



HAL
open science

Modélisation et commande d'une architecture d'automatisation en réseau sous contrainte temporelle en utilisant l'algèbre Min-Plus

Amari Saïd, Tebani Karima, Kara Redouane

► To cite this version:

Amari Saïd, Tebani Karima, Kara Redouane. Modélisation et commande d'une architecture d'automatisation en réseau sous contrainte temporelle en utilisant l'algèbre Min-Plus. 9th International Conference on Modeling, Optimization & SIMulation, Jun 2012, Bordeaux, France. hal-00728625

HAL Id: hal-00728625

<https://hal.science/hal-00728625>

Submitted on 30 Aug 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MODELISATION ET COMMANDE D'UNE ARCHITECTURE D'AUTOMATISATION EN RESEAU SOUS CONTRAINTE TEMPORELLE EN UTILISANT L'ALGEBRE Min-Plus

Karima Tebani³, Saïd Amari^{1,2}

Redouane Kara³

¹Laboratoire LURPA, ENS Cachan, 61 av. du Président Wilson, 94235 Cachan Cedex, France

²Université Paris XIII, place 8 mai 1954, Saint-Denis
tebani_karima@yahoo.fr, samari@ens-cachan.fr

³Laboratoire de Conception et Conduite des systèmes de Production

Université Mouloud Mammeri de Tizi-Ouzou, Algérie
redouk@yahoo.fr

Résumé – Dans cet article, nous calculons des lois de commande en boucle fermée pour satisfaire une contrainte temporelle dans une architecture d'automatisation distribuée en réseau. L'approche de commande utilisée, est appliquée sur un graphe d'événements temporisés qui modélise tous les composants de l'architecture. Le comportement dynamique de l'architecture est représenté par un système d'équations linéaires dans l'algèbre Min-Plus. La contrainte temporelle à garantir est exprimée par une inéquation Min-Plus. Les lois de commande calculées sont des feedbacks causaux qu'on peut représenter par des places de contrôle.

Mots clés — Architecture d'Automatisation en Réseau, Graphes d'Événements Temporisés, Algèbre Min-Plus, Contrainte temporelle, Modélisation, Commande.

1 INTRODUCTION

Toutes les applications industrielles ont une caractéristique commune qui réside dans l'existence de 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 thermiques ou chimiques [Amari S. et al., 2004], [Kim J-H et al., 2003], dans les systèmes embarqués [Dasarathy B., 1985], dans les réseaux de transport urbain ou ferroviaire [Van den Boom T. et De Schutter B., 2004], [Houssin L., 2007] et dans les systèmes de commande en réseau [Diouri L., 2010]. 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 [Addad B. et Amari S., 2008], [Addad B. et al., 2010], [Ruel S. 2009], [K.C. Lee et Lee S. 2002], [Greifeneder J et Frey G., 2006], [Gianluca C. 2006]. 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 [Brandin B.A et W.M Wonham,1994], 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 [Cottenceau B. et al,1999], [Lahaye S. et al.,2004], [Amari S et al., 2005a],[Amari S et al,2005b], [Atto A.M et al,2011].

Les auteurs [Cottenceau B. et al 1999], [Lahaye S. et al 2004], 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 [Amari S. et al, 2005a], [Amari S. et al., 2005b], [Atto A. M. et al., 2011] 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 [Amari S. et al., 2005a], [Amari S. et al., 2005b] pour résoudre le problème de commande de l'AAR sous contrainte temporelle.

Ce papier est structuré de la manière suivante. Dans la section 2, nous rappelons quelques définitions et notations de base concernant l'algèbre Min-Plus, les graphes d'événements temporisés et les systèmes Min-Plus linéaires. Le fonctionnement et la modélisation de l'AAR sont présentés dans la section 3. La section 4 est consacrée à la commande sous contrainte temporelle de l'AAR. La section 5 est réservée à la conclusion et aux perspectives.

2 RAPPELS

2.1 Algèbre Min-Plus

Nous présentons dans cette section quelques définitions et notions de base de l'algèbre Min-Plus. Ces rappels sont issus des références [Baccelli F. et al.1992a], [Baccelli F. et al.1992b].

Monoïde : Un ensemble D muni d'une loi interne, notée \oplus , est un dit monoïde si cette loi est associative et admet un élément neutre noté ε , tel que $\forall a \in D, a \oplus \varepsilon = \varepsilon \oplus a = a$.

Un semi anneau : Un semi anneau est un monoïde qui est muni d'une seconde loi interne notée \otimes . Cette deuxième loi est associative, distributive par rapport à la loi \oplus et admet un élément neutre notée e , ($\forall a \in D, a \otimes e = e \otimes a = a$).

Dioïde : Un semi anneau D est un dioïde si la loi interne \oplus est idempotent c'est-à-dire : $\forall a \in D, a \oplus a = a$.

Le dioïde est dit commutatif si la loi \otimes est commutative.

Algèbre Min-Plus : Le dioïde commutatif Min-Plus (ou algèbre Min-Plus) est noté $(\mathbb{R}, \min, +)$. Dans ce dioïde, l'ensemble $\mathbb{R} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$, la loi \oplus correspond à l'opérateur minimum ($1 \oplus 2 = 1$) avec un élément neutre ($\varepsilon = +\infty$) et la deuxième loi \otimes est l'addition classique ($1 \otimes 2 = 1.2 = 3$) avec $e = 0$ comme élément neutre. Dans la suite, nous remplaçons le produit Min-Plus \otimes par un point $(.)$.

Cette algèbre est étendue aux cas matriciels. Soit (D, \oplus, \otimes) un dioïde, on définit une matrice sur l'ensemble $D^{p \times n}$ avec $p, n \in \mathbb{N}$. La somme matricielle de deux matrices $A, B \in D^{p \times n}$, est une matrice notée $A \oplus B$, avec des coefficients donnés par $(A \oplus B)_{ij} = A_{ij} \oplus B_{ij}$.

Les nombres $p, n, q \in \mathbb{N}$ et les matrices $A \in D^{n \times p}, B \in D^{p \times q}$ étant donnés, le produit matriciel de ces deux matrices est une matrice notée $A \otimes B$ ou $A \cdot B$ avec des coefficients donnés par la formule $(A \otimes B)_{ij} = \bigoplus_{k=1}^p (A_{ik} \otimes B_{kj})$.

L'élément nul est la matrice composée exclusivement de ε . L'élément unité, noté I_n est la matrice dont les éléments de la diagonale sont égaux à e et ε ailleurs.

L'étoile de Kleene d'une matrice carrée A , notée A^* , est définie comme suit :

$$A^* = \bigoplus_{i \in \mathbb{N}} A^i$$

Avec A^0 est la matrice unité.

2.2 Graphes d'événements temporisés

Un graphe d'événements est un réseau de Petri ordinaire où chaque place possède exactement une transition d'entrée et exactement une transition de sortie. Un graphe d'événement est dit temporisé (GET) si des temporisations sont associées aux transitions et/ou aux places. Dans ce qui suit, on considère des graphes d'événements P-temporisés (les temporisations sont associées aux places).

Nous adoptons les notations suivantes : P , l'ensemble des places du GET ; T , l'ensemble des transitions du graphe. Le nombre de transitions ayant des places en amont et en aval est noté n , et m désigne le nombre de transitions sources qui n'ont pas de places en amont. Pour $t_i, t_j \in T$, la place p_{ij} qui relie la transition t_i à t_j est unique. Si elle existe, la temporisation correspondante est notée τ_{ij} et le marquage initial de cette place est noté m_{ij} . La plus grande temporisation du graphe d'événements considéré est τ^{max} .

Un chemin α allant de la transition t_u à la transition t_j est une séquence de transitions et de

places $(t_u, p_{k1u}, t_{k1}, p_{k2k1}, t_{k2}, \dots, t_j)$. On note par τ_α la somme des temporisations des places le long du chemin α .

2.3 Modèle Min-plus linéaire

Le comportement d'un graphe d'événement temporisé peut s'exprimer sous forme d'équations linéaires dans le dioïde Min-Plus (\mathbb{R}_{min}). Pour cela, on associe à chaque transition du graphe une fonction appelée compteur, qui correspond au nombre cumulé de franchissement de cette transition à l'instant t . Les compteurs correspondants aux transitions sources sont les composantes du vecteur $u(t) \in \mathbb{R}_{min}^m$, et les compteurs des autres transitions composent le vecteur $\theta(t) \in \mathbb{R}_{min}^n$. En général, on s'intéresse à l'évolution au plus tôt des graphes d'événements temporisés, c'est-à-dire que toute transition validée est immédiatement franchie. Il est bien montré dans [Bccelli F. et al., 1992b] que le comportement dynamique d'un graphe d'événements temporisé peut s'exprimer sous forme d'équation linéaire dans le dioïde \mathbb{R}_{min} , donnée comme suit :

$$\theta(t) = \bigoplus_{\tau=0}^{\tau_{max}} (A_\tau \cdot \theta(t - \tau) \oplus B_\tau \cdot u(t - \tau)) \quad (1)$$

où $A_\tau \in \mathbb{R}_{min}^{n \times n}$ 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 ete sinon. Les termes des matrices $B_\tau(k) \in \mathbb{R}_{min}^{n \times m}$ correspondent aux marquages initiaux des places de sortie des transitions sources. L'équation (1) est implicite en général. Elle est souvent remplacée par sa solution explicite suivante,

$$\theta(t) = \bigoplus_{\tau=1}^{\tau_{max}} (A_0^* \cdot A_\tau \theta(t - \tau) \oplus A_0^* B_\tau u(t - \tau)) \quad (2)$$

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

Par analogie avec la théorie des systèmes linéaires classiques, l'équation explicite (2) peut être transformée en une forme d'état. Pour obtenir un modèle d'état, nous décomposons toutes les places dont la temporisation $\tau > 1$ en τ places temporisées à 1, et nous ajoutons $(\tau-1)$ transitions intermédiaires. Le comportement dynamique du graphe d'événements temporisé étendu est décrit par une équation de la forme :

$$x(t) = A_0 \cdot x(t) \oplus A_1 \cdot x(t - 1) \oplus B \cdot u(t)$$

qui peut s'écrire par la forme explicite ou d'état : $x(t) = A \cdot x(t - 1) \oplus B \cdot u(t)$ (3)

Avec $A = A_0^* \cdot A_1$ et $B = A_0^* \cdot B$

Exemple : Nous considérons le graphe d'événements temporisé de la Figure 1 (a). Pour cet exemple, nous avons : $\tau^{max} = 2, m^{max} = 2$.

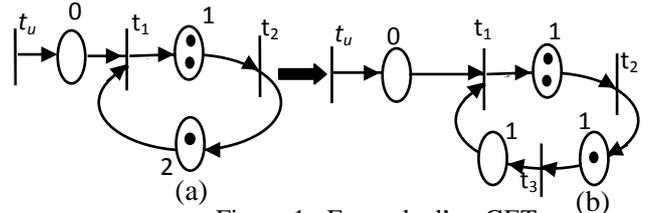


Figure 1 : Exemple d'un GET

Nous considérons un franchissement au plus tôt, c'est-à-dire que toute transition validée est immédiatement franchie. Les fonctions compteurs des transitions du GET de la Figure 1 (a) vérifient les équations suivantes :

$$\begin{cases} \theta_1(t) = \min(0 + u(t), 1 + \theta_2(t)) \\ \theta_2(t) = 1 + \theta_1(t - 1) \end{cases}$$

Ces équations deviennent linéaires dans l'algèbre Min-Plus :

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

Ce qui donne l'équation matricielle explicite suivante :

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

Pour obtenir une représentation d'état, nous décomposons la place temporisée à deux unités de temps en deux places, chacune temporisée à une unité de temps et nous ajoutons la transition intermédiaire t_3 . Nous obtenons alors le GET étendu de la Figure 1(b) L'équation d'état est donnée comme suit :

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

3 MODELISATION DE L'AAR PAR UN GET

3.1 Présentation de l'AAR

La structure physique des architectures d'automatisation en réseau AAR est illustrée par la Figure 2 Cette architecture de commande fonctionne suivant le protocole TCP/IP [Neumann P., 2007] qui utilise les requêtes Modbus standard (lecture, écriture et lecture/écriture) qui sont encapsulées dans une trame Ethernet.

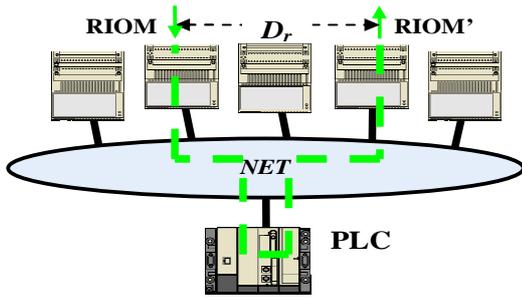


Figure 2 : Architecture d'Automatisation en réseau

L'AAR permet des communications de type client /serveur entre les différents composants connectés au réseau, les contrôleurs étant les clients, et les modules d'entrée/sortie (E/S) sont des serveurs. L'automate programmable PLC ou le contrôleur joue deux rôles dans l'AAR, il exécute un programme qui permet de déterminer le nouvel état des sorties de l'état antérieur de ses variables et en fonction de l'état courant des entrées. Il doit également gérer les échanges avec les modules E/S pour récupérer les valeurs des entrées et transmettre les valeurs de sortie. Il est de ce fait décomposé en deux parties, le processus de calcul (CPU), et la carte de communication (COM). Les deux parties du contrôleur fonctionnent de manière cyclique. Le processeur de traitement exécute périodiquement les tâches : lecture, traitement et écriture avec comme seule contrainte, la période du cycle à ne pas dépasser.

La carte de communication de son côté, elle commence par copier toutes les sorties calculées par le CPU de la mémoire tampon entre le CPU et la carte de communication dans sa mémoire propre puis elle envoie des requêtes aux modules E/S et attend que les réponses reviennent. Si avant de terminer le temps de cycle toutes les réponses sont arrivées, il se met en attente, sinon il recommence directement un nouveau cycle de scrutation. Le temps de réponse (D_r) ou de réactivité est le délai qui s'écoule entre la variation d'une entrée ou de l'état des capteurs (événement déclencheur) et sa conséquence sur les actionneurs du système (émission de la consigne de sortie correspondante) Figure2 [Addad B., Amari S. 2008]. Deux cas peuvent être considérés dans une architecture de commande : le temps entre le moment de détection d'un danger et le déclenchement d'une alarme. Dans ce cas, la borne maximale du temps de réaction est très importante à satisfaire parce que la violation de cette contrainte temporelle peut conduire à des défaillances avec des conséquences graves sur le système. C'est ce qu'on appelle les contraintes strictes. A l'inverse le temps réel souple s'accommode de dépassements des contraintes temporelles dans certaines limites au-delà desquelles le système devient inutilisable. 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.

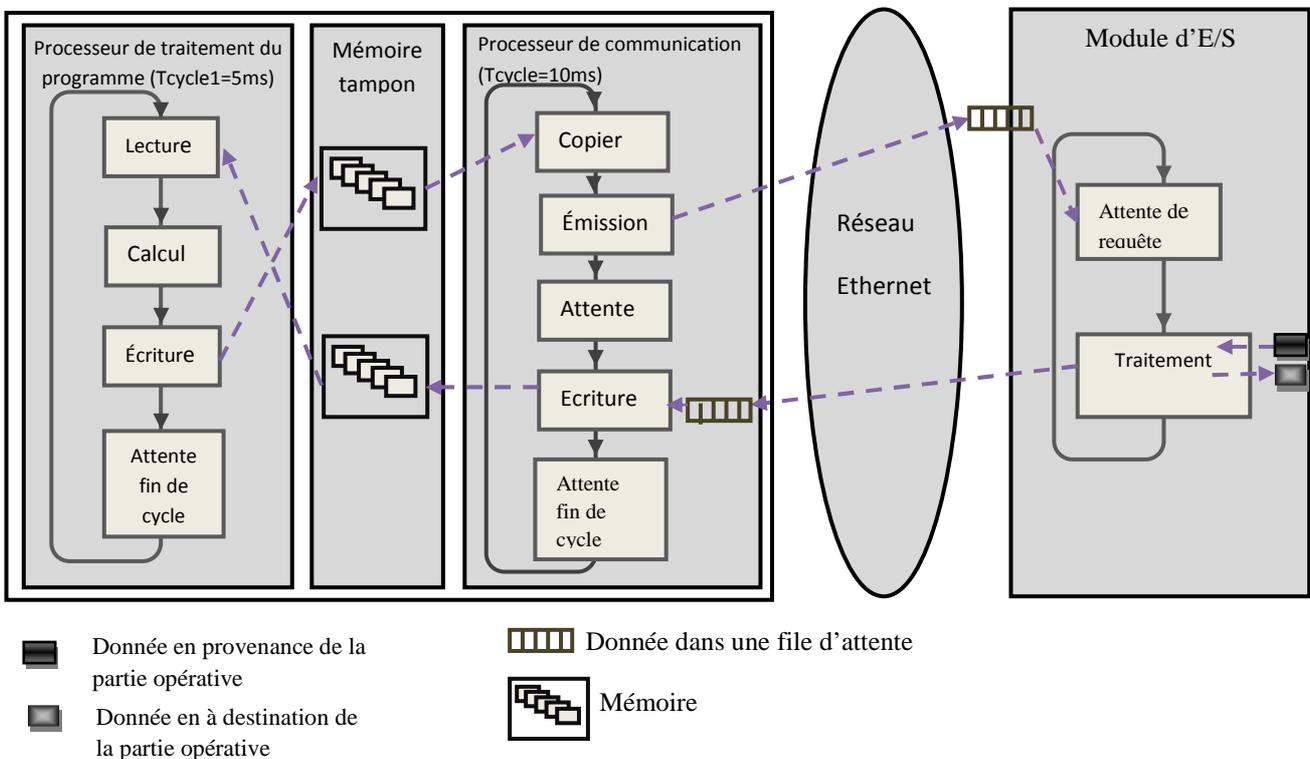


Figure 3 : Comportement de l'AAR

Hypothèse : Nous considérons dans ce travail une AAR composée d'un seul PLC, un module d'E/S et le réseau. Dans notre étude, nous supposons que la CPU est dédiée, c'est-à-dire elle traite juste les informations qui proviennent du module d'E/S.

3.2 Modèle GET de l'AAR

Compte tenu du fonctionnement de l'architecture de commande décrit dans la section III.1 et en tenant compte des travaux effectués dans [Addad B. et al., 2010], [Addad B., Amari S., 2008], nous modélisons l'AAR par un graphe d'événements P-temporisé de la Figure 4. Ce GET traduit directement le modèle fonctionnel de l'AAR et représente tous les composants. Il comporte trois parties essentielles, la partie de CPU et celle de la carte de communication. Ces deux parties sont liées par deux mémoires tampon. La troisième partie représente le module E/S. Cette dernière est liée à la carte de communication via un réseau de terrain qui assure les échanges entre ces deux modules.

Un message envoyé par un capteur dans une trame Ethernet pour être envoyée vers le contrôleur et le retour de la réponse de ce dernier vers la partie opérative peut simplement être représentée par une seule place avec un délai égal au retard global induit par la phase de l'envoi. Le même principe de modélisation est appliqué pour tous les constituants de l'AAR qui fonctionnent de la même manière.

Les places P_{12}, P_{21}, P_{11} avec les temporisations $\tau_{12}, \tau_{21}, \tau_{11}$ du CPU représentent respectivement les phases d'attente, de traitement programme, non disponibilité du CPU. La période de la CPU est représentée par τ_{11} .

Avant la fonction de traitement des données, la CPU lit toutes les nouvelles valeurs d'entrées des modules E/S qui étaient mises dans la mémoire tampon P_{19} par la carte de communication et à la fin du traitement des données, la CPU remet les résultats dans la mémoire tampon P_{32} . Les mémoires tampons sont les seuls liens existant entre la CPU et la carte de communication.

De manière similaire, τ_{33} représente la période de la carte de communication et la place P_{33} la non disponibilité du module pendant au moins τ_{33} . Le début d'envoi de la requête se fait avec le franchissement de la transition t_3 et se termine avec le franchissement de t_4 . Un jeton dans P_{65} indique que la requête est envoyée vers le module de sorti et la carte de communication se met en attente des réponses. Les places P_{54} et P_{98} modélisent respectivement les retards aller et retour induits par le réseau. Cette séparation est possible car les liaisons considérées sont full duplex et il n'y a pas de risque de collision ou de conflit. Pour éviter d'encombrer le modèle, τ_{98} inclut aussi le temps nécessaire pour la copie de la réponse du buffer d'entrée du coupleur vers la mémoire tampon modélisé par p_{19} . Les places P_{87} et P_{65} représentent respectivement le module d'entrée et le module de sortie. Il commence le traitement avec le franchissement de t_7 pendant un temps τ_{87} puis envoie le résultat à travers le réseau P_{98} . La transition de module de sortie t_5 est franchie après la réception de la réponse du CPU à travers le réseau et l'écoulement du temps de réponse τ_{58} . Les deux arcs discontinus modélisent l'interface du module d'E/S avec la partie opérative (informations en provenance des capteurs et les ordres donnés aux actionneurs). Elles ne sont pas prises en considération car le système à commander est non contraint. Tant que le capteur est fonctionnel, il y a toujours une valeur disponible à sa sortie.

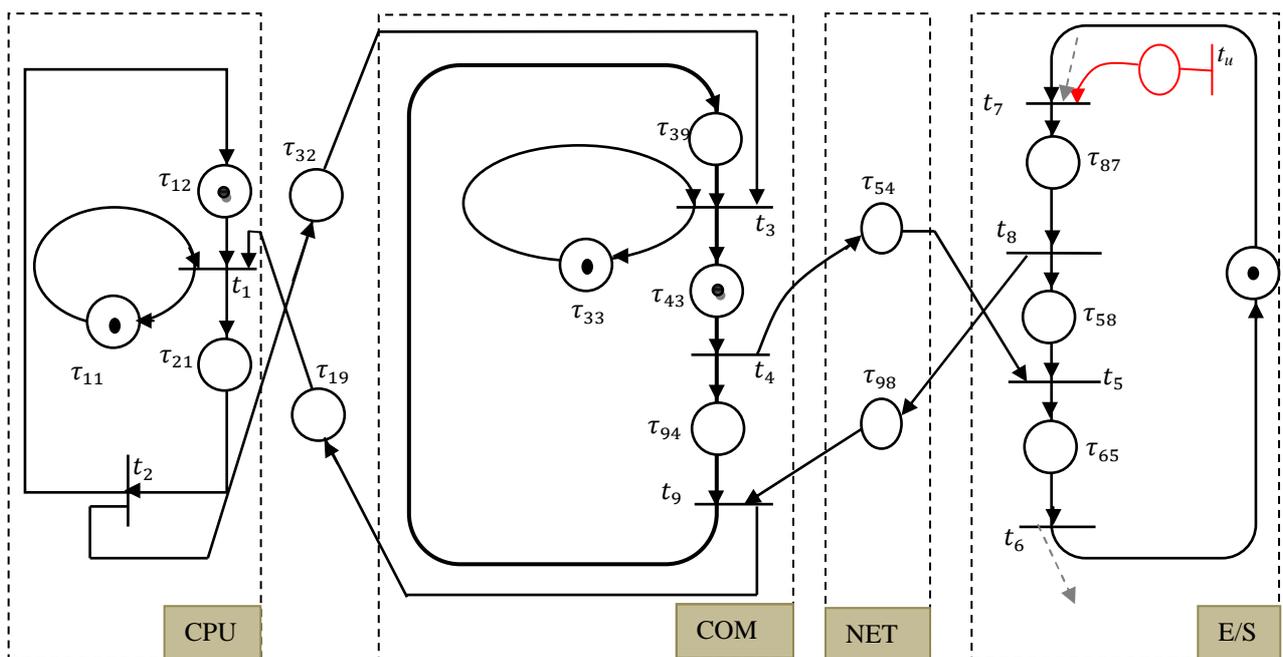


Figure 4 :Modèle GET de l'AAR (un PLC + un module E/S + le réseau)

3.3 Représentation de la dynamique de l'AAR

Nous considérons que le GET de l'AAR est temporisé comme suit : ($\tau_{12} = \tau_{32} = 0, \tau_{94} = \tau_{43} = \tau_{87} = \tau_{56} = \tau_{54} = \tau_{98} = 1, \tau_{11} = 5, \tau_{33} = 10, \tau_{58} = 12, \tau_{21} = 2$).

Nous associons à chaque transition du GET de la Fig.4 une fonction compteur. En tenant compte de la section 2.3 et en suivant la même démarche de l'exemple 2.4, nous montrons facilement que le comportement dynamique du l'AAR est représenté par l'équation Min-Plus linéaire suivante :

$$x(t) = A_0 \cdot x(t) \oplus A_1 x(t-1) \oplus A_2 \cdot x(t-2) \\ \oplus A_3 \cdot x(t-5) \oplus A_4 \cdot x(t-10) \\ \oplus A_5 \cdot x(t-12) \oplus B \cdot u(t)$$

avec :

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

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

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

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

$$A_4 = \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix},$$

$$A_5 = \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}, B = \begin{bmatrix} e \\ \varepsilon \end{bmatrix}.$$

D'après la section 2.3, 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 4 afin avoir un autre GET équivalent avec des temporisations binaires (1 ou 0). Toutes les places ayant des temporisations supérieures à 1 seront décomposées en plusieurs places temporisées à 1 et qui sont reliées par des transitions intermédiaires. Nous définissons de nouvelles variables d'état qu'on va additionner aux variables initiales. Les expressions des fonctions compteurs du graphe étendu de l'AAR sont données comme suit :

$$\begin{aligned} x_1(t) &= 1 \cdot x_2(t) \oplus e \cdot x_{14}(t) \oplus e \cdot x_9(t) \\ x_2(t) &= e \cdot x_{10}(t-1) \\ x_3(t) &= e \cdot x_2(t) \oplus e \cdot x_{34}(t-1) \oplus e \cdot x_9(t) \\ x_4(t) &= 1 \cdot x_3(t-1) \\ x_5(t) &= e \cdot x_{25}(t-1) \oplus e \cdot x_4(t-1) \\ x_6(t) &= e \cdot x_5(t-1) \\ x_7(t) &= 1x_6(t-1) \oplus e \cdot u(t) \\ x_8(t) &= e \cdot x_7(t-1) \\ x_9(t) &= e \cdot x_8(t-1) \oplus e \cdot x_4(t-1) \\ x_{10}(t) &= e \cdot x_1(t-1) \\ x_{11}(t) &= 1x_{11}(t-1) \\ x_l(t) &= e \cdot x_{(l-1)}(t-1), \text{ pour } l = 12 \text{ à } 14 \\ x_{15}(t) &= e \cdot x_8(t-1) \\ x_l(t) &= x_{(l-1)}(t-1), \text{ pour } l = 16 \text{ à } 25 \\ x_{26}(t) &= 1 \cdot x_3(t-1) \end{aligned} \tag{4}$$

$$x_l(t) = e \cdot x_{(l-1)}(t-1), \text{ pour } l = 27 \text{ à } 34$$

Ces 34 équations peuvent s'écrire sous forme matricielle :

$$x(t) = A \cdot x(t-1) \oplus B \cdot u(t) \quad (5)$$

Où $A \in \mathbb{R}_{min}^{34 \times 34}$ est la matrice d'état du graphe étendu.

4 COMMANDE SOUS CONTRAINTE TEMPORELLE DE L'AAR

4.1 Commande sous contraintes temporelles

Dans les graphes d'événements temporisés la temporisation de chaque place, correspond à la durée minimale de séjour des jetons dans cette place. Alors la durée maximale apparaît comme une contrainte supplémentaire qui devrait être vérifiée [Atto A. et al., 2011]. Plutôt qu'un problème de vérification, nous allons formuler la question comme un problème de commande qui consiste à calculer le paramètre $u(t)$ de façon à assurer que la contrainte temporelle est bien satisfaite. On considère la place p_{ij} soumise à une contrainte stricte [Amari S. et al., 2005a]. Un intervalle de temps $[\tau_{ij}^{min}, \tau_{ij}^{max}]$ est associé à cette place Figure 5.

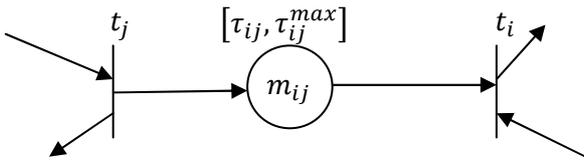


Figure 5 : Contrainte temporelle

Cette contrainte s'exprime dans \mathbb{R}_{min} par l'inéquation (6).

$$m_{ij} \cdot x_j(t - \tau_{ij}^{max}) \leq x_i(t) \quad (6)$$

C'est une contrainte supplémentaire à satisfaire. Pour résoudre ce problème, on détermine des lois de commande $u(t)$ qui garantissent le respect de cette contrainte. Ces lois de commande sont données sous forme d'un retour d'état (feedback) causal, exprimé comme suit :

$$u(t) = F \cdot x(t-1)$$

Une approche de commande des GET soumis à des contraintes temporelles strictes en utilisant l'algèbre Min-Plus est développée dans [Amari S. et al. 2005b]. Dans notre étude, on s'intéresse au cas d'une seule contrainte et d'une seule commande. Le théorème suivant donne des conditions suffisantes d'existence de telles lois.

Théorème 1 : Posons $(\varphi = \tau_{ij}^{max} + \tau_\alpha + 1)$, l'inéquation $u(t) \leq \bigoplus_{r=1}^N [(A^\varphi)_{ir} - (A^{\tau_\alpha})_{ju} - m_{ij}] \cdot x_r(t-1)$,

définit des commandes causales qui garantissent le respect de la contrainte (6), si les conditions suivantes sont satisfaites :

$$(i) (A^\varphi)_{ir} \geq (A^{\tau_\alpha})_{ju} \cdot m_{ij}, \quad \text{pour } r = 1 \text{ à } N$$

$$(ii) (A^k \cdot B)_i \geq (A^{\tau_\alpha})_{ju} \cdot m_{ij}, \quad \text{pour } k = 0 \text{ à } (\varphi - 1)$$

Corollaire 1 : Il existe toujours une loi de commande causale qui satisfait la contrainte (6) si le marquage initial de la place P_{ij} et le marquage des places du chemin α de u à t_j , sont nuls. Cette loi de commande est donnée par :

$$u(t) = \bigoplus_{r=1}^N (A^\varphi)_{ir} \cdot x_r(t-1)$$

Les démonstrations du théorème 1 et du corollaire 1 sont détaillées dans la référence [Amari S. et al., 2005b].

4.2 Commande de l'AAR

L'AAR comporte une contrainte temporelle maximale sur le temps de réponse (Dr) qui est le délai entre l'occurrence d'un événement sur l'entrée du module E/S et l'arrivée de sa conséquence issue du PLC sur le système commandé. Notre objectif est de calculer des lois de commande sous forme d'équations aux différences Min-Plus linéaires proposés dans (4.1). En considérant le graphe d'événement temporisé de la Figure 4 et les équations (4), la temporisation de la place P_{58} correspondant au temps de réponse du système vaut 12 unités de temps comme temps minimal et 15 unités de temps comme temps maximal. Cette contrainte temporelle maximale se traduit en Min-Plus par l'inéquation suivante :

$$x_5(t) \geq x_8(t-15)$$

Nous calculons la loi de commande $u(t)$ qui garantit le respect de la contrainte temporelle. Pour cela, nous appliquons le corollaire 1 sur l'équation d'état (5). Ayant $(\tau_{ij}^{max} = \tau_{58}^{max} = 15)$ et $\tau_\alpha = 1$, la temporisation de chemin $\alpha = (t_7, p_{87}, t_8)$ qui relie les transitions t_7 et t_8 , le marquage initial de la place $m_{58} = e$. 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 suivante :

$$u(t) = \bigoplus_{r=1}^{34} (A^{17})_{5r} \cdot x_r(t-1),$$

garantit la satisfaction de la contrainte temporelle.

$$\begin{aligned}
 u(t) = & 4x_1(t-1) \oplus 4x_3(t-1) \oplus 3x_5(t-1) \\
 & \oplus 4x_7(t-1) \oplus 2x_8(t-1) \\
 & \oplus 2x_{10}(t-1) \oplus 3x_{11}(t-1) \\
 & \oplus 2x_{12}(t-1) \oplus 4x_{13}(t-1) \\
 & \oplus 3x_{14}(t-1) \oplus 2x_{16}(t-1) \\
 & \oplus 3x_{18}(t-1) \oplus 4x_{19}(t-1) \\
 & \oplus 3x_{20}(t-1) \oplus 3x_{21}(t-1) \\
 & \oplus 4x_{22}(t-1) \oplus 1x_{23}(t-1) \\
 & \oplus 3x_{24}(t-1) \oplus 4x_{25}(t-1) \\
 & \oplus 3x_{26}(t-1) \oplus 3x_{27}(t-1) \\
 & \oplus 3x_{28}(t-1) \oplus 3x_{29}(t-1) \\
 & \oplus 2x_{30}(t-1) \oplus 4x_{31}(t-1) \\
 & \oplus 3x_{32}(t-1) \oplus 4x_{33}(t-1) \\
 & \oplus 2x_{34}(t-1).
 \end{aligned}$$

Après simplification, cette loi de commande se réduit à l'expression suivante :

$$\begin{aligned}
 u(t) = & 3x_5(t-1) \oplus 4x_7(t-1) \oplus 2x_1(t-1) \oplus 3x_1(t-4) \\
 & \oplus 4x_1(t-6) \oplus 1x_8(t-10) \\
 & \oplus 3x_8(t-11) \oplus 4x_8(t-12) \\
 & \oplus 3x_3(t-10)
 \end{aligned}$$

Grâce aux équations (4) on remplace par exemple, $1x_8(t-10)$ par $2x_6(t-12) \oplus u(t-11)$ et d'après les chemins du graphe on a : $x_6(t-12) \leq x_3(t-10) \Rightarrow 2x_6(t-12) \leq 3x_3(t-10)$.

De la même façon, en comparant les autres termes de cette commande, la loi de commande se simplifie à la somme Min-Plus des trois termes suivants :

$$u(t) = 3. x_5(t-1) \oplus 4. x_7(t-1) \oplus 1. x_8(t-10)$$

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 6). Ces places seront connectées au GET de l'AAR pour garantir le respect de la contrainte temporelle. Les transitions t_5 , t_7 et t_8 qui composent la commande sont observables

5 CONCLUSION

Dans ce papier, Nous avons modélisé une architecture d'automatisation distribuée en réseau (AAR) par un graphe d'événements temporisés. Nous avons appliqué une méthode formelle, développée dans l'algèbre Min-Plus, pour commander cette AAR sous une contrainte temporelle stricte. Les résultats obtenus sont des feedbacks causaux qui peuvent se représenter par trois places de contrôle qui seront connectées au modèle de l'AAR pour garantir la contrainte temporelle. Une première perspective de ce travail est la mise en œuvre des lois de commande trouvées sur une AAR réelle. Il reste à étendre la validité des résultats de cette étude et leur généralisation, en se penchant sur des architectures d'automatisation plus complexes. Il serait également intéressant d'utiliser d'autres approches de commande et faire une étude comparative.

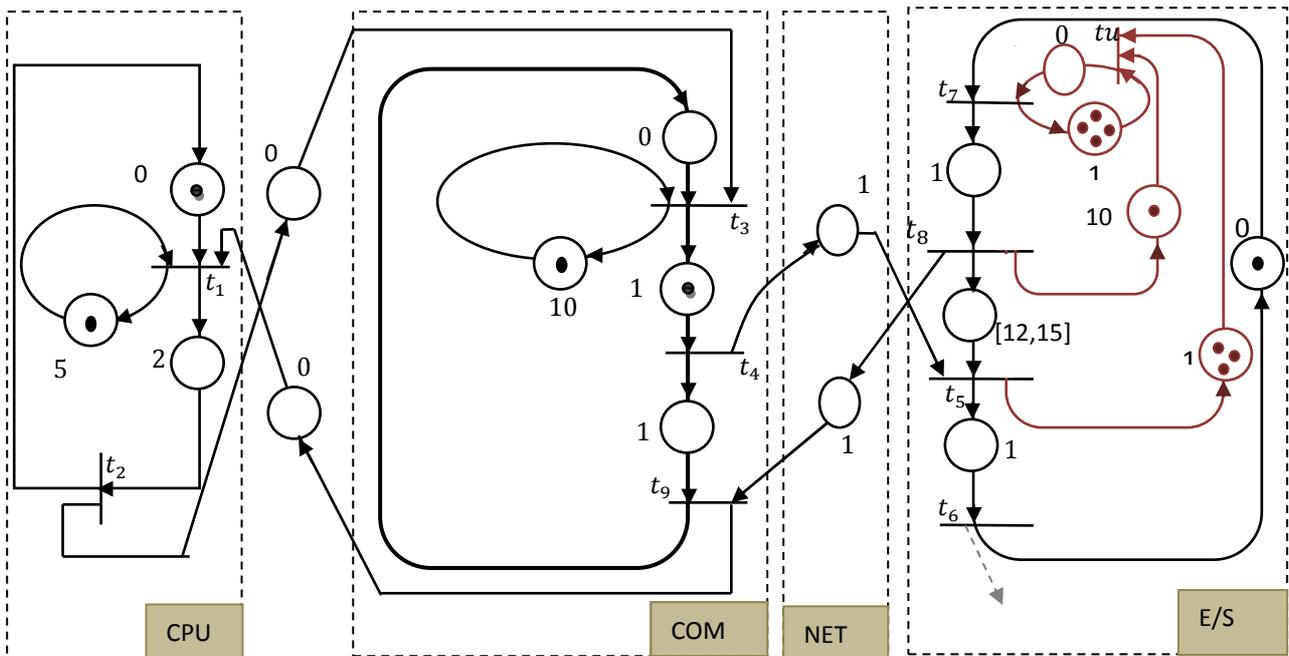


Figure 6 : Modèle GET de l'AAR avec les places de contrôle

REFERENCES

- Addad B., Amari S., 2008. Modeling and response time evaluation of ethernet-based control architectures using timed event graphs and Max-plus algebra. *4th IEEE CASE, Washington DC*,
- Addad B., Amari S. 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*.
- Amari S., I. Demongodin and J-J. Loiseau, 2004. Sizing, cycle time and plant control using dioid algebra. *Chapter 6 in Supply Chain Optimization, Series Applied Optimization, A. Dolgui, J. Soldek and O. Zaikin (Eds), Springer*, pp. 71-85.
- 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*.
- Amari S., I. Demongodin et J.J. Loiseau, 2005b. Méthode formelle de commande sous contraintes de temps dans les dioides. *MSR'05, Grenoble*.
- Atto A. M., C. Martinez and S. Amari, 2011. Control of Discrete Event Systems with Respect to Strict Duration: Supervision of an Industrial Manufacturing Plant. *Computers & Industrial Engineering, Volume 61, Issue 4, November, Pages 1149-1159*.
- Baccelli F., G. Cohen, G. Olsder, and J. Quadrat, 1992a. *Synchronization and Linearity*. Wiley.
- Baccelli F., G. Cohen and B. Gaujal, 1992b. *Recursive equations and basic properties of timed Petri nets. Discrete Event Dynamic Systems: Theory and Applications, l(4)*.
- 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*.
- Cottenceau B., L. Hardouin, J.L. Boimond, and J.L. Ferrier, 1999. Synthesis of Greatest linear feedback for timed event graphs in dioid. *IEEE Transactions on Automatic Control, Vol. 44(6), p.p. 1258-1262*.
- 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*.
- Diouri I., 2010. *Propositions de méthodes pour adapter le réseau aux contraintes d'applications temps-réel*. Thèse de doctorat, Université Henri Poincaré, Nancy 1.
- Gianluca C., I. Cibrario Bertolotti, and A. Valenzano, 2006. Experimental analysis of latencies in Etherne. In *Proc. of WFCS'06*.
- 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*
- 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*.
- Kim J-H., Tae-Eog Lee, Hwan-Yong Lee and Doo-Byeong Park, 2003. Scheduling Analysis of Time-Constrained Dual-Armed Cluster Tools, *IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 3, pp. 521-534*.
- Lahaye S., B. Cottenceau, A. Correia, 2004. Commande de graphe d'événements temporisés avec contraintes de temps critique. *CIFA, Tunisie, 2004*.
- Lee K. C. and S. Lee, 2002. Performance evaluation of switched Ethernet for real-time industrial communications, *Computer standards & interfaces, (24):411–423*.
- Neumann P., 2007. Communication in industrial automation-what is going on? *Control Engineering Practice, 15, pp. 1332-1347*.
- Ruel S., 2009. *Evaluation des bornes des performances temporelles des Architectures d'Automatisation en Réseau par preuves itératives de propriétés logiques*. Thèse de doctorat en EEA, ENS de Cachan.
- Van den Boom T. and B. De Schutter, 2004. Modeling and Control of railway networks. *Proc. of American Control Conference, Boston, pp. 5728-5733*.