouveau cycle de calcul.

- La carte de communication possède un fonctionnement similaire. En effet le franchissement de la transition $t_3$modélise le début d'un cycle de scrutation et celui de t_4, la fin de l'émission de la réponse vers le module de sortie. Un jeton dans p_{94} représente l'entrée en phase d'attente de réception de la requête. Une fois que la requête est reçue, la transition t_9 est immédiatement franchie, la requête est transférée vers la mémoire tampon P_{19}, et la COM entre dans une phase d'attente jusqu'à la fin d'écoulement de la période de scrutation τ_{33}. Une fois l'attente terminée la COM commence un nouveau cycle de scrutation.

3.2.1.2 Module d'E/S : Le modèle du module E/S est représenté par le GET de la Figure 3.8. 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_{76}. 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 avec le franchissement de la transition t7. Un jeton dans p_{87} indique que le traitement de la requête a commencé. Ce processus dure τ_{87} unités de temps avant de se terminer par le tir 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. Le franchissement de la transition t_5 est franchie après la réception de la réponse du CPU à la requête envoyé par le module d'entrée et l'écoulement de temps de réponse τ_{58}. Un jeton dans P_{65} indique que la réponse de la CPU est arrivée au module de sortie à travers le réseau de communication.

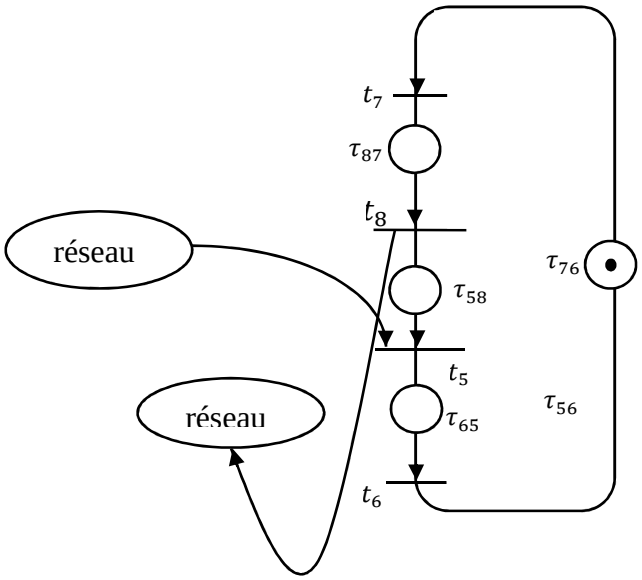


Figure 3.8 Modèle en GET du Module d'E/S

3.2.1.3 Réseau de communication : le réseau de communication est représenté par deux délais :un délai que subit la requête lors de son envoi du module d'entrée vers le calculateur et un autre que subit la réponse lors de son envoi du calculateur vers le module de sortie. Les deux éléments de GET suivants modélisent ces deux délais :

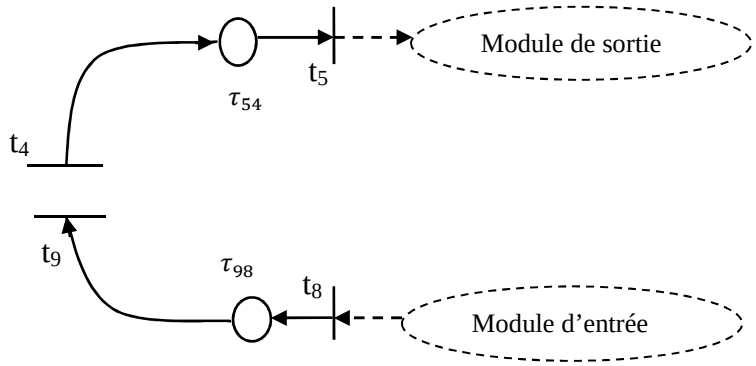


Figure 3.9 Modèle GET du réseau de communication

Comme dans les modèles GET de la COM et de la CPU, le franchissement de t_4 signifie la fin d'émission de la requête depuis le contrôleur. La requête entre dans le réseau de communication et y subit un délai de bout-en-bout total de τ_{54} unités de temps. Le franchissement de t_5 indique que la requête a traversé le réseau de communication. Dans le sens du retour, la réponse entre dans le réseau par le franchissement de la transition t_8 qui

modélise la sortie du module d'entrée et y subit un délai total de τ_{98} unités de temps, la place entre la transition t_7 et t_u modélise la commande du système.

Au final, nous pouvons lier les modèles des différents composants pour constituer un modèle global de l'AAR, représenté à la Figure 3.10 :

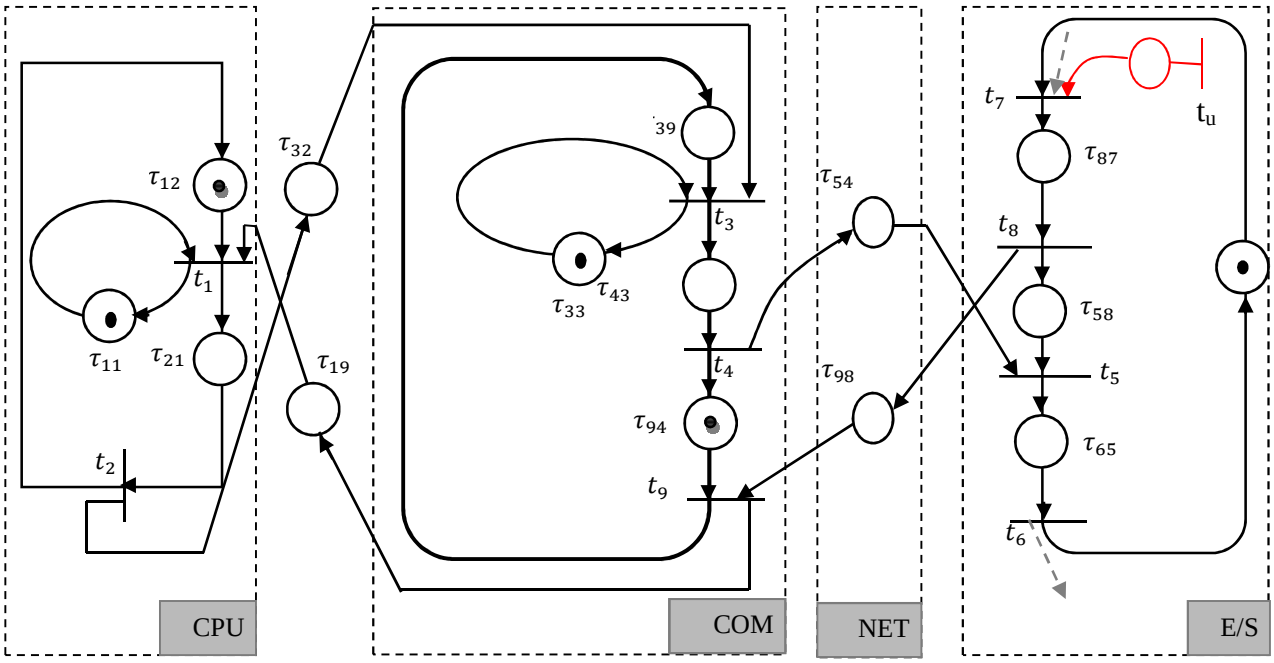


Figure 3.10 Modèle GET d'une AAR mono client mono serveur

3. 3.1.4 Le temps de réponse de l'AAR

Le temps de réponse (Dr) ou bien le temps de réactivité de l'AAR [12,13], est le délai entre l'occurrence de l'événement sur l'entrée du module d'entrée/sortie et sa conséquence issue du PLC sur le système commandé. Il est décomposable en un ensemble de délais élémentaires, subis aux différents niveaux de l'AAR comme le montre la Figure 3.11

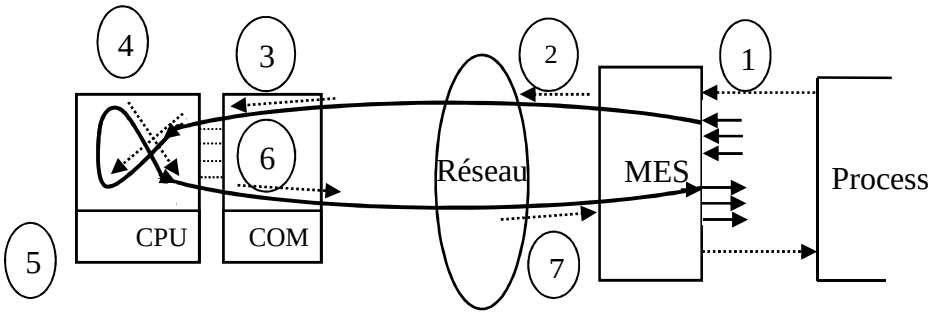


Figure 3.11 Le temps de réponse de l'AAR

D'après la Figure 3.11 nous remarquons que le temps de réponse est décomposé en sept étapes, suivant le niveau dans lequel se trouve l'événement généré par le capteur.

1- Génération d'un événement par le capteur puis attente d'arrivée et le traitement de cette requête par le module d'entrée, cette étape va prendre τ_{87} unité de temps.

2- Après l'arrivée d'une requête et son traitement par le MES, une réponse est envoyée à travers le réseau de communication durant τ_{98} unité de temps.

2- L'arrivée de la réponse au buffer d'entrée de la COM, et sa mise en mémoire tampon ce qui prend un temps entre 0 et τ_{87}.

4- La requête est dans la mémoire tampon, la CPU va traiter la requête immédiatement après son arrivée, si cette dernière est arrivée après la fin de cycle de CPU, si non elle va attendre un nouveau cycle de CPU. Donc le temps nécessaire pour cette étape est de 0 au minimum et $\tau_{11} = T_{cpu}$ au maximum.

5- Exécution du programme utilisateur pour générer l'événement de réaction (conséquence). Ce qui va prendre τ_{21} unité de temps.

6- Mise en mémoire tampon de la conséquence et attente du début d'un nouveau cycle de la COM, ça va prendre au minimum 0 unité de temps et au maximum $\tau_{33} = T_{com}$.

7- La réponse va prendre τ_{43} unité de temps dans la COM et traverse le réseau de communication durant τ_{54} unités de temps.

7- L'arrivée de la conséquence sur le module de sortie et son traitement va prendre τ_{65} unité de temps.

Finalement on a un temps de réponse entre deux bornes, une borne minimum (Dr_{min}) et une borne maximum (Dr_{max}):

$$Dr_{min} = \tau_{87} + \tau_{98} + \tau_{94} + 0 + \tau_{21} + 0 + \tau_{43} + \tau_{54} + \tau_{65}.$$

$$Dr_{max} = \tau_{87} + \tau_{98} + \tau_{94} + \tau_{11} + \tau_{21} + \tau_{33} + \tau_{43} + \tau_{54} + \tau_{65}.$$

3.2.1.5 Représentation en équations Min-Plus linéaires du graphe d'événements de l'AAR

Nous avons vu dans le chapitre précédent qu'on peut décrire le comportement dynamique d'un graphe d'événements temporisés par des équations Min-Plus linéaires. Pour représenter le comportement dynamique du graphe de la Figure 3.10, nous associons à chaque transition t_i, avec i = 1 à 9, une fonction compteur $\theta_i(t)$, qui représente le nombre cumulé de franchissement de la transition ti à l'instant t et ce depuis l'instant initial. La fonction compteur attribuée à la transition d'entrée (la commande) est notée u(t). On considère un franchissement au plus tôt, c'est-à-dire que toute transition validée est immédiatement franchie.

Nous considérons que le GET de la Figure 3.10 est temporisé comme suit :

$\tau_{12} = \tau_{32} = 0, \tau_{94} = \tau_{43} = \tau_{87} = \tau_{56} = \tau_{54} = \tau_{98} = 1, \tau_{11} = 5, \tau_{33} = 10, \tau_{58} = \tau_{58}, \tau_{21} = 2, Dr \in [7 \quad 22]$.

Nous calculons le temps de cycle de graphe d'événements de l'AAR, en utilisant l'expression donnée dans le premier chapitre. Nous décomposons le graphe d'événement temporisé de la Figure 3.10 en circuits élémentaires. On obtient 7 circuits, qu'on note γ_i pour i=1 à 7. Ces circuits sont donnés comme suit :

$\gamma_1 = (t_7, P_{87}, t_8, P_{58}, t_5, P_{56}, t_6, P_{76}, t_7)$, $\gamma_2 = (t_3, P_{43}, t_4, P_{94}, t_9, P_{39}, t_3)$, $\gamma_3 = (t_1, P_{21}, t_2, P_{12}, t_1)$,

$\gamma_4 = (t_3, P_{33}, t_3)$, $\gamma_5 = (t_1, P_{11}, t_1)$, $\gamma_6 = (t_7, P_{87}, t_8, P_{98}, t_9, P_{39}, t_3, P_{43}, t_4, P_{54}, t_5, P_{65}, t_6, P_{76}, t_7)$,

$\gamma_7 = (t_7, P_{87}, t_8, P_{98}, t_9, P_{19}, t_1, P_{21}, t_2, P_{32}, t_3, P_{43}, t_4, P_{54}, t_5, P_{65}, t_6, P_{76}, t_7)$.

En appliquant la formule du temps du cycle, le temps de cycle λ du graphe de l'AAR est donné par :

$$\lambda = \max_{\gamma \in \delta} \frac{T(\gamma)}{M(\gamma)} = \max\left(\frac{9}{1}, \frac{1}{1}, \frac{2}{1}, \frac{5}{2}, \frac{6}{2}, \frac{10}{1}, \frac{5}{1}\right) = 10.$$

Est comme λ=10, et pour ne pas avoir un temps de réponse inferieur au temps de cycle on prend $Dr_{min} = 11$.

L'équation implicite (2.7) du graphe d'événement temporisé de la Figure 3.10, s'écrit comme suit :

$$x(t) = A_0.\theta(t) \oplus A_1\theta(t-1) \oplus A_2.\theta(t-2) \oplus A_3.\theta(t-5) \oplus A_4.\theta(t-10) \oplus A_5.\theta(t-11) \oplus B.u(t) \qquad (3.1)$$

Tel que :

$$A_0 = \begin{bmatrix} \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, A_1 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \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 & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \end{bmatrix},$$

$$A_2 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \qquad A_3 = \begin{bmatrix} 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix},$$

$$A_4 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \qquad A_5 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, \mathrm{B} = \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ e \\ \varepsilon \\ \varepsilon \end{bmatrix}.$$

La solution explicite (2.8) dans ce cas, est donnée par l'équation suivante :

$$x(t) = {A_0}^*A_1\theta(t-1) \oplus {A_0}^*A_2.\theta(t-2) \oplus {A_0}^*A_3.\theta(t-5) \oplus {A_0}^*A_4.\theta(t-10) \oplus {A_0}^*A_5.\theta(t-11) \oplus {A_0}^*B.u(t) \qquad (3.2)$$

$$A_0{}^{*} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

Equation d’état

Nous pouvons représenter le comportement de l’AAR par une équation d’état dans l’algèbre Min-Plus. Pour cela, nous allons transformer le graphe de la Figure 3.10 afin d’avoir un autre GET équivalent avec des temporisations binaires (1 ou 0) Figure 3.11. Toutes les places ayant des temporisations supérieures à 1 seront décomposées en plusieurs places temporisées à 1 qui sont reliées par des transitions intermédiaires. Nous définissons de nouvelles variables d’état qu’on va concaténer aux variables initiales.

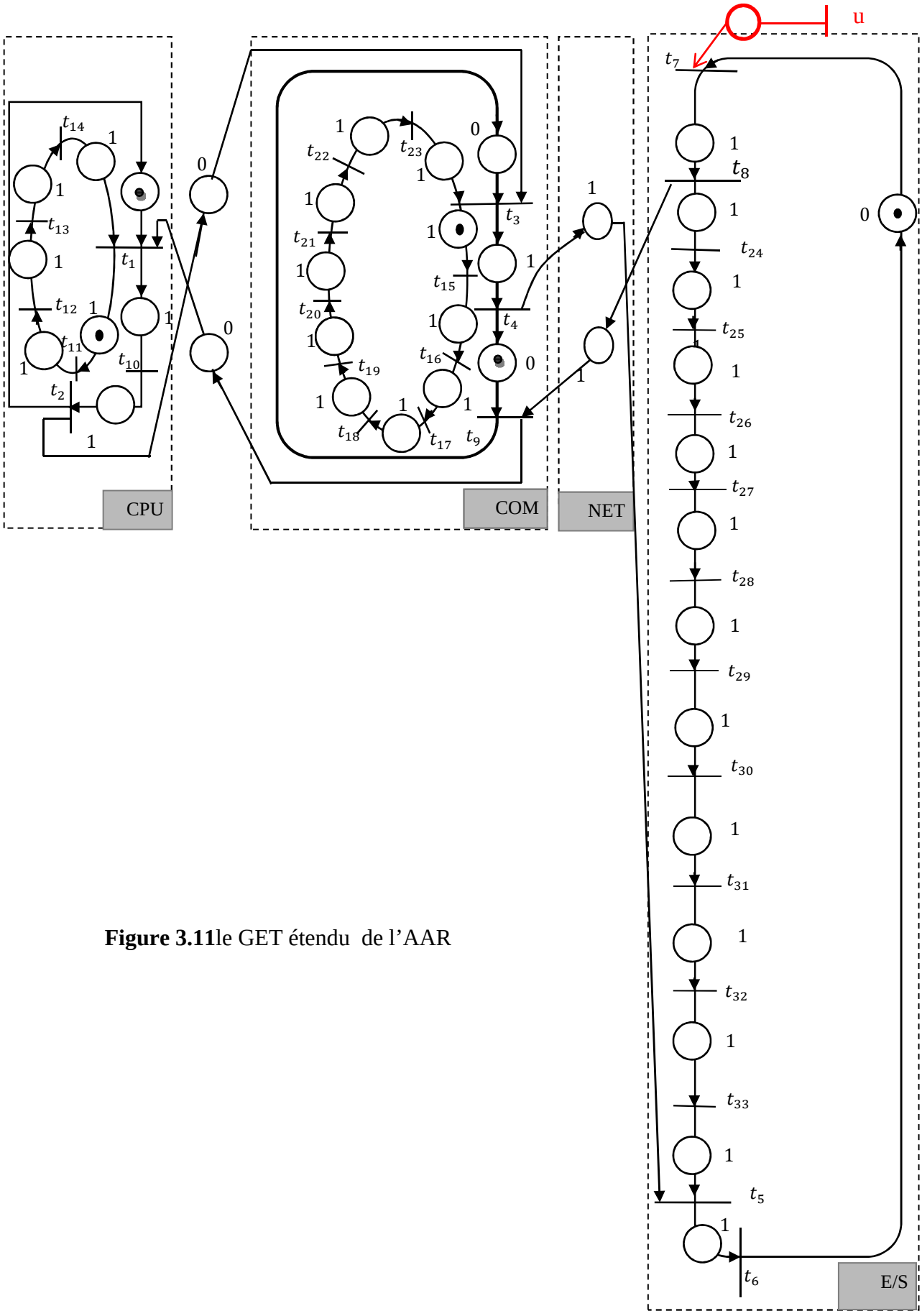


Figure 3.11le GET étendu de l'AAR

Le comportement dynamique du graphe d'événement étendu de l'AAR est décrit par les équations suivantes :

$$
\begin{cases}
x_1(t) = 1.x_2(t) \oplus e.x_{14}(t) \oplus e.x_9(t) \\
x_2(t) = e.x_{10}(t-1) \\
x_3(t) = e.x_2(t) \oplus e.x_{23}(t-1) \oplus e.x_9(t) \\
x_4(t) = e.x_3(t-1) \\
x_5(t) = e.x_{33}(t-1) \oplus e.x_4(t-1) \\
x_6(t) = e.x_5(t-1) \\
x_7(t) = 1x_6(t-1) \oplus e.u(t) \\
x_8(t) = e.x_7(t-1) \\
x_9(t) = e.x_8(t-1) \oplus 1.x_4(t) \\
x_{10}(t) = e.x_1(t-1) \\
x_{11}(t) = 1x_1(t-1) \\
x_l(t) = e.x_{(l-1)}(t-1) \text{ , pour } l = 12 \text{ à } 14 \\
x_{15}(t) = 1.x_3(t-1) \\
x_l(t) = x_{(l-1)}(t-1), \text{ pour } l = 16 \text{ à } 23 \\
x_{24}(t) = x_8(t-1) \\
x_l(t) = e.x_{(l-1)}(t-1), \text{ pour } l = 25 \text{ à } 33
\end{cases}
\qquad (3.3)
$$

Ces équations s'écrivent sous la forme matricielle,

$x(t) = A_0x(t) \oplus A_1x(t-1) \oplus B_1u(t)$,

tel que $A_0, A_1 \in \mathbb{R}_{min}^{33\times33}$,et $B_1 \in \mathbb{R}_{min}^{1\times1}$.

L'équation d'état de graphe étendu est donnée par :

x(t)=A.x(t)$\oplus$ B.u(t), (3.4)

avec $A = {A_0}^* A_1, et\ B = {A_0}^* B_1$,

$$A=\begin{pmatrix}
3 & \varepsilon & 4 & 3 & 4 & \varepsilon & 3 & 3 & \varepsilon & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 \\
3 & \varepsilon & 3 & 3 & 4 & \varepsilon & 2 & 3 & \varepsilon & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 \\
3 & \varepsilon & 3 & 3 & 4 & \varepsilon & 2 & 2 & \varepsilon & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 2 & 3 & 2 & 3 & 3 & 3 & 3 & 2 & 2 & 3 & 2 & 3 & 3 & 3 & 3 & 3 \\
3 & \varepsilon & 3 & 3 & 3 & \varepsilon & 3 & 2 & \varepsilon & 3 & 2 & 2 & 2 & 3 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 3 & 3 & 3 \\
2 & \varepsilon & 3 & 3 & 3 & \varepsilon & 2 & 1 & \varepsilon & 3 & 2 & 2 & 2 & 3 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 3 & 3 & 3 \\
3 & \varepsilon & 3 & 3 & 3 & \varepsilon & 2 & 2 & \varepsilon & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 2 & 1 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 3 & 3 \\
4 & \varepsilon & 4 & 4 & 4 & \varepsilon & 3 & 3 & \varepsilon & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 \\
3 & \varepsilon & 3 & 4 & 4 & \varepsilon & 3 & 3 & \varepsilon & 4 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 \\
3 & \varepsilon & 4 & 3 & 4 & \varepsilon & 3 & 3 & \varepsilon & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 \\
3 & \varepsilon & 3 & 4 & 3 & \varepsilon & 3 & 3 & \varepsilon & 3 & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 \\
4 & \varepsilon & 4 & 5 & 4 & \varepsilon & 4 & 4 & \varepsilon & 4 & 4 & 3 & 4 & 4 & 4 & 3 & 4 & 4 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 3 & 4 & 4 & 4 & 4 & 4 & 5 \\
4 & \varepsilon & 4 & 4 & 5 & \varepsilon & 3 & 4 & \varepsilon & 4 & 4 & 4 & 3 & 4 & 4 & 4 & 3 & 4 & 4 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 3 & 4 & 4 & 4 & 4 & 4 \\
4 & \varepsilon & 4 & 4 & 4 & \varepsilon & 4 & 3 & \varepsilon & 4 & 3 & 4 & 4 & 3 & 3 & 4 & 4 & 3 & 4 & 4 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 3 & 4 & 4 & 4 & 4 \\
3 & \varepsilon & 4 & 4 & 4 & \varepsilon & 3 & 4 & \varepsilon & 4 & 4 & 3 & 4 & 4 & 3 & 3 & 4 & 4 & 3 & 4 & 43 & 4 & 3 & 4 & 4 & 4 & 4 & 4 & 3 & 4 & 4 & 4 \\
4 & \varepsilon & 4 & 4 & 4 & \varepsilon & 4 & 3 & \varepsilon & 4 & 3 & 3 & 4 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 & 4 & 4 \\
3 & \varepsilon & 4 & 4 & 4 & \varepsilon & 3 & 4 & \varepsilon & 4 & 4 & 3 & 3 & 4 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 & 4 \\
4 & 4 & 4 & 4 & 4 & \varepsilon & 3 & 3 & \varepsilon & 3 & 3 & 4 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 4 \\
3 & \varepsilon & 3 & 4 & 4 & \varepsilon & 3 & 3 & \varepsilon & 4 & 33 & 4 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 \\
3 & \varepsilon & 4 & 3 & 4 & \varepsilon & 3 & 3\varepsilon & 3 & 33 & 3 & 4 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 & 3 \\
4 & \varepsilon & 3 & 4 & 3 & \varepsilon & 3 & 3 & \varepsilon & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 & 4 \\
3 & \varepsilon & 3 & 3 & 4 & \varepsilon & 2 & 3 & \varepsilon & 4 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 & 3 \\
3 & \varepsilon & 4 & 3 & 3 & \varepsilon & 3 & 2 & \varepsilon & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 & 3 \\
3 & \varepsilon & 3 & 4 & 3 & \varepsilon & 2 & 3 & \varepsilon & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 4 & 3 & 4 & 3 & 3 & 4 \\
3 & \varepsilon & 4 & 3 & 4 & \varepsilon & 3 & 3 & \varepsilon & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 & 3 & 3 & 4 & 3 \\
3 & \varepsilon & 3 & 4 & 3 & \varepsilon & 3 & 3 & \varepsilon & 3 & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 & 3 & 3 & 4 \\
3 & \varepsilon & 3 & 3 & 4 & \varepsilon & 2 & 3 & \varepsilon & 3 & 3 & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 & 3 & 3 \\
3 & \varepsilon & 3 & 3 & 3 & \varepsilon & 3 & 2 & \varepsilon & 3 & 2 & 3 & 3 & 2 & 2 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 & 3 \\
2 & \varepsilon & 3 & 4 & 3 & \varepsilon & 2 & 3 & \varepsilon & 3 & 3 & 2 & 3 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 3 & 3 & 4 & 3 & 2 & 3 & 3 & 4 \\
3 & \varepsilon & 3 & 3 & 4 & \varepsilon & 2 & 2 & \varepsilon & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 3 & 34 & 3 & 2 & 3 & 3 \\
3 & \varepsilon & 2 & 3 & 3 & \varepsilon & 3 & 2 & \varepsilon & 3 & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 33 & 4 & 3 & 2 & 3 \\
2 & \varepsilon & 3 & 2 & 3 & \varepsilon & 2 & 3 & \varepsilon & 3 & 3 & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 & 2 & 3 & \varepsilon & 2 & 3 & 2 & 3 & 3 & 3 & 4 & 3 & 2 \\
3 & \varepsilon & 3 & 3 & 2 & \varepsilon & 2 & 2 & \varepsilon & 2 & 2 & 3 & 2 & 3 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 & 2 & \varepsilon & \varepsilon & 2 & 3 & 2 & 3 & 3 & 3 & 4 & 3 \\
3 & \varepsilon & 2 & 4 & 3 & \varepsilon & 1 & 2 & \varepsilon & 3 & 3 & 2 & 3 & 2 & 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 & 3 & \varepsilon & \varepsilon & \varepsilon & 2 & 3 & 2 & 3 & 3 & 3 & 4
\end{pmatrix} , \quad B=\begin{pmatrix}
\varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ e \\ \varepsilon
\end{pmatrix} ,$$

$$A^{*}=\begin{bmatrix}
\varepsilon & 1 & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & 0 & 0 & 1 & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0
\end{bmatrix}$$

3.2.1.6 Représentation en équations Max-plus linéaire du graphe d'événements de l'AAR

On peut aussi décrire le comportement dynamique d'un graphe d'événements temporisé par des équations Max-Plus linéaires. Pour représenter le comportement dynamique du graphe de la Figure 3.10 nous associons a chaque transition t_i, avec i = 1 à 9, une fonction dateur $\theta_i(k)$, qui représente la date du k^{éme} franchissement de la transition t_i. La fonction dateur attribuée à la transition d'entrée est notée u(k). On considère un franchissement au plus tôt, c'est-à-dire que toute transition validée est immédiatement franchie. Les fonctions dateurs vérifient alors les équations suivantes :

$$\begin{cases} \theta_1(k) = e.\theta_2(k-1) \oplus 5.\theta_1(k-1) \oplus e.\theta_9(k) \\ \theta_2(k) = 2.\theta_1(k) \\ \theta_3(k) = e.\theta_2(k) \oplus 10.\theta_3(k-1) \oplus e.\theta_9(k) \\ \theta_4(k) = 1.\theta_3(k) \\ \theta_5(k) = 11.\theta_8(k) \oplus 1.\theta_4(k) \\ \theta_6(k) = 1.\theta_5(k) \\ \theta_7(k) = e.\theta_6(k-1) \oplus e.u(k) \\ \theta_8(k) = 1.x\theta_7(k) \\ \theta_9(k) = 1.\theta_8(k) \oplus e.\theta_4(k-1) \end{cases} \quad (3.5)$$

Ces 9 équations peuvent s'écrire sous forme matricielle :

$$\theta(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 11 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix} \theta(k) \oplus \begin{bmatrix} 5 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(k-1)$$

$$\oplus \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ e \\ \varepsilon \\ \varepsilon \end{bmatrix} u(k)$$

Equation d'état

Non pouvons toujours décrire le comportement d'un graphe d'événements temporisé par une équation d'état, définie dans l'algèbre Max-plus. Nous remarquons que chaque place du graphe de la Figure 3.10 comporte au plus un jeton alors il suffit de calculer A* pour avoir une équation d'état.

$$x(k) = Ax(k-1) \oplus Bu(k), \quad (3.6)$$

$$avec\ A = A_0^*.A_1, B = A_0^*.B_1.$$

$$A_0^* = \begin{bmatrix} 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & 1 & 0 \\ 2 & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 \\ 2 & 0 & 0 & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 \\ 3 & 1 & 1 & 0 & \varepsilon & \varepsilon & 5 & 4 & 3 \\ 4 & 2 & 2 & 1 & 0 & \varepsilon & 12 & 11 & 4 \\ 5 & 3 & 3 & 2 & 1 & 0 & 13 & 12 & 5 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & 1 & 0 \end{bmatrix}.$$

L'équation d'état est donnée par:

$$x(k) = \begin{bmatrix} 5 & 0 & \varepsilon & 0 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ 7 & 2 & \varepsilon & 2 & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon \\ 7 & 2 & 10 & 2 & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon \\ 8 & 3 & 11 & 3 & \varepsilon & 6 & \varepsilon & \varepsilon & \varepsilon \\ 9 & 4 & 12 & 4 & \varepsilon & 13 & \varepsilon & \varepsilon & \varepsilon \\ 10 & 5 & 13 & 5 & \varepsilon & 14 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} 2 \\ 4 \\ 4 \\ 5 \\ 12 \\ 13 \\ 0 \\ 1 \\ 2 \end{bmatrix} u(k) \qquad (3.7)$$

3.2.2 Cas 2: mono client- multi serveurs :

Dans ce cas, *N* modules d'entrées/sorties sont interrogés par un seul contrôleur. Durant un cycle de scrutation, les modules d'entrées envoient une rafale de *N* requêtes vers le contrôleur qui traite ces requêtes puis, envoie les réponses vers les modules de sorties à travers le réseau de communication. Le module CPU est modélisé exactement comme dans le cas mono serveur. En effet, au début d'un cycle CPU, toutes les entrées sont lues en bloc (comme une seule variable et donc comme dans le cas mono serveur) et les sorties sont mises à jour en bloc également.

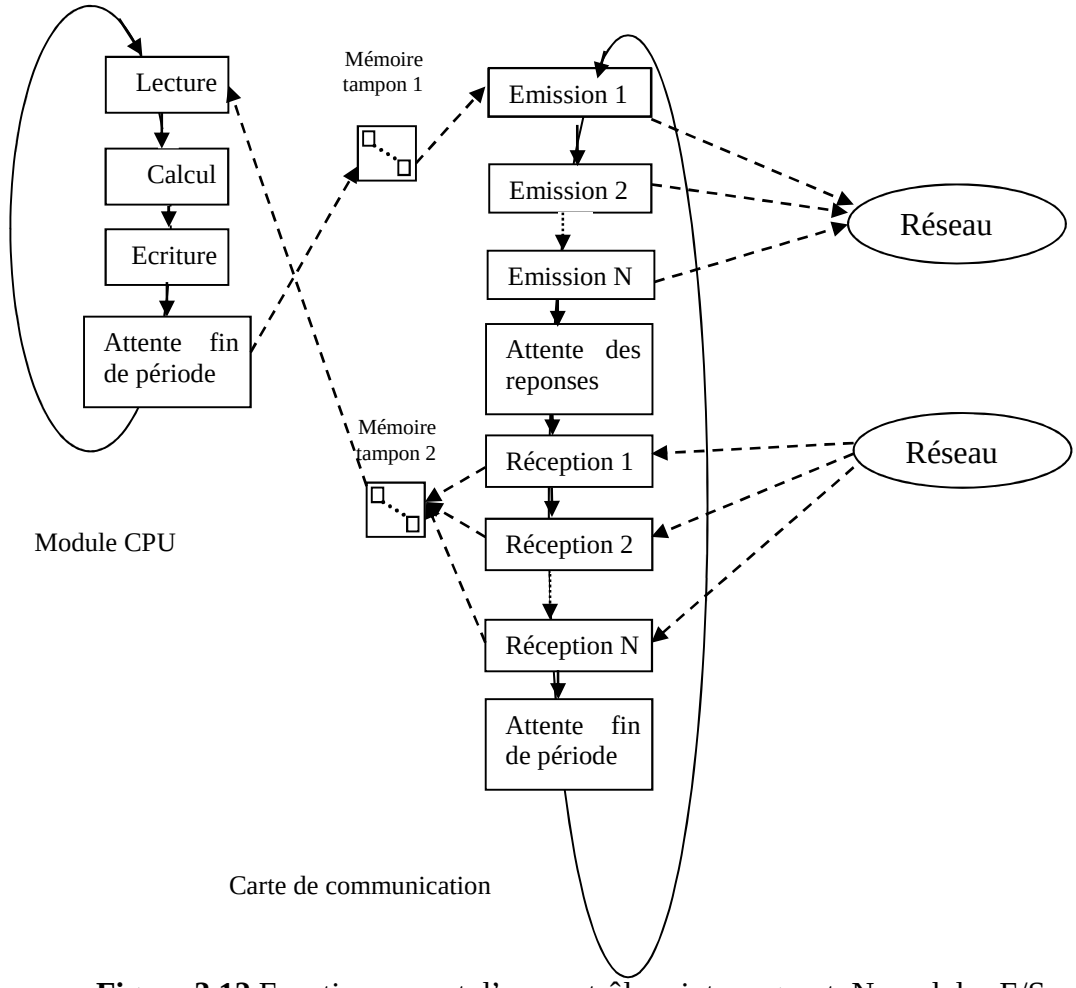


Figure 3.12 Fonctionnement d'un contrôleur interrogeant N modules E/S

De manière similaire que dans le cas mono serveur, chaque requête/réponse échangée avec un serveur subit un délai lors de la traversée du réseau de communication. Sur la Figure 3.13, seuls les délais de premier module d'E/S sont représentés. Les délais de *n* modules d'E/S sont représentés de manière similaire. Les délais notées est le temps nécessaire pour l'émission de cette requête depuis le contrôleur et le délai subi par la réponse correspondante.

Modélisation :

Le modèle des modules E/S reste quant à lui inchangé puisque chaque module ne reçoit qu'une seule requête à la fois, du seul contrôleur du l'AAR. La même chose pour le modèle de la CPU.

Le modèle complet du l'AAR est donné sur la figure Fig. 3.12.

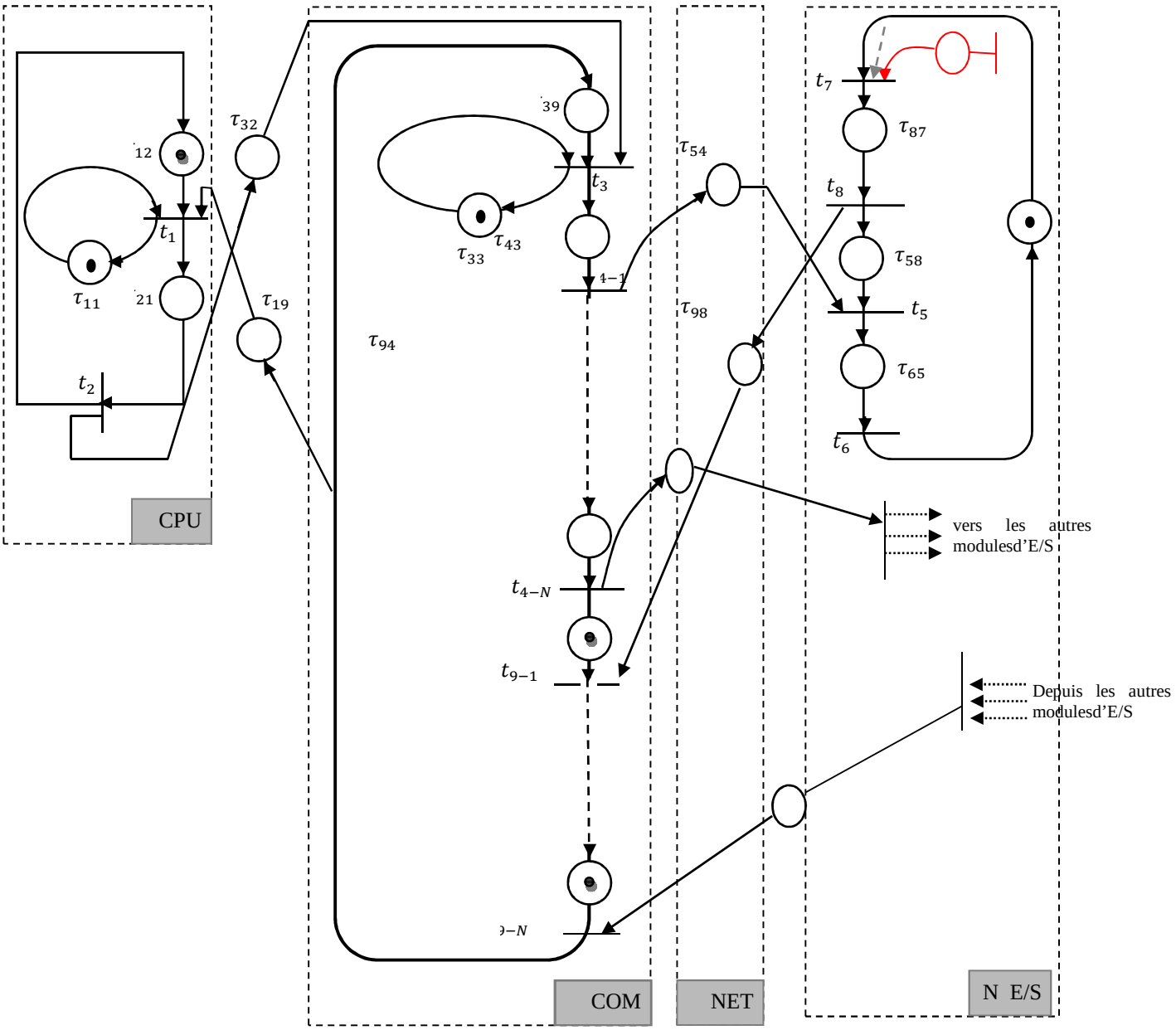


Figure 3.13 Modèle de l'AAR mono client multi serveur

Conclusion

Nous avons représenté tout d'abord les composantes et le fonctionnement de l'architecture d'automatisation en réseau dans laquelle les contrôleurs logiques communiquent avec les modules d'entrées/sorties via un réseau de communication. La deuxième partie de ce

chapitre est consacrée à la modélisation de cette AAR par un graphe d'événement temporisé et la représentation de la dynamique de cette AAR dans l'algèbre Max-Plus et Min-Plus.

Le modèle de l'AAR étant obtenu, nous pouvons passer à la synthèse de commandes des AAR permettant de satisfaire les contraintes temporelles imposées. C'est l'objet du prochain chapitre.

Chapitre 3

Commande de l'AAR sous contrainte temporelle

Dans ce chapitre nous calculons des lois de commande causale qui vont satisfaire des contraintes temporelles imposées sur le temps de réponse de l'AAR. Après avoir présenté les contraintes temporelles à satisfaire, des méthodes de calcul de lois de commande permettant de garantir le respect de ces contraintes sont discutées. Les méthodes de synthèse de ces lois de commande sont initialement développées dans [4] et [9]. Une application de la méthode à une AAR comportant des contraintes temporelles sur son temps de réponse est effectuée dans ce chapitre.

4.1Commande des systèmes à événements discrets sous contraintes temporelles

Les systèmes à événements discrets sont essentiellement caractérisés par des changements d'état dus à l'occurrence d'événements asynchrones. L'introduction de paramètres temporels permet d'évaluer leurs performances (taux de production d'une cellule d'usinage, temps de parcours dans un système de transport). Ces systèmes dynamiques peuvent également faire l'objet de contraintes de temps de séjour (temps maximal de cuisson d'une pièce dans un four, attente maximum d'une correspondance entre trains,...). Le non respect de ces contraintes conduit à des défaillances dans le système. Il est donc nécessaire, de disposer de techniques et d'outils qui permettent d'assurer ces spécifications temporelles. Dans cette partie nous développons, la méthode proposée dans [4],[9]

4.1.1Les contraintes temporelles

Généralement on distingue trois types de contraintes temporelles

- Systèmes à **contraintes temporelles critiques** le non respect de ces contraintes de temps peut conduire à des défaillances avec des conséquences pouvant être graves. Si l'échéance est dépassée, il y a faute.
- Systèmes à **contraintes temporelles souples** où le dépassement des échéances est considéré comme une faute bénigne. Lorsqu'une échéance est dépassée, il n'y a pas faute ; le résultat peut être exploitable même s'il est fourni après l'échéance.
- Systèmes à **contraintes temporelles strictes** où le dépassement occasionnel des échéances est toléré. Il y a faute (bénigne) si l'échéance n'est pas respectée.

Dans notre cas, nous nous intéressons au cas général, où il s'agit de respecter le temps pour tout événement en connaissant son moment d'occurrence et l'état du système.

Dans les graphes d'événements temporisés, la temporisation de chaque place, correspond à la durée minimale de séjour des jetons dans cette place. La durée maximale apparaît comme une contrainte supplémentaire qui devrait être vérifiée. On associe alors un intervalle du temps pour chaque place comportant une contrainte temporelle.

On considère p_{ij} la place soumise à une contrainte temporelle, un intervalle de temps $[\tau_{ij}^{min}, \tau_{ij}^{max}]$ est associé à cette place, comme illustré à la Figure 4.1.

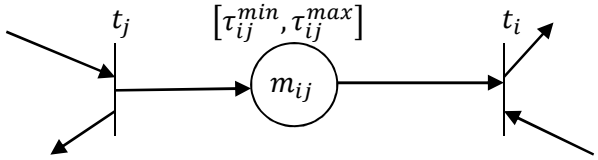


Figure 4.1 : Contrainte temporelle

Cette contrainte se traduit dans l'algèbre (min,+) par les inéquations suivantes :

$m_{ij}.x_j(t-\tau_{ij}^{min}) \geq x_i(t) \geq m_{ij}.x_j(t-\tau_{ij}^{max})$,

Où m_{ij} est le marquage initial de la place p_{ij}.

L'inéquation $x_i(t) \leq m_{ij}.x_j(t-\tau_{ij}^{min})$ est satisfaite en tenant compte de la définition d'un graphe d'événements temporisé et du modèle Min-plus linéaire associé à son comportement.

$x_i(t) \geq m_{ij}.x_j(t-\tau_{ij}^{max})$ (4.1)

L'inéquation (4.1) dont le produit est défini dans $\overline{\mathbb{R}}_{min}$, est une contrainte supplémentaire à satisfaire. La condition (4.1) impose aux jetons de passé au minimum τ_{ij}^{min} unités de temps dans la place p_{ij}. Il faudrait alors calculer des lois de commande causales u (t) qui permettent de garantir le respect de cette contrainte temporelle.

4.1.2 Commande temporelle des systèmes Min-Plus

Dans cette section nous développons une méthode de commande des systèmes Min-Plus soumis à des contraintes temporelles. Nous limitons notre étude à un système avec une seule entrée de commande et comportant une ou plusieurs places soumises à une contrainte de temps. Nous calculons pour cette classe de système des lois de commande qui permettent de satisfaire les contraintes imposées.

Dans un premier temps, nous considérons un graphe d'événements temporisé comportant une seule place p_{ij} soumise à une contrainte temporelle. Cette place admet t_j comme transition d'entrée et t_i comme transition de sortie. La transition source, notée *tu*, qui est la seule transition commandée a une seule place de sortie avec un marquage initial nul. Le comportement de ce graphe est représenté par l'équation Min-Plus linéaire (2.4). On fait l'hypothèse qu'il existe au moins un chemin α reliant la transition *tu* à la transition t_j. Nous notons par τ_α la somme des temporisations de toutes les places qui appartiennent à ce chemin, son expression est donnée par :

$$\tau_\alpha = \sum_{p\epsilon\alpha} \tau_p,$$

où τ_p est la temporisation de la place p.

A partir de la définition de la fonction compteur de la transition t_j, et du fait qu'il existe un chemin α de la transition t_u à la transition t_j, nous avons l'inégalité suivante :

$$x_j(t) \leq (A^{\tau_\alpha}.B)_j.u(t-\tau_\alpha), \qquad (4.2)$$

où $(A^{\tau_\alpha}.B)_j$ est le marquage initial du chemin α, qui represente la somme des jetons des places situées sur ce chemin et τ_α désigne le retard de la fonction $x_j(t)$ par rapport à l'unique entrée de commande u(t). L'inéquation (4.2) exprime une relation et un lien entre la transition t_j et la transition t_u. Il est clair que cette transition de commande peut agir sur le franchissement de la transition t_j et sur l'entrée des jetons dans la place p_{ij}.

Un entier $\varphi \geq 1$ étant fixé, appliquons la propriété (2.1), pour $\tau = \varphi$, à l'éqution Min-plus linéaire du graphe d'événement temporisé, on obtient l'expression explicite suivante :

$$x(t) = A^\varphi x(t-\varphi) \oplus \left[\oplus_{k'=0}^{\varphi-1} \left(A^k.B.u(t-k)\right)\right]. \qquad (4.3)$$

En considérant la i^{ème} composante des termes de cette équation, on obtient :

$$x_i(t) = \oplus_{r=1}^{N} \left((A^\varphi)_{ir}.x_r(t-\varphi) \oplus [\oplus_{k=0}^{\varphi-1} (A^k.B)_i u(t-k)\right] \qquad (4.4)$$

En remplaçant dans l'inéquation (4.1), la fonction $x_i(t)$ par son expression donnée en (4.4), la contrainte (4.1) est satisfaite si les deux inéquations suivantes sont vérifiées :

$$\oplus_{r=1}^{N} (A^\varphi)_{ir}.x_r(t-\varphi) \geq \tau_{ij}^{max} x_j(t-{\tau_{ij}}^{max})$$

$$\oplus_{k=0}^{\varphi-1} (A^k.B)_i u(t-k) \geq \tau_{ij}^{max} x_j(k-{\tau_{ij}}^{max})$$

En tenant compte de l'inégalité (4.2), les deux dernières inéquations sont vérifiées à travers les inégalités (4.5) (4.6).

$$\oplus_{r=1}^{N} (A^\varphi)_{ir}.x_r(t-\varphi) \geq m_{ij}.(A^{\tau_\alpha}.B)_j.u(t-\tau_\alpha-{\tau_{ij}}^{max}) \qquad (4.5)$$

$$\oplus_{k=0}^{\varphi-1} (A^k.B)_i u(t-k) \geq m_{ij}.(A^{\tau_\alpha}.B)_j.u(t-\tau_\alpha-{\tau_{ij}}^{max}) \qquad (4.6)$$

Le respect de la contrainte temporelle (4.1) revient donc à vérifier les inéquations (4,5) et (4,6).

L'objectif du Théorème 4.1 est de définir des lois de commande causales qui satisfont les conditions (4,5) et (4,6) et par conséquent le respect de la contrainte temporelle (4.1).

Theoreme 4.1 , [4]

L'équation

$$u(t) \leq \bigoplus_{r=1}^{N} [((A^{\varphi})_{ir} - m_{ij} - (A^{\tau_\alpha}.B)_j).x_r(t-1)],$$

définit des lois de commande causales qui garantissent le respect de la contrainte (4.1) si $\varphi = {\tau_{ij}}^{max} + \tau_\alpha + 1$ et si les deux conditions (4.7) et (4.8) sont satisfaites.

$$(A^{\varphi})_{ir} \geq m_{ij}.(A^{\tau_\alpha}.B)_j \text{ ,} \qquad \text{pour r = 1 à N} \qquad (4.7)$$

$$(A^{k}.B)_i \geq m_{ij}.(A^{\tau_\alpha}.B)_j \qquad \text{pour k = 0 à } \varphi - 1 \qquad (4.8).$$

Corollaire, [4] Il existe toujours une loi de commande causale qui satisfait la contrainte (4.1), si le marquage initial de la place p_{ij} et des places du chemin α de la transition tu à la transition t_j sont nuls. Cette loi de commande est donnée par l'équation suivante :

$$u(t) = \bigoplus_{r=1}^{N} \left[\left(A^{\Phi}\right)_{ir} . x_r\ (t-1) \right],$$

Avec $\Phi = \tau_{ij}^{\max} + \tau_\alpha + 1$

4.1.3 Commande temporelle des systèmes Max-Plus

4.1.3.1 Une seule contrainte et une seule commande

Nous considérons un graphe d'événements temporisé avec une seule transition de commande. Nous supposons qu'une place est soumise à une contrainte temporelle supplémentaire. Le comportement de ce graphe est représenté par l'équation Max-Plus linéaire (2.8) et soumise à la contrainte temporelle (4.9). La loi de commande u(k) $\in \overline{\mathbb{R}}_{\max}$.

$$x_i(k) \leq {\tau_{ij}}^{max}.x_j(k - m_{ij}) \qquad (4.9)$$

L'inéquation (4.9) exprime une borne maximale de temps de séjour des jetons dans la place p_{ij} qui admet t_j comme transition d'entrée et t_i comme transition de sortie. Nous calculons des lois de commande u(k) qui vont retarder, si nécessaire, les dates de franchissement de la transition t_j, afin de garantir le respect de la contrainte temporelle. Comme nous l'avons vu dans la section 2.3.3.2 (propriété 2.2), l'équation Max-plus linéaire (2.9) conduit à l'expression suivante :

$$x(k) = A^{\varphi}x(k-\varphi) \oplus \left[\oplus_{k'=0}^{\varphi-1}\left(A^{k'}.B\right).u(k-k')\right],$$

pour $\varphi \geq 1\ et\ k > \varphi$.

L'expression de la i^{éme} composante du vecteur x(k) est donnée ainsi par l'équation (4.10).

$$x_i(k) = \oplus_{r=1}^{N}\left((A^{\varphi})_{ir}.x_r(k-\varphi) \oplus \left[\oplus_{k'=0}^{\varphi-1}\left(A^{k'}.B\right)_i u(k-k')\right]\right. \quad (4.10)$$

En tenant compte de l'équation (4.10), la contrainte temporelle (4.9) est équivalente aux inégalités suivantes :

$$\oplus_{r=1}^{N}(A^{\varphi})_{ir}.x_r(k-\varphi) \leq \tau_{ij}^{max}x_j(k-m_{ij}) \quad (4.11)$$

$$[\oplus_{k'=0}^{\varphi-1}\left(A^{k'}.B\right)_i u(k-k') \leq \tau_{ij}^{max}x_j(k-m_{ij}) \quad (4.12)$$

Il suffit donc de vérifier les conditions (4.11) et (4.12) pour valider la contrainte temporelle (4.9). Nous supposons qu'il existe un chemin α, avec un marquage initial noté m_{α}, qui relie la transition de commande *tu* à la transition t_j. Cette hypothèse exprime une relation entre les deux transitions, ce qui se traduit par l'inéquation (4.13).

$$(A^{m_{\alpha}}.B)_j.u(k-m_{\alpha}) \leq x_j(k). \quad (4.13)$$

Nous prenons en considération l'inégalité (4.13), les inéquations (4.11) et (4 .12) sont impliquées par les inégalités (4 .14) et (4 .15)

$$\oplus_{r=1}^{N}(A^{\varphi})_{ir}.x_r(k-\varphi) \leq \tau_{ij}^{max}.(A^{m_{\alpha}}.B)_j.u(k-m_{\alpha}-m_{ij}) \quad (4.14)$$

$$\oplus_{k'=0}^{\varphi-1}\left(A^{k'}.B\right)_i u(k-k') \leq \tau_{ij}^{max}.(A^{m_{\alpha}}.B)_j.u(k-m_{\alpha}-m_{ij}) \quad (4.15)$$

Pour le cas où $m_{\alpha} = m_{ij} = 0$, on a $\varphi = m_{\alpha} + m_{ij} + 1 \Rightarrow \varphi = 1$, on a aussi k'=0 à $\varphi - 1$ ce qui implique que k'=0

L'équation (4.14) donne

$$\oplus_{r=1}^{N} (A)_{ir}.x_r(k-1) \leq \tau_{ij}^{max}.B_j.u(k)$$

Ce qui donne

$$u(k) \geq \oplus_{r=1}^{N} \left[((A)_{ir} - B_j - \tau_{ij}^{max}).x_r(k-1)\right], \quad (4.16)$$

et l'équation (4.15) donne

$$B_i \leq \tau_{ij}^{max}.B_j \quad (4.17)$$

Nous avons une condition à satisfaire (4,17) pour résoudre le problème du respect de la contrainte temporelle (4.9). Les feedbacks vérifiant l'inéquation suivante :

$$u(k) \geq \oplus_{r=1}^{N} \left[((A)_{ir} - B_j - \tau_{ij}^{max}).x_r(k-1)\right],$$

Ces feedbacks sont bien posés si $\varphi = m_{ij} + m_{\alpha} + 1$

Théorème 4.2, [9] L'inéquation

$$u(k) \geq \oplus_{r=1}^{N} F_r.x_r(k-1)],$$

avec $F_r = \max(0, A_{ir} - B_j - \tau_{ij}^{max})$, définit des lois de commande causales qui garantissent le respect de la contrainte (4.9) s'il y a un chemin vide qui relie la transition de commande à la transition de la contrainte temporelle et si la condition suivante $B_i \leq \tau_{ij}^{max}.B_j$ est bien respectée.

4.1.3.2 Plusieurs contraintes et plusieurs commandes

Nous considérons maintenant un graphe d'événements temporisé, modélisé par l'équation d'état (3.21), est soumis à Z contraintes temporelles supplémentaires, avec $Z \geq 1$. Les places contraintes sont notées p_z, pour z=1 à Z. Pour chaque place p_z, nous avons m_z, τ_z^{min} et τ_z^{max} qui représentent respectivement le marquage initial, la temporisation minimale et la temporisation maximale de cette place Figure 4.2. Les transition t_z et $t_{z'}$ correspondant aux transitions d'entrée et de sortie de la place p_z . Les fonctions $x_z(k)$ et $x_{z'}(k)$ définissent les dateurs correspondants. Nous supposons qu'il existe un

chemin, noté α_z, d'une transition de commande tu_{sz} vers la transition t_z. Le marquage initial du chemin α_z est donnée par la valeur de m_{α_z}. Ces hypothèses sont exprimées par l'inéquation (4.18)

$$x_z(k) \geq (A^{m_{\alpha z}} B)_{zs_z} u_{s_z}(k - m_{\alpha_z}), \tag{4.18}$$

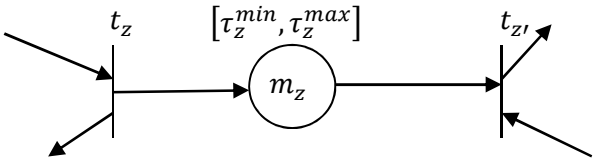


Figure 4.2 : Contrainte temporelle

La contrainte temporelle est donnée par l'inéquation suivante :

$$x_{z'}(k) \leq \tau_z^{max}.x_z(k - m_z), \tag{4.19}$$

Pour z allant de 1 à Z, l'inéquation (2.8) peut s'écrire sous la forme suivante :

$$x(k) = A^{\varphi_z} x(k - \varphi_z) \oplus \left[\oplus_{k'=0}^{\varphi_z - 1} \left(A^{k'}.B\right).u(k - k')\right]$$

Pour $\varphi_z \geq 1$, la fonction $x_{z'}(k)$ est donnée par l'équation suivante

$$x_{z'}(k) = \oplus_{r=1}^{N} \left((A^{\varphi_z})_{z'r}.x_r(k - \varphi_z) \oplus [\oplus_{k'=0}^{\varphi_z - 1} (\oplus_{l=1}^{m} \left(A^{k'}.B\right)_{z'l} u_l(k - k'))\right] \tag{4.20}$$

Nous remplaçons dans l'inéquation (4.19) de la contrainte temporelle la fonction $x_{z'}(k)$ donnée par l'équation (4.20), ce qui permet d'avoir ces deux inéquations

$$\oplus_{r=1}^{N} (A^{\varphi_z})_{z'r}.x_r(k - \varphi_z) \leq \tau_z^{max} x_z(k - m_z) \tag{4.21}$$

$$\oplus_{k'=0}^{\varphi_z - 1} (\oplus_{l=1}^{m} \left(A^{k'}.B\right)_{z'l} u_l(k - k')) \leq \tau_z^{max} x_z(k - m_z) \tag{4.22}$$

En tenant compte de l'inégalité (4.18), les inéquation (4.21) et (4.22) sont impliquées par les équations (4.23) et (4.24) .

$$\oplus_{r=1}^{N} (A^{\varphi_z})_{z'r}.x_r(k - \varphi_z) \leq \tau_z^{max}.(A^{m_{\alpha_z}}.B)_{zs_z}.u(k - m_{\alpha_z} - m_z) \tag{4.23}$$

$$\oplus_{k'=0}^{\varphi - 1} (\oplus_{l=1}^{m} \left(A^{k'}.B\right)_{z'l} u_l(k - k')) \leq \tau_z^{max}.(A^{m_{\alpha_z}}.B)_{zs_z}.u_{s_z}(k - m_{\alpha_z} - m_z) \tag{4.24}$$

Pour z=1 à Z

$$\varphi_z = m_z + m_{\alpha_z} + 1$$

(4.23) implique

$$u_{s_z}(k) \geq \bigoplus_{r=1}^{N} [((A^{\varphi_z})_{z'r} - \tau_z^{max}\_(A^{m_{\alpha_z}}.B)_{zs_z}).x_r(k-1)$$

D'après l'équation (4.24) on a

$$\left(A^{k'}.B\right)_{z'l} \leq \tau_z^{max}.(A^{m_{\alpha_z}}.B)_{zs_z} \qquad (4.25)$$

Théorème 4. 3 [9] le vecteur u(k) dont les composantes sont données par

Pour l=s_1 à s_z

$u_l(k) = \bigoplus_{z=1}^{Z} [\bigoplus_{r=1}^{N} \left(A_{z'r} - B_{zs_z} - \tau_z^{max}\right).x_r(k-1)]$,

Avec $m_z = m_{\alpha_z} = 0$, $\varphi_z = 1$,et $u_l(k) = \varepsilon$ pour $l \neq s_z$ avec z=1 à Z, définit une loi de commande causale qui garantit le respect de toutes les Z contraintes temporelles si la condition ((4.25)) est vérifiée

4.1.3.3 Exemple : nous considérons le graphe d'événements temporisé donné par la

Figure 4.3 :

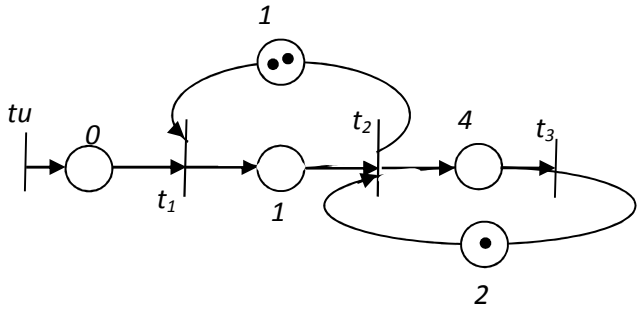


Figure 4.3 Graphe d'événements temporisé

Ce graphe est soumis à une contrainte temporelle, donnée par l'inéquation suivante :

$$2x_1(k) \geq x_2(k).$$

Nous calculons des lois de commande causales pour satisfaire cette contrainte temporelle. Nous appliquons les résultats du Théorème 4.2 pour résoudre ce problème de contrainte de temps. On ne peut pas représenter le fonctionnement du graphe d'événements temporisé considéré tel qu'il est, par une équation d'état de même type que l'équation (2.8). Pour cela, nous décomposons les places ayant plus d'un jeton pour avoir un nouveau graphe d'événements temporisé équivalant avec des places marquée à un ou à zéro, ce qui donne un marquage initial maximum $m^{max} = 1$. Nous décomposons la place p_{12} en deux places, notées p_{42} et p_{14}, qui sont marquées à 1. Le graphe étendu est donné par la Figure 4.4.

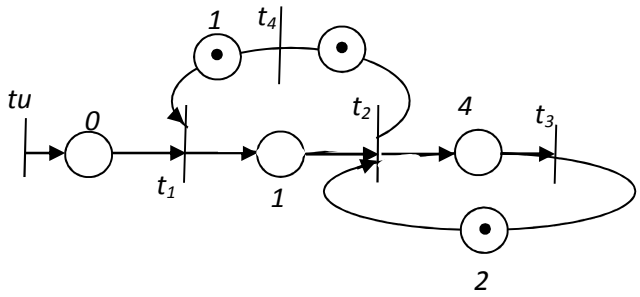


Figure 4.4 : Graphe d'événements étendu

L'équation d'état associée au graphe d'événements temporisé étendu de la figure 4.4 est donnée comme suit

$$x(k+1) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & 2 & 2 \\ \varepsilon & \varepsilon & 5 & 5 \\ \varepsilon & e & \varepsilon & \varepsilon \end{bmatrix} . x(k-1) \oplus \begin{bmatrix} e \\ 1 \\ 5 \\ \varepsilon \end{bmatrix} . u(k),$$

où les composantes du vecteur x(k) sont les fonctions des dateurs associées aux transitions $t_1,\ t_2, t_3\ et\ t_4$. La fonction u(k) est le dateur de la transition source, qui représente l'entrée de commande. Dans cet exemple, nous avons la temporisation maximale associée à la place p_{ij} qui est $\tau_{ij}^{max} = \tau_{21}^{min} = 2$ et la temporisation minimale égale à 1. On voit bien qu'il existe un chemin de la transition source *tu* à la transition t_j, dans ce cas de la transition *tu* à la transition t_1. On constate que la temporisation de ce chemin est $\tau_\alpha = 0$, et le marquage initial est $m_\alpha = 0$. l'hypothèse du théorème est bien vérifiée puisque $m_\alpha = m_{ij} = 0$. Nous avons aussi

$\tau_\alpha + \tau_{ij}^{max} = \tau_\alpha + \tau_{21}^{max} = 2,$

et $A_{ir} = A_{2r} = [\varepsilon, \varepsilon, 2,2]$ respectivement pour r=1 à 4. On voit que la condition (4.17) donnée dans le théorème 3.5 est vérifiée. D'une manière similaire, nous avons $B_i = B_2 = 1$, donc la condition du théorème est respectée aussi.

Dans cet exemple, les lois de commande données par le Théorème 4.3, vérifient l'inéquation

$u(k) \geq x_3(k-1) \oplus x_4(k-1)$,

Et garantissent le respect de la contrainte temporelle.

4.2 Application à l'AAR

Le temps de réponse [12], [13] d'une AAR dépend des différents d'attente que met une requête pour être prise en charge par les trois composants (MES, COM, CPU).En effet, une requête attend toujours que la ressource qui doit le traiter (CPU, MES, COM) soit disponible. Ces délais d'attente ont un impact direct sur la réactivité du système. On définit alors le temps de réactivité de l'AAR par le délai qui sépare l'occurrence d'un événement dans le module d'E/S et sa conséquence issue du PLC sur le système commandé. On a montré dans le chapitre deux que ce délai appartient à un intervalle [Dr_{min} Dr_{max}].

La question qui se pose alors est de trouver une loi de commande qui permet de retarder les requêtes, si nécessaire dans l'objectif de respecter la contrainte imposée sur le temps de réponse.

4.2.1 Commande de l'AAR sous contrainte temporelle dans le dioïde Min-Plus

Nous considérons le graphe d'événements temporisé de la Figure 3.10. La temporisation de la place p_{58} correspondant au temps de réponse du système vaut 11 unités de temps comme temps minimal et 22 unités de temps comme temps maximal.
Cette contrainte temporelle maximale se traduit en Min-Plus par l'inéquation suivante :

$x_5(t) \geq x_8(t-22)$ (4.25)

Nous appliquons la méthode proposée précédemment pour calculer la loi de commande u (t*)* qui garantit le respect de cette contrainte temporelle.

Nous utilisons le modèle d'état du graphe de l'AAR donné par l'équation (3.4). Nous appliquons le corollaire 1 à l'équation d'état (3.4). Ayant ($\tau_{ij}^{max} = \tau_{58}^{max} = 22$) et $\tau_\alpha = 1$, la temporisation du chemin $\alpha = (t_7, p_{87}, t_8)$ qui relie les transitions t_u et t_8, le marquage initial

de la places $m_{58} = e$. On voit que les conditions du corollaire 1 sont bien vérifiées, ce qui assure l'existence des lois de commande causales qui vérifient la contrainte temporelle. En tenant compte de l'expression donnée dans le corollaire 1, la loi de commande est donnée par :

$$\begin{aligned} u(t) = {} & 2x_1(t-1) \oplus 3x_3(t-1) \oplus 3x_4(t-1) \oplus 3x_5(t-1) \oplus 2x_7(t-1) \oplus 1x_8(t-1) \\ & \oplus 3x_{10}(t-1) \oplus 2x_{11}(t-1) \oplus 2x_{12}(t-1) \oplus 2x_{13}(t-1) \oplus 3x_{14}(t-1) \\ & \oplus 1x_{15}(t-1) \oplus 2x_{16}(t-1) \oplus 2x_{17}(t-1) \oplus 2x_{18}(t-1) \oplus 2x_{19}(t-1) \\ & \oplus 2x_{20}(t-1) \oplus 3x_{21}(t-1) \oplus 2x_{22}(t-1) \oplus 3x_{23}(t-1) \\ & \oplus 2x_{24}(t-1) \oplus 2x_{25}(t-1) \oplus 3x_{26}(t-1) \oplus 2x_{27}(t-1) \\ & \oplus 2x_{28}(t-1) \oplus 3x_{29}(t-1) \oplus 2x_{30}(t-1) \oplus 3x_{31}(t-1) \\ & \oplus 3x_{32}(t-1) \oplus 3x_{33}(t-1) \end{aligned}$$

Pour simplifier cette loi de commande nous utilisons les équations (3.3) de la partie 3.2.1.5 et nous remplaçons quelque terme dans la loi de commande, on va avoir cette loi de commande

$$\begin{aligned} & u(t) = 2x_1(t-1) \oplus 3x_3(t-1) \oplus 3x_4(t-1) \oplus 3x_5(t-1) \oplus 2x_7(t-1) \oplus \\ & 1x_8(t-1) \oplus 3x_1(t-2) \oplus 3x_1(t-2) \oplus 3x_1(t-3) \oplus 3x_1(t-4) \oplus 4x_1(t-5) \oplus \\ & 2x_3(t-2) \oplus 3x_3(t-3) \oplus 3x_3(t-4) \oplus 3x_3(t-5) \oplus 3x_3(t-6) \oplus 3x_3(t-7) \oplus \\ & 4x_3(t-8) \oplus 3x_3(t-9) \oplus 4x_3(t-10) \oplus 2x_8(t-2) \oplus 2x_8(t-3) \oplus 3x_8(t-4) \oplus \\ & 2x_8(t-5) \oplus 2x_8(t-6) \oplus 3x_8(t-7) \oplus 2x_8(t-8) \oplus 3x_8(t-9) \oplus 3x_8(t-10) \oplus \\ & 3x_8(t-11). \end{aligned}$$

En utilisant la comparaison entre les termes de la commande afin de la réduire.

Par exemple $\min(x_8(t-1), 3x_8(t-11)) = 3x_8(t-11)$.

$$\begin{aligned} & u(t) = 2x_1(t-1) \oplus 3x_4(t-1) \oplus 3x_5(t-1) \oplus 2x_7(t-1) \oplus 1x_8(t-1) \oplus \\ & 3x_1(t-5) \oplus 4x_1(t-6) \oplus 2x_3(t-2) \oplus 3x_3(t-9) \oplus 4x_3(t-10) \oplus 2x_8(t-8) \oplus \\ & 3x_8(t-11) \end{aligned} \qquad (4.26)$$

D'après les équations (3.3) de la partie 3.2.1.5 on a

$$3x_8(t-11) = 3x_6(t-12) \oplus 3u(t-12).$$

La loi de commande de l'équation 4.26 contient des éléments non observable par exemple $x_1(t)$. On remplace $3x_8(t-11)$ par sa valeur dans l'équation (4.26) et on utilise les chemins du graphe pour simplifier cette loi de commande. L'objectif de cette opération est de donner une loi de commande composée de fonctions dateurs observables.

$$u(t) = 1x_8(t-1) \oplus 2x_8(t-8) \oplus 3x_8(t-11)$$

$$u(t) = 1x_7(t-2) \oplus 2x_8(t-8) \oplus 3x_8(t-11) \qquad (4.27)$$

La loi de commande *u(t)* est un retour d'état causal qui peut être représenté par trois places de contrôle marquées et temporisées Figure 4.5. Ces places seront connectées au GET de l'AAR pour garantir le respect de la contrainte temporelle. Les transitions t_7 et t_8 qui composent la commande sont observables.

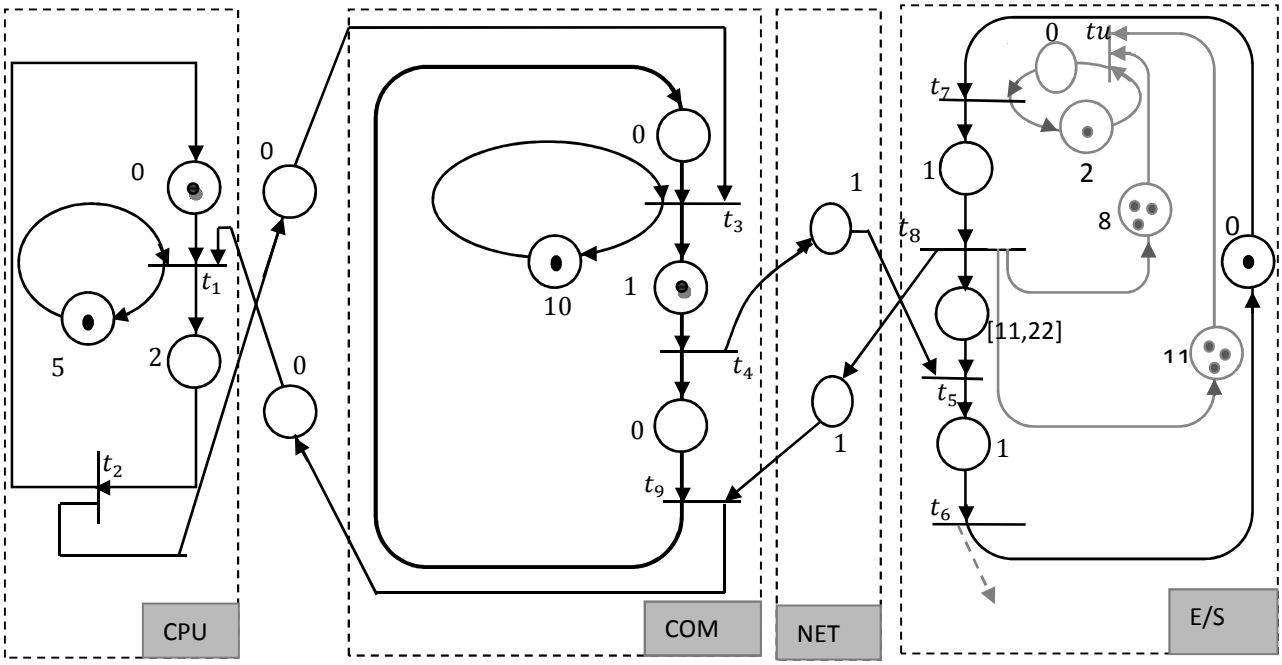


Figure 4.5 : Modèle GET de l'AAR avec les places de contrôle

4.2.2 Commande de l'AAR sous contrainte temporelle dans le dioïde Max-Plus

Nous considérons le modèle de l'AAR Figure 3.10 présenté dans la partie 3.2.1, nous avons vu dans cette partie que l'AAR comporte une contrainte temporelle sur son temps de réponse. Notre objectif est de calculer une loi de commande qui va garantir le respect de cette contrainte temporelle afin de garantir le bon fonctionnement du système. Nous appliquons dans cette partie la méthode développée dans la section 4.1.3.1 pour calculer la loi de commande.

La contrainte temporelle est donnée par $x_5(k) \leq 22x_8(k)$

L'équation d'état de l'AAR est donnée par l'équation (3.7)

D'après la figure 3.10 on voit bien qu'il existe un chemin vide de la transition de commande *tu* à la transition t_8. La temporisation de ce chemin est $\tau_\alpha = 1$ et son marquage initial est $m_\alpha = 0$. Alors l'hypothèse du Théorème est bien vérifiée puisque $m_\alpha = m_{ij} = 0$. Nous avons aussi $B_i = B_5 = 12$, $B_j = B_8 = 1$ et ${\tau_{ij}}^{max} = 22$. Les conditions de Théorème 4.2 sont bien vérifiées. La loi de commande est donnée par l'équation suivante :

$u(k) = \bigoplus_{r=1}^{9} [A_{5r} - B_8 - \tau_{58}^{max}].x_r(k-1)]$,

On a

$A_{5r} = [9 \;\; 4 \;\; 12 \;\; 4 \;\; \varepsilon \;\; 13 \;\; \varepsilon \;\; \varepsilon \;\; \varepsilon]$.

$$u(k) = e.x_1(k-1) \oplus e.x_2(k-1) \oplus e.x_3(k-1) \oplus e.x_4(k-1) \oplus e.x_6(k-1)$$

D'après les chemins du graphe d'événement temporisé de la figure 3.10 on a

Max $(e.x_1(k-1), e.x_2(k-1), e.x_3(k-1), e.x_4(k-1), e.x_6(k-1)) = x_6(k-1))$

Alors :

u(k)$= x_6(k-1) = 1.x_5(k-1))$ (4.27)

4.2.3 Interprétation des résultats

La loi de commande calculée par l'algèbre Min-plus 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 des places marquées et temporisées.

L'inconvénient de cette méthode réside dans sa lourdeur et sa complexité face à des problèmes de grande taille, plusieurs modules d'entrée sorties.

La loi de commande calculée dans l'algèbre Max-Plus est $u(k) = x_6(k-1)$, et d'après les équations (3.5), on a $x_7(k) = x_6(k-1) \oplus u(k)$, ce qui veut dire que cette loi de commande n'a pas apporté un plus ou fonctionnement de l'AAR.

La modélisation de l'AAR proposé dans le chapitre 2 ne permet pas de séparer le module d'entrée et celui de sortie (on utilise le port d'abord comme entrée puis comme sortie) or dans la majorité des systèmes industriels le port d'entrée est différent de celui de sortie. On plus la modélisation proposée ne permet pas de traiter plus d'une requête à la fois.

Dans le chapitre 4, on propose une modélisation différente de l'AAR. Cette modélisation est basée sur le protocole producteur/consommateur.

Chapitre 4

Commande de l'AAR producteur/consommateur

Pour résoudre le problème de limitation de nombre de requête traité par la CPU dans un cycle, nous proposons dans ce chapitre une autre façon de modélisation de l'AAR dans laquelle on sépare le module d'entrée et celui de sortie. Cette architecture fonctionne suivant le modèle de coopération producteur/consommateur. Nous abordons par la suite le problème de commande sous contrainte temporelle pour le cas d'un seul module d'entrée/sortie ensuite on fait une généralisation pour le cas de plusieurs modules d'entrées/sorties.

5.1 Commande de l'AAR

5.1.1Cas 1 : un seul module d'entrée sortie

5.1.1.1 Modélisation de l'architecture d'automatisation en réseau

La partie du module de CPU est modélisée exactement comme dans la première modélisation Figure 3.1. Nous faisons juste un petit changement au niveau de module d'entrée/sortie. Dans cette modélisation on sépare le module d'entrée et celui de sortie Figure 5.1 pour que le module d'entrée puisse recevoir à chaque instant les requêtes provenant de la partie opérative.

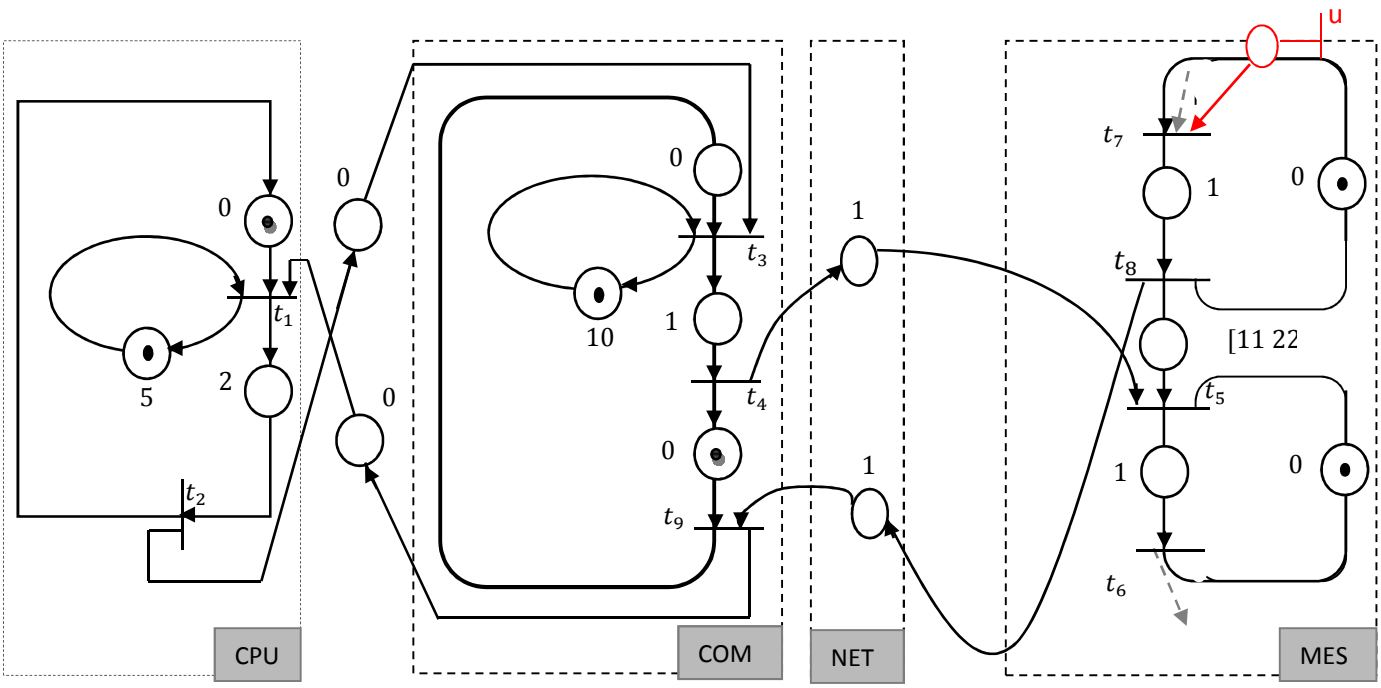


Figure 5.1 Modèle GET d'une AAR producteur/consommateur (un seul module d'E/S)

5.1.1.2 Représentation en équations Max-Plus linéaire du graphe d'événements de l'AAR

On considère un franchissement au plus tôt, c'est-à-dire que toute transition validée est immédiatement franchie. Les fonctions dateurs vérifient alors les équations suivantes :

$$\begin{cases} \theta_1(k) = \max(e + \theta_2(k-1), 5 + \theta_1(k-1), e + \theta_9(k)) \\ \theta_2(k) = 2 + \theta_1(k) \\ \theta_3(k) = \max(e + \theta_2(k), 10 + \theta_3(k-1), e + \theta_9(k) \\ \theta_4(k) = 1 + \theta_3(k) \\ \theta_5(k) = \max(11+.x_8(k), 1 + \theta_4(k)\,, e + \theta_6(k-1)) \\ \theta_6(k) = 1 + \theta_5(k) \\ \theta_7(k) = \max(e + \theta_8(k-1), e + u(k)) \\ \theta_8(k) = 1 + \theta_7(k) \\ \theta_9(k) = \max(1 + \theta_8(k), e + \theta_4(k-1)) \end{cases}$$

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

$$\begin{cases} \theta_1(k) = e.\theta_2(k-1) \oplus 5.\theta_1(k-1) \oplus e.\theta_9(k) \\ \theta_2(k) = 2.\theta_1(k) \\ \theta_3(k) = e.\theta_2(k) \oplus 10.\theta_3(k-1) \oplus e.\theta_9(k) \\ \theta_4(k) = 1.\theta_3(k) \\ \theta_5(k) = 11.\theta_8(k) \oplus 1.\theta_4(k) \oplus e.\theta_6(k-1) \\ \theta_6(k) = 1.\theta_5(k) \\ \theta_7(k) = e.\theta_8(k-1) \oplus e.u(k) \\ \theta_8(k) = 1.x\theta_7(k) \\ \theta_9(k) = 1.\theta_8(k) \oplus e.\theta_4(k-1) \end{cases} \quad (5.1)$$

Ces 9 équations peuvent s'écrire sous la forme matricielle suivante :

$$\theta(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 11 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix} \theta(k) \oplus \begin{bmatrix} 5 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \theta(\mathrm{k} -$$

$$1) \oplus \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ e \\ \varepsilon \\ \varepsilon \end{bmatrix} \mathrm{u(k)}.$$

Equation d'état

Non pouvons toujours décrire le comportement d'un graphe d'événements temporisé par une équation d'état, définie dans l'algèbre Max-plus. Nous remarquons que chaque place du graphe de la figure 4.1 comporte au plus un jeton alors il suffit de calculer A* pour avoir une équation d'état

$$x(k) = Ax(k-1) \oplus Bu(k),$$

$avec\ A = A_0^*.A_1, B = A_0^*.B_1.$

Avec :

$$A_0^* = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & 1 & e \\ 2 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 \\ 2 & e & e & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 \\ 3 & 1 & 1 & e & \varepsilon & \varepsilon & 5 & 4 & 3 \\ 4 & 2 & 2 & 1 & e & \varepsilon & 12 & 11 & 4 \\ 5 & 3 & 3 & 2 & 1 & e & 13 & 12 & 5 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & 1 & e \end{bmatrix},$$

$$x(k) = \begin{bmatrix} 5 & e & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon \\ 7 & 2 & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon \\ 7 & 2 & 10 & 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon \\ 8 & 3 & 11 & 3 & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon \\ 9 & 4 & 12 & 4 & \varepsilon & e & \varepsilon & 12 & \varepsilon \\ 10 & 5 & 13 & 5 & \varepsilon & 1 & \varepsilon & 13 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} 2 \\ 4 \\ 4 \\ 5 \\ 12 \\ 13 \\ e \\ 1 \\ 2 \end{bmatrix} u(k) , \qquad (5.2)$$

où $x(k) \in \mathbb{R}^{9}_{max}$ et $B \in \mathbb{R}^{1}_{max}$.

5.1.1.3 Commande de l'AAR

Le graphe de la Figure 5.1 est soumis à une seule contrainte temporelle qui est représentée par l'inéquation suivante :

$$x_5(k) \leq 22x_8(k) \qquad (5.3)$$

où $x_5(k), x_8(k)$ designent les fonctions dateurs respectivement des transitions t_5 et t_8.

Le problème consiste à calculer une loi de commande qui va satisfaire cette contrainte temporelle, pour cela nous appliquons les résultats du la section 4.1.3.

D'après la Figure 5.1 nous remarquons que le chemin qui relie la transition de commande à la transition t_8 (la transition d'entrée de la place contrainte) est vide $m_\alpha = m_{58} = 0$. Nous avons montré précédemment que la condition (4.8) est suffisante pour l'existence d'une loi de commande causale qui garantit le respect de la contrainte temporelle (4.3).

On a $B_i = B_5 = 12,\ B_j = B_8 = 1,\ \tau_{ij}^{max} = 22.\ \varphi = 1, k' = 0.$

$B_5 \leq 22 + 1$, alors la condition 4.8 est vérifiée.

$A_{5r} = [9\ \ 4\ \ 12\ \ 4\ \ \varepsilon\ \ e\ \ \varepsilon\ \ 12\ \ \varepsilon]$,

$u(k) = \bigoplus_{r=1}^{9} [(A_{5r} - 1 - 22)x_r(k-1)]$.

$u(k) = ex_1(k-1) \oplus ex_2(k-1) \oplus ex_3(k-1) \oplus ex_4(k-1) \oplus ex_6(k-1) \oplus ex_8(k-1)$.

D'après les chemins du graphe de l'AAR on a :

$x_6(k-1) \geq x_4(k-1) \geq x_3(k-1) \geq x_2(k-1) \geq x_1(k-1) \geq x_8(k-1).$

Alors

$$u(k) = ex_6(k-1) = 1x_5(k-1). \qquad (5.4)$$

5.1.1.4 Interprétation

La loi de commande trouvée 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

5.1.2 Cas de plusieurs modules d'entrées/ sorties

Dans ce cas, *N* modules d'entrées/sorties sont interrogés par un seul contrôleur. Durant un cycle de scrutation, les modules d'entrées envoient une rafale de *N* requêtes vers le contrôleur qui traite ces requêtes, puis il envoie ces repense vers les modules de sorties à travers le réseau de communication. Ici on garde la même modélisation sauf le nombre de module d'entrée sortie qui va changer. Au début d'un cycle CPU, toutes les entrées sont lues en bloc comme une seule variable et donc comme dans le cas d'un seul module d'entrée/sortie et les sorties sont mises à jour en bloc également. Dans cette partie c'est impossible de décrire le comportement du graphe de l'AAR avec N modules d'entrée/ sortie d'une manier générale, pour cela nous décrivons le comportement du graphe de l'AAR pour 2 et 3 modules d'E/S après nous calculons les lois de commande pour chaque cas, et à la fin on fait une comparaison entre les différentes lois de commande calculées pour avoir une loi de commande généralisé pour N modules d'E/S.

5.1.2 .1 Cas1 : deux modules d'entrées/ sorties

Dans ce cas la CPU est interrogée par deux modules d'entrées/sorties comme montré à la Figure 5.2

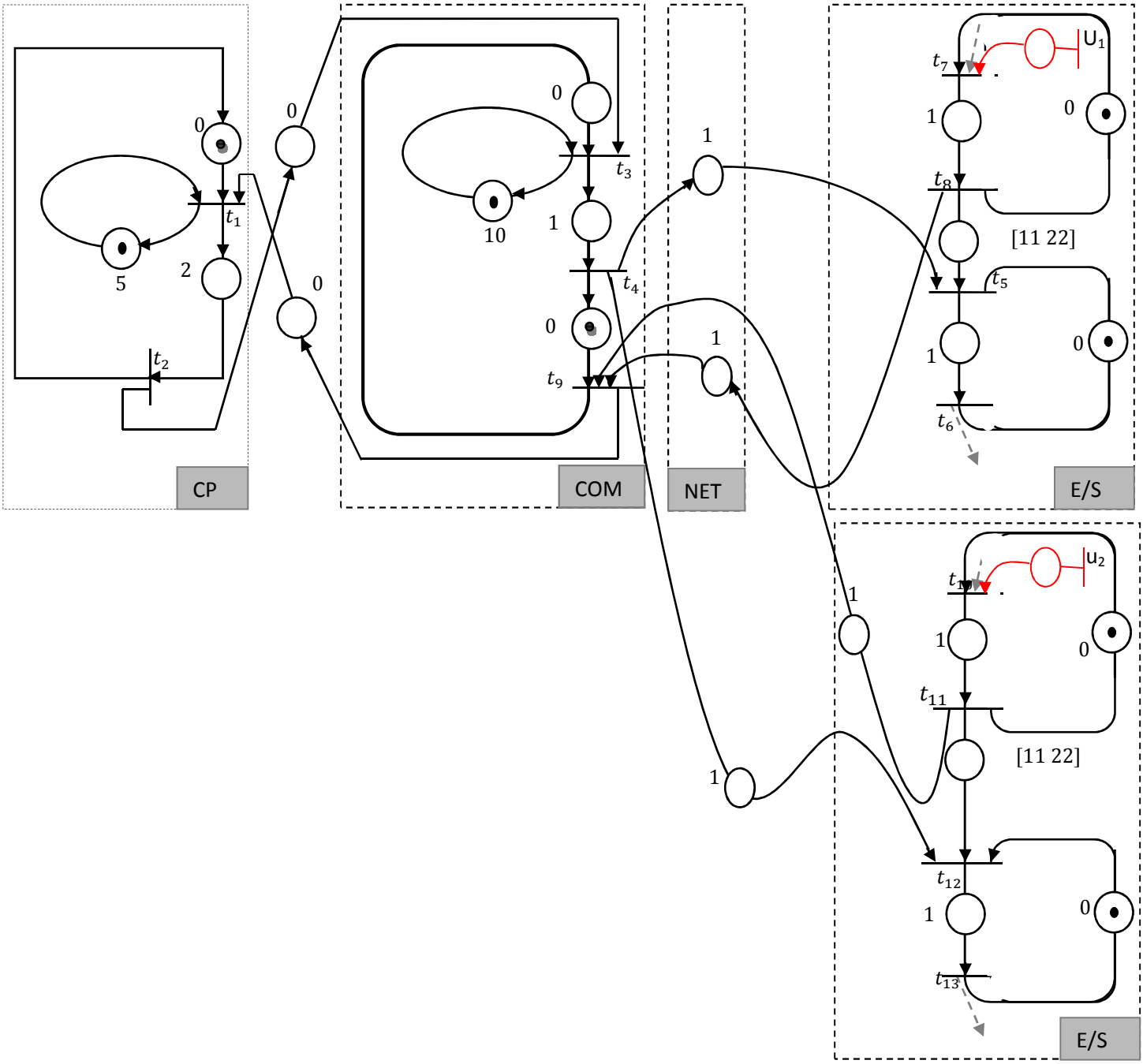


Figure 5.2 Modèle GET d'une AAR (deux modules d'E/S)

5.1.2 .1.1 Représentation en équations Max-Plus linéaire du graphe de l'AAR

L'équation implicite (2.6) associée au graphe d'événements temporisé de la Figure 5.2 est donnée par

$$\theta(k) = A_0\theta(k) \oplus A_1\theta(k-1) \oplus B_1u(k)$$

Avec

$$A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 11 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \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 & 11 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon \end{bmatrix}, A_1 \begin{bmatrix} 5 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}, B_0 \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ e & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}.$$

L'équation d'état dans ce cas, est donnée par l'équation suivante

$$x(t) = Ax(t) \oplus Bu(t)$$

Tel que $A = {A_0}^* A_1$ et $B = {A_0}^* B_1$

$$A^* = \begin{bmatrix} 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & 1 & 0 & 2 & 1 & \varepsilon & \varepsilon \\ 2 & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 & 4 & 3 & \varepsilon & \varepsilon \\ 2 & 0 & 0 & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 & 4 & 3 & \varepsilon & \varepsilon \\ 3 & 1 & 1 & 0 & \varepsilon & \varepsilon & 5 & 4 & 3 & 5 & 4 & \varepsilon & \varepsilon \\ 4 & 2 & 2 & 1 & 0 & \varepsilon & 12 & 11 & 4 & 6 & 5 & \varepsilon & \varepsilon \\ 5 & 3 & 3 & 2 & 1 & 0 & 13 & 12 & 5 & 7 & 6 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 0 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 0 & 2 & 1 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & 0 & \varepsilon & \varepsilon \\ 4 & 2 & 2 & 1 & \varepsilon & \varepsilon & 6 & 5 & 4 & 12 & 11 & 0 & \varepsilon \\ 5 & 3 & 3 & 2 & \varepsilon & \varepsilon & 7 & 6 & 5 & 13 & 12 & 1 & 0 \end{bmatrix}$$

alors

$$x(k) = \begin{bmatrix} 5 & e & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\ 7 & 2 & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon \\ 7 & 2 & 10 & 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon \\ 8 & 3 & 11 & 3 & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon \\ 9 & 4 & 12 & 4 & \varepsilon & e & \varepsilon & 12 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon \\ 10 & 5 & 13 & 5 & \varepsilon & 1 & \varepsilon & 13 & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\ 9 & 4 & 12 & 4 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon & 12 & \varepsilon & \varepsilon \\ 10 & 5 & 13 & 5 & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & 12 & \varepsilon & \varepsilon \end{bmatrix} x(k-1) \oplus \begin{bmatrix} 2 & 2 \\ 4 & 4 \\ 4 & 4 \\ 5 & 5 \\ 12 & 6 \\ 13 & 7 \\ e & \varepsilon \\ 1 & \varepsilon \\ 2 & 2 \\ \varepsilon & e \\ \varepsilon & 1 \\ 6 & 12 \\ 7 & 13 \end{bmatrix} u(k) \qquad (5.5)$$

$A \in \overline{\mathbb{R}}_{max}^{17\times17}$ et la matrice d'état du graphe d'événements de la Figure 5.2.

5.1.2 .1.2 Calcul de lois de commande

Pour ce cas (deux modules d'entrées/sorties), l'AAR est soumis à deux contraintes temporelles notées z1, z2 et qui sont représentées par les deux inéquations suivantes :

$$z_1 \quad x_5(k) \leq 22x_8(k) \tag{5.6}$$

$$z_2 \quad x_{12}(k) \leq 22x_{11}(k),$$

où $x_5(k), x_8(k), x_{12}(k)$ et $x_{11}(k)$ désignent les fonctions dateurs respectivement des transition t_5, t_8, t_{12}, t_{11}.

Nous appliquons les résultats de la partie 4.1.3.2 pour déterminer des lois de commande qui vont assurer le respect de ces deux contraintes. Nous avons z=2 (nombre de contrainte temporelle). La commande est un vecteur d'ordre 2. Pour chaque contrainte temporelle z, on vérifie d'abord qu'il existe un chemin vide α_z qui relie une transition de commande tu_{sz} et la transition t_z et aussi l'hypothèse $m_{\alpha z} = m_z = 0$ est vérifiée. $m_{\alpha z}$ est le marquage du chemin αz et m_z le marquage de la transition soumise à la contrainte temporelle z.

Pour la première contrainte z_1 ou t_z=t_8, on a $\tau_z^{max} = \tau_8^{max} = 22$, et $m_z = m_8 = 0$ (la place soumise à la contrainte est vide). On voit aussi qu'il existe un chemin vide qui relie la transition de commande tu_1 et la transition t_8 et que son marquage initial est $m_{\alpha z} = m_8 = 0$.

La deuxième contrainte z_2 pour t_z=t_{11}, on a $\tau_z^{max} = \tau_{11}^{max} = 22$. On remarque sur le graphe de Figure 5.2 qu'il existe un chemin vide allant de la transition de commande t_{u2} à la transition t_{11}. On remarque aussi que la place soumise à cette contrainte est non marquée. D'après la partie 4.1.3.2 la condition 4.25 est suffisante pour l'existence d'une loi de commande causale qui garantit le respect des contraintes temporelles

La condition est donnée comme suite

$B_{5l} \leq 22 + B_{81}$

$B_{12\,l} \leq 22 + B_{11\,1}$

Avec *l*=1 à 2 et l'opérateur + désigne l'addition usuelle, la matrice B et la matrice de commande. En tenant compte de la matrice d'état (5.5) du graphe d'événements temporisé de

la Figure 5.2, nous avons $B_{81} = 1$,et $B_{11\,1} = 1$, pour l=1 à 2 on a $B_{5l} = 12,6$ et $B_{12\,l} = 6,12$ on voit bien que la condition 4.25 est bien vérifiée dans les deux cas. Alors la loi de commande est donnée par

$u_1(k) = ex_1(k-1) \oplus ex_2(k-1) \oplus ex_3(k-1) \oplus ex_4(k-1) \oplus ex_6(k-1) \oplus ex_8(k-1) \oplus ex_{11}(k-1)$.

$u_2(k) = ex_1(k-1) \oplus ex_2(k-1) \oplus ex_3(k-1) \oplus ex_4(k-1) \oplus ex_8(k-1) \oplus ex_{11}(k-1) \oplus ex_{13}(k-1)$.

Pour l'équation de la loi de commande $u_1(k)$, on a d'après les chemins du graphe da la Figure 4.2

$x_6(k-1) \geq x_4(k-1) \geq x_3(k-1) \geq x_2(k-1) \geq x_1(k-1) \geq x_8(k-1)$

Ce qui implique que

$$u_1(k) = x_6(k-1) = 1x_5(k-1)$$

La même chose pour l'équation de la loi de commande $u_2(k)$

$x_{13}(k-1) \geq x_4(k-1) \geq x_3(k-1) \geq x_2(k-1) \geq x_1(k-1) \geq x_{11}(k-1)$.

Alors

$$u_2(k) = x_{13}(k-1) = 1x_{12}(k-1)$$

Finalement le vecteur suivant

$$u(k) = \begin{bmatrix} 1x_5(k-1) \\ 1x_{12}(k-1) \end{bmatrix} \quad (5.7)$$

Définit un feedback causal qui garantit le respect de deux contraintes temporelles z_1 et z_2.

5.1.2 .2 Cas 2 : 3 modules d'entrées/sorties

Dans ce cas la CPU est interrogée par trois modules d'entrées/sorties comme montré à la Figure 5.3

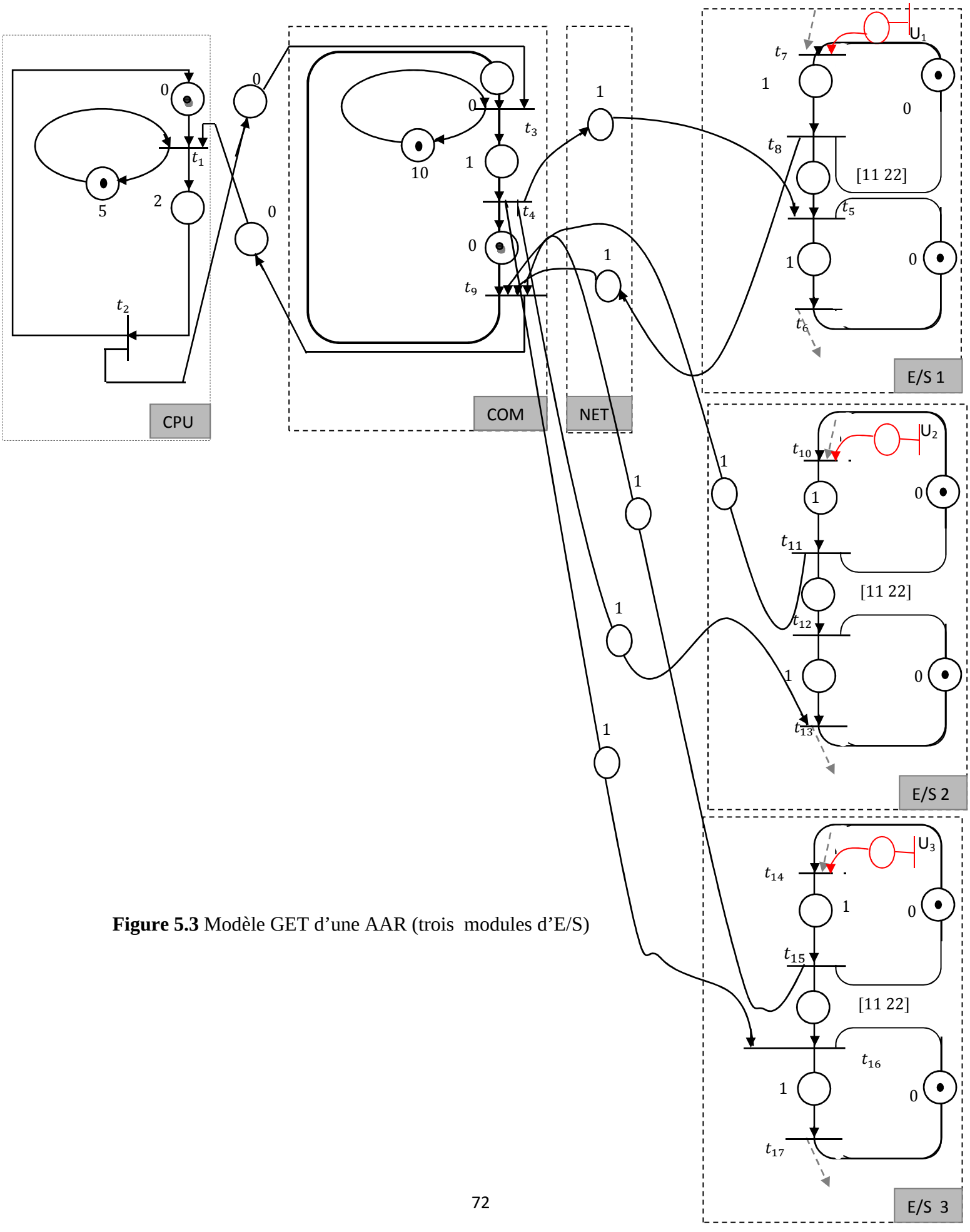


Figure 5.3 Modèle GET d'une AAR (trois modules d'E/S)

5.1.2 .2 .1 Représentation en équations Max-Plus linéaire de graphe de l'AAR

L'équation implicite (2.6) associée au graphe d'événements temporisé de la Figure 5.3 est donnée par

$$\theta(k) = A_0\theta(k) \oplus A_1\theta(k-1) \oplus B_1u(k)$$

Avec

$$A_0 = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 11 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \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 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\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 & \varepsilon & \varepsilon & \varepsilon & 11 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon
\end{bmatrix}$$

$$A_1 = \begin{bmatrix}
5 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & 10 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon
\end{bmatrix}$$

$$B_1 = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
e & \varepsilon & \varepsilon \\
\varepsilon & e & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & e \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon
\end{bmatrix}$$

L'équation d'état est donnée par l'équation suivante

$$x(t) = Ax(t) \oplus Bu(t) \qquad (5.8)$$

Tel que $A = {A_0}^* A_1$ et $B = {A_0}^* B_1$

avec

$$A^{*}=\begin{bmatrix}
e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & 1 & e & 2 & 1 & \varepsilon & \varepsilon & 2 & 1 & \varepsilon & \varepsilon \\
2 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 & 4 & 3 & \varepsilon & \varepsilon & 4 & 3 & \varepsilon & \varepsilon \\
2 & e & e & \varepsilon & \varepsilon & \varepsilon & 4 & 3 & 2 & 4 & 3 & \varepsilon & \varepsilon & 4 & 3 & \varepsilon & \varepsilon \\
3 & 1 & 1 & e & \varepsilon & \varepsilon & 5 & 4 & 3 & 5 & 4 & \varepsilon & \varepsilon & 5 & 4 & \varepsilon & \varepsilon \\
4 & 2 & 2 & 1 & e & \varepsilon & 12 & 11 & 4 & 6 & 5 & \varepsilon & \varepsilon & 6 & 5 & \varepsilon & \varepsilon \\
5 & 3 & 3 & 2 & 1 & e & 13 & 12 & 5 & 7 & 6 & \varepsilon & \varepsilon & 7 & 6 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & 2 & 1 & e & 2 & 1 & \varepsilon & \varepsilon & 2 & 1 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
4 & 2 & 2 & 1 & \varepsilon & \varepsilon & 6 & 5 & 4 & 12 & 11 & e & \varepsilon & 6 & 5 & \varepsilon & \varepsilon \\
5 & 3 & 3 & 2 & \varepsilon & \varepsilon & 7 & 6 & 5 & 13 & 12 & 1 & e & 7 & 6 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & e & \varepsilon & \varepsilon \\
4 & 2 & 2 & 1 & \varepsilon & \varepsilon & 6 & 5 & 4 & 6 & 5 & \varepsilon & \varepsilon & 12 & 11 & e & \varepsilon
\end{bmatrix}$$

$$A=\begin{bmatrix}
5 & e & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\
7 & 2 & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon \\
7 & 2 & 10 & 2 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon \\
8 & 3 & 11 & 3 & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon \\
9 & 4 & 12 & 4 & \varepsilon & e & \varepsilon & 12 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon \\
10 & 5 & 13 & 5 & \varepsilon & 1 & \varepsilon & 13 & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
9 & 4 & 12 & 4 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon & 12 & \varepsilon & e & \varepsilon & 6 & \varepsilon & \varepsilon \\
10 & 5 & 13 & 5 & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & 13 & \varepsilon & 1 & \varepsilon & 7 & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 & \varepsilon & \varepsilon \\
9 & 4 & 12 & 4 & \varepsilon & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon & 6 & \varepsilon & \varepsilon & \varepsilon & 12 & \varepsilon & e \\
10 & 5 & 13 & 5 & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & 7 & \varepsilon & \varepsilon & \varepsilon & 13 & \varepsilon & 1
\end{bmatrix}, \quad B=\begin{bmatrix}
2 & 2 & 2 \\
4 & 4 & 4 \\
4 & 4 & \varepsilon \\
5 & 5 & 5 \\
12 & 6 & 6 \\
13 & 7 & 7 \\
e & \varepsilon & \varepsilon \\
1 & \varepsilon & \varepsilon \\
2 & 2 & 2 \\
\varepsilon & e & \varepsilon \\
\varepsilon & 1 & \varepsilon \\
6 & 12 & 6 \\
7 & 13 & 7 \\
\varepsilon & \varepsilon & e \\
\varepsilon & \varepsilon & 1 \\
6 & 6 & 12 \\
7 & 7 & 13
\end{bmatrix}$$

La matrice d'état du graphe d'événements de la Figure 4.2 est $A \in \overline{\mathbb{R}}_{max}^{17\times17}$. Pour ce cas (trois modules d'entrées/sorties, l'AAR est soumis aux trois contraintes temporelles notée z_1, z_2, z_3 qui sont représentées par les inéquations suivantes :

$$z_1, \quad x_5(k) \leq 22x_8(k),$$

$$z_2, \quad x_{12}(k) \leq 22x_{11}(k), \tag{5.9}$$

$$z_3, \; x_{15}(k) \leq 22x_{14}(k),$$

Où $x_5(k), x_8(k), x_{12}(k), x_{11}(k)$ et $x_{14}(k), x_{15}(k)$ désignent les fonctions dateurs respectivement des transition t_5, t_8, t_{12}, t_{11}, t_{14}, t_{15}

Nous avons z=3 (nombre de contrainte temporelle). La commande est un vecteur d'ordre 3. Pour chaque contrainte temporelle z, on vérifie d'abord l'existence d'un chemin vide α_z qui relie une transition de commande tu_{sz} et la transition tz et aussi la vérification de l'hypothèse $m_{\alpha z} = m_z = 0$. $m_{\alpha z}$ est le marquage de chemin αz et m_z est le marquage de la transition soumise à la contrainte temporelle z.

Pour la première contrainte z_1 ou $t_z = t_8$, on à $\tau_z^{max} = \tau_8^{max} = 22$, et $m_z = m_8 = 0$ (la place soumise à la contrainte est vide), on voit aussi qu'il existe un chemin vide qui relie la transition de commande tu_1 et la transition t_8 et que son marquage initial est $m_{\alpha z} = m_8 = 0$.

La deuxième contrainte z_2 pour $t_z = t_{11}$ vérifie aussi l'hypothèse précédente, on a $\tau_z^{max} = \tau_{11}^{max} = 22$, et on remarque sur le graphe de la Figure 5.3 qu'il existe un chemin vide allant de la transition de commande t_{u2} à la transition t_{11}. On remarque aussi que la place soumis à cette contrainte est non marquée. La même chose pour la troisième contrainte $t_z = t_{15}$, $\tau_z^{max} = \tau_{14}^{max} = 22$, il existe un chemin vide allant de la transition de commande t_{u3} à la transition t_{15} alors l'hypothèse précédente est bien vérifiée.

D'après la partie (4.1.3.2), la condition (4.25) est suffisante pour l'existence d'une loi de commande causale qui garantit le respect des contraintes temporelles

La condition est donnée comme suite

$B_{5l} \leq 22 + B_{81}$

$B_{12\,l} \leq 22 + B_{11\,2}$

$B_{16\,l} \leq 22 + B_{15\,3}$

Avec *l*=1 à 2 et l'opérateur + désigne l'addition usuelle, la matrice B et la matrice de commande. On tenant compte de la matrice d'état (5.8) du graphe d'événements temporisé de la Figure 5.3, nous avons $B_{81} = 1$, et $B_{11\,2} = 1, B_{15\,3} = 1$. Pour l=1 à 2 on a $B_{5l} = 12, 6{,}6$ et $B_{12\,l} = 6, 12$,6 et $B_{16\,l} = 6{,}6{,}12$, on voit bien que la condition (4.25) est bien vérifiée dans les deux cas. Alors la loi de commande est donnée par

$$u_1(k) = ex_1(k-1) \oplus ex_2(k-1) \oplus ex_3(k-1) \oplus ex_4(k-1) \oplus ex_6(k-1) \oplus ex_8(k-1) \oplus ex_{11}(k-1) \oplus ex_{15}(k-1). \quad (5.10)$$

$$u_2(k) = ex_1(k-1) \oplus ex_2(k-1) \oplus ex_3(k-1) \oplus ex_4(k-1) \oplus ex_8(k-1) \oplus ex_{11}(k-1) \oplus ex_{13}(k-1) \oplus ex_{15}(k-1). \quad (5.11)$$

$$u_3(k) = ex_1(k-1) \oplus ex_2(k-1) \oplus ex_3(k-1) \oplus ex_4(k-1) \oplus ex_8(k-1) \oplus ex_{11}(k-1) \oplus ex_{15}(k-1) \oplus ex_{17}(k-1) \quad (5.12)$$

Pour l'équation (5.10) on a d'après les chemins du graphe da la Figure 4.2

$$x_6(k-1) \geq x_4(k-1) \geq x_3(k-1) \geq x_2(k-1) \geq x_1(k-1) \geq x_8(k-1)$$

Ce qui implique que

$$u_1(k) = x_6(k-1) = 1x_5(k-1)$$

La même chose pour l'équation (5.11)

$$x_{13}(k-1) \geq x_4(k-1) \geq x_3(k-1) \geq x_2(k-1) \geq x_1(k-1) \geq x_{11}(k-1)\ .$$

Alors

$$u_2(k) = x_{13}(k-1) = 1x_{12}(k-1)$$

Pour l'équation (5.12)

$$x_{17}(k-1) \geq x_4(k-1) \geq x_3(k-1) \geq x_2(k-1) \geq x_1(k-1) \geq x_{15}(k-1)$$

Finalement le vecteur suivant

$$u(k) = \begin{bmatrix} 1x_5(k-1) \\ 1x_{12}(k-1) \\ 1x_{16}(k-1) \end{bmatrix} \quad (5.13)$$

Définit un feedback causal qui garantit le respect de trois contraintes temporelles z_1 et z_2, z_3, ces lois de commande sont illustrées sur la Figure 5.4.

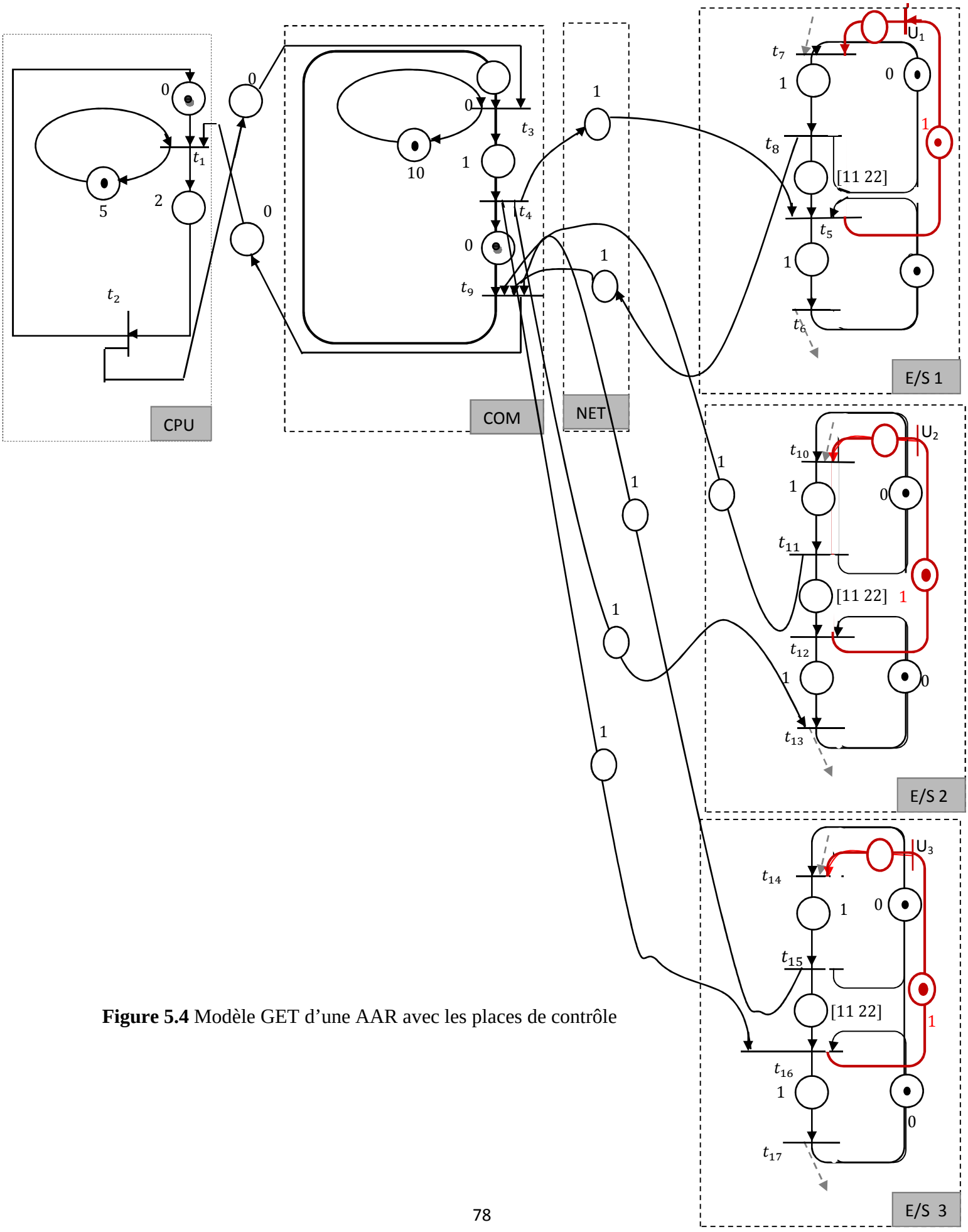


Figure 5.4 Modèle GET d'une AAR avec les places de contrôle

D'après le principe de fonctionnement de l'architecture d'automatisation en réseau la carte de communication reçoit les requêtes de la part de module d'entrée comme des paquets d'information c'est-à-dire que tous les modules d'entrées envoient les nouveaux états des entrées au même instant. On peut, alors, remplacer ces lois de commande par une seule loi donnée comme suite :

U(k)= $1x_5(k-1) \oplus 1x_{12}(k-1) \oplus 1x_{16}(k-1)$

Et pour le cas deux modules d'entrées/ sorties

U(k)= $1x_5(k-1) \oplus 1x_{12}(k-1)$

Alors pour le cas général

$$U(k) = \bigoplus_{z=1}^{N} (1x_{z'}(k-1))$$

Tel que z=1 à N est le nombre de module d'E/S et z' est la transition de sortie de la place soumise à la contrainte temporelle.

La loi de commande *u(k)* est un retour d'état causal qui peut être représenté par N places de contrôle Figure 5.5. Ces places seront connectées au GET de l'AAR pour garantir le respect des contraintes temporelles.

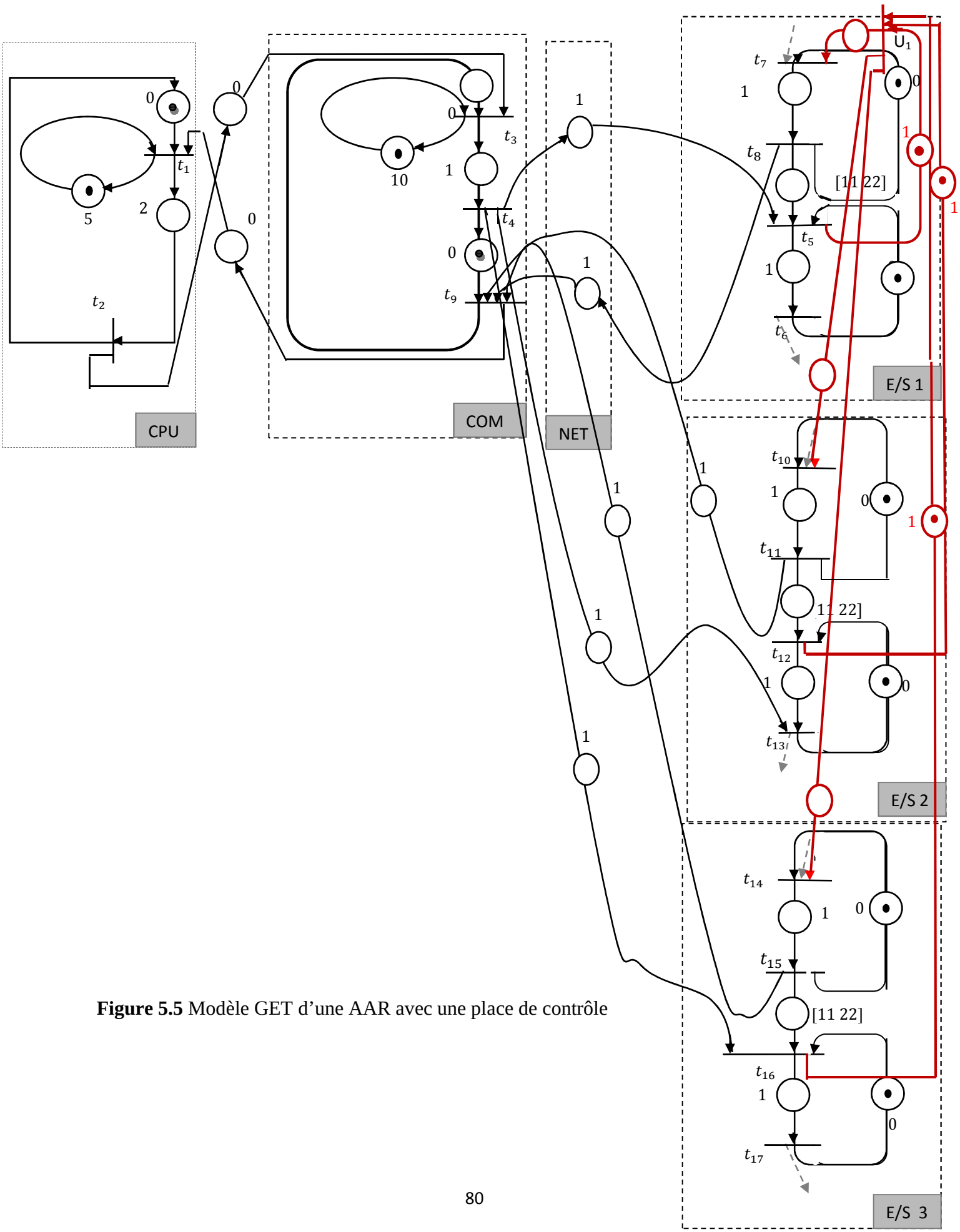


Figure 5.5 Modèle GET d'une AAR avec une place de contrôle

5.2 Conclusion

Dans ce chapitre nous avons modélisé une AAR fonctionnant suivant le protocole producteur/consommateur par un graphe d'événements temporisé. Ensuite, nous avons calculé une loi de commande, pour cette AAR, avec un seul module d'entrée/sortie. Nous avons par la suite, généralisé cette étude au cas général comportant N modules d'entrées/sorties.

Dans ce travail, nous avons abordé le problème de commande d'une architecture d'automatisation en réseau sous contraintes temporelles modélisée par les graphes d'événements temporisés et représentée par des modèles Max-plus et Min-plus linéaires.

Dans le premier chapitre de ce mémoire, nous avons donné quelques rappelle sur l'algèbre des dioïdes et sur les graphes d'événements temporisés ainsi que la représentation de leur dynamique par des modèles linéaires dans l'algèbre Max-Plus et Min-Plus.

Dans le deuxième chapitre, nous avons traité de la modélisation des AAR client/serveur. Nous avons en premier décrit le fonctionnement des différents composants technologiques utilisés. Nous avons associé un graphe d'événements temporisé à l'AAR. Ensuite, nous avons représenté le comportement de ce graphe par des modèles Max-Plus et Min-Plus linéaires. Nous avons ainsi donné les formules analytiques de calcul du temps de réponse de l'AAR.

Dans le troisième chapitre, nous avons traité le problème de la commande sous contraintes temporelles. Nous avons donné les théorèmes qui permettent le calcul des lois de commande qui vont garantir les contraintes temporelles associées au GET étudié. Le problème est étudié dans l'algèbre Max-Plus ainsi que dans l'algèbre Min-Plus. Dans la première partie nous avons abordé ce problème de commande pour les graphes d'événements temporisés avec une entrée de commande, ensuite nous avons généralisé la méthode aux cas des graphes d'événements temporisés avec plusieurs entrées de commande et qui sont soumis à plusieurs contraintes temporelles. Ensuite nous avons appliqué ces méthodes de calcul sur une AAR. Dans la première partie nous avons traité ce problème de commande pour une AAR sous contrainte temporelle dans l'algèbre Min-Plus. Les résultats de cette partie ont fait l'objet d'une communication dans une conférence internationale (MOSIM) [34]. Par la suite, nous avons abordé ce même problème dans l'algèbre Max-Plus. A la fin de ce chapitre, nous avons donné l'interprétation des résultats trouvés dans les deux parties ainsi que les inconvénients de la modélisation client/serveur. Ce qui nous a poussés à proposer une autre modélisation pour une AAR avec le protocole de communication producteur/consommateur, ce qui a fait l'objet du quatrième chapitre ; dans lequel nous avons abordé le problème de commande de l'AAR producteur/consommateur. Dans un premier temps nous avons calculé les lois de commande qui vont satisfaire les contraintes temporelle dans le cas où l'AAR est

soumise à une seule contrainte temporelle puis, nous avons généralisé les résultats pour le cas d'une AAR avec N modules d'entrée sortie et qui est soumise à N contraintes temporelles.

Comme perspectives, il serait très intéressant de mettre en œuvre ces lois de commande calculées sur une architecture d'automatisation en réseau réel.

Il serait également intéressant d'utiliser d'autres approches de commande sur les AAR et de faire ensuite une étude comparative avec les résultats qu'on a obtenus.

Nous avons abordé les seuls AAR avec un seul contrôleur, il serait intéressant maintenant de nous pencher sur les possibilités d'étendre nos résultats à des architectures d'automatisation en réseau plus complexe avec plusieurs contrôleurs et plusieurs modules d'entrées/sorties.

Comme la méthode utilisée pour le calcul des lois de commande est restreinte pour le cas où le chemin α qui relie la transition d'entrée et la place soumise à la contrainte temporelle est vide, il serait alors très intéressant d'envisager une amélioration de la technique proposée ou bien une nouvelle méthode pour les autres cas.

Bibliographie

[1] Baccelli F., G. Cohen, G. Olsder, and J. Quadrat, 1992a. *Synchronization and Linearity*. Wiley.

[2] Baccelli F., G. Cohen et B. Gaujal, 1992b.*Recursive equations and basic properties of timed Petri nets. Discrete Event Dynamic Systems*: Theory and Applications, l(4).

[3] Cohen, G., S. Gaubert, and J. P. Quadrat, (1999). *Max plus algebra and systems theory: wherewe are and where to go now,* Annual Reviews in Control, vol. 23, pp. 207-219.

[4] Said Amari *''Commande des graphes d'événements temporises sous contraintes temporelles''* Thèse de doctorat, Université de Nantes, 2005

[5] Mehdi Lhommeau'' *Étude de systèmes à événements discrets dans l'algèbre (max;+)''* Thèse de doctorat, l'ISTIA - Université d'Angers, 2003.

[6] Silvain RUEL, *''Evaluation des bornes des performances temporelles des Architectures d'Automatisation en Réseau par preuves itératives de propriétés logiques''*, Thèse de doctorat, ENS Cachan, 2009

[7] Boussad ADDAD, *''Evaluation analytique du temps de réponse des systèmes de commande en réseau en utilisant l'algèbre (max,+)''*, Thèse de doctorat, ENS Cachan, 2011.

[8] Addad B., S.Amari, 2008. *Modeling and response time evaluation of ethernet-based control architectures using timed event graphs and Max-plus algebra*. 4th IEEE CASE, Washington DC,

[9] Amari S., I. Demongodin, J.-J. Loiseau and C. Martinez, *« Max-Plus Control Design for Temporal Constraints Meeting in Timed Event Graphs »*. IEEE Trans. on Automatic Control, volume: 57, Issue : 2, Pages: 462 – 467. 2012

[10] Gondran, M. and Minoux,2001. *Graphes, dioïdes et semi-anneaux*. Tec & Doc, Paris, France.

[11]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.

[12] Gaubert S."*Théorie des systèmes linéaires dans les dioïdes*" . Thèse de doctorat, Ecole des Mines de Paris, 1992.

[13] Cohen, G. (1998a). Residuation and Applications. In *Algèbres Max-Plus et applications en informatique et automatique, Ecole de printemps d'informatique théorique*, Noirmoutier, France.

[14] Cottenceau, B. (1999). *Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes*. Thèse, LISA - Université d'Angers.

[15] Menguy, E. (1997). *Contribution à la commande des systèmes linéaires dans les dioïdes*. Thèse, LISA - Université d'Angers.

[16] M.K. Didi-Alaoui. *Étude et supervision des graphes d'événements temporisés et temporels : vivacité, estimation et commande*. Thèse, ISTIA, Université d'Angers, Décembre 2005.

[17] Cohen, G., Dubois, D., Quadrat, J.-P., and Viot, M. (1985). *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.

[18] Le Boudec, J.-Y. and Thiran, P. (2001). *Newtork Calculus*. Springer- Verlag. http://ica1www.epfl.ch/PS_files/NetCal.htm.

[19] Houssin, L. (2003). *Contribution à l'étude des systèmes à événements discrets dans l'algèbre des dioïdes. Applications aux systèmes de transport.* Rapport de stage de DEA, DEA Automatique et Informatique Appliquée, Nantes-Angers, France.

[20] Amari S., I. Demongodin and J-J. Loiseau, "*Sizing, cycle time and plant control using dioïd algebra"*. Chapter 6 in Supply Chain Optimization, Series Applied Optimization, A. DolguiJ.Soldek and O. Zaikin (Eds), Springer, pp. 71-85, 2004.

[21] Dasarathy B.,1985. *Timing constraints of real-time systems: constructs for expressing them, methods for validating them.*IEEE T.S.E., 17:34-44, 1985.

[22] Van den Boom T. and B. De Schutter, 2004. *Modeling and Control of railway networks.* Proc. of American Control Conference, Boston, pp. 5728-5733.

[23] Houssin L., S. Lahaye, and J.L. Boimond.2007. *Just in time control of constrained (max, +)-linear systems*. Journal of Discrete Event Dynamic Systems(17):pp 159–178, 2007.

[24] Diouri I., 2010. *Propositions de méthodes pour adapter le réseau aux contraintes d'applications temps-réel*. Thèse de doctorat, Université Henri Poincaré, Nancy 1.

[25] Addad B.,S.Amari and J-J.Lesage, 2010 .*Analytic calculus of response time in networked automation*. IEEE Transactions on Automation Science and Engineering, volume 7, issue 4, Page(s): 858 – 869.

[26] Lee K. C. and S. Lee, 2002. *Performance evaluation of switched Ethernet for real-time industrial communications,"* Computer standards & interfaces, (24):411–423.

[27] Greifeneder J., G. Frey, 2006. *Optimizing quality of control in networked automation systems using probabilistic models*. in Proc. of 11th IEEE Int. Conf. on ETFA, Prague.

[28] Brandin B.A and W.M Wonham.1994. *Supervisory control of timed discrete event systems .IEEE Transactions on Automatic Control*, Vol. 39 (2), p.p. 329-341.

[29] Cottenceau B., L. Hardouin, J.L. Boimond, and J.L. Ferrier, « *Synthesis of Greatest linear feedback for timed event graphs in dioid* ». IEEE Transactions on Automatic Control, Vol. 44(6), p.p.1258-1262, 1999.

[30] Lahaye S., B. Cottenceau, A. Correïa, « *Commande de graphe d'événements temporisés avec contraintes de temps critique »*. CIFA, Tunisie, 2004.

[31] Amari S., J-J.Loiseau, and I. Demongodin, 2005a. *Control of linear Min-plus systems under temporal constraints*. In *IEEE CDC-ECC, pp. 7738–7743*. Seville, Spain.

[32] Amari S., I. Demongodin et J.J. Loiseau, 2005b. *Méthode formelle de commande sous contraintes de temps dans les dioïdes*. MSR'05, Grenoble.

[33] Atto A. M., C. Martinez and S. Amari, « *Control of Discrete Event Systems with Respect to Strict Duration: Supervision of an Industrial Manufacturing Plant »*. Computers & Industrial Engineering, Volume 61, Issue 4, November 2011, Pages 1149-1159.

[34] *K.TEBANI, S.AMARI, R.KARA "Modélisation et commande d'une architecture d'automatisation en réseau sous contrainte temporelle en utilisant l'algèbre Min-Plus"*, 9th International Conférence on Modeling, Optimization & SIMulation Bordeaux, MOSIM'12, 9 p, France, 6-8 June, 2012.

[35] Maia C.A., C.R. Andrade, L. Hardouin, *On the control of max- plus linear system subject to state restriction.* Automatica. Volume 47, Issue 5, Pages 988-992,2011,