



**HAL**  
open science

## Contribution à la commande des systèmes à événements discrets temporisés (SEDTS)

Imane Tahiri, A Tajer, Y Qamsane, A. Philippot

► **To cite this version:**

Imane Tahiri, A Tajer, Y Qamsane, A. Philippot. Contribution à la commande des systèmes à événements discrets temporisés (SEDTS). Conférence Francophone d'Optimisation et Simulation (MOSIM), 2016, Montréal, Canada. hal-02114230

**HAL Id: hal-02114230**

**<https://hal.science/hal-02114230>**

Submitted on 29 Apr 2019

**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.

## **CONTRIBUTION A LA COMMANDE DES SYSTEMES A EVENEMENTS DISCRETS TEMPORISES (SEDTS)**

**I. TAHIRI, A. TAJER, Y. QAMSANE**

Laboratoire de génie électrique et commande des systèmes.  
Université Cadi Ayyad-ENSA Marrakech,  
MAROC  
imane.tahiri68@gmail.com, a.tajer@uca.ma,  
qamsaneyassine@gmail.com

**A. PHILIPPOT**

Centre de Recherche en Science et Technologie de l'Information et de la Communication (CReSTIC)  
Université de Reims Champagne Ardenne Reims,  
France  
alexandre.philippot@univ-reims.fr

**RESUME:** *Ce papier présente une approche de synthèse de commande sûre de fonctionnement des Systèmes à Evénements Discrets Temporisés (SEDTS), à base de propriétés temporelles. Pour établir cette synthèse, la démarche proposée s'appuie sur la modularité des systèmes manufacturiers retrouvée dans les travaux (Qamsane et al., 2014a,b). Un exemple de système automatisé de production vient illustrer nos propos.*

**MOTS-CLES:** *Systèmes à événements discrets temporisés, théorie de la supervision, commande distribuée, systèmes manufacturiers, modélisation.*

### **1 INTRODUCTION**

Un Système à Événement Discret (SED) se définit comme un système dynamique pouvant se décrire dans un espace d'état discret et dont l'évolution est décrite par des transitions d'états déclenchées par des événements (Cassandras et Lafortune, 2008). Un des problèmes récurrent à ces systèmes est l'établissement d'une loi de commande adéquate garantissant la sécurité du matériel et du personnel tout en effectuant les tâches demandées. Dans (Ramage et Wonham, 1989), les auteurs ont proposé une théorie de supervision utilisant les automates à états finis et les langages pour modéliser ces systèmes. Bien que cette approche soit très formelle, elle manifeste un inconvénient majeur de son utilisation sur les systèmes complexes: l'explosion combinatoire. Dans les travaux (Leduc et al., 2009), (Cai et Wonham, 2010), (Tajer et al., 2011), (Khousi et Chakib, 2014) et (Qamsane et al., 2014 a,b), des approches pour la synthèse de la commande des SEDs ont été proposées pour lever ce problème. Bien que ces approches aient pu diminuer le problème d'explosion combinatoire, elles n'en restent pas moins dépendantes de l'information booléenne. Or, elle n'est pas la seule information présente lorsque l'on y considère le temps. On parle alors des Systèmes à Evénements Discrets Temporisés (SEDTS).

Plusieurs approches de la littérature utilisent les automates temporisés, les réseaux de Petri ou les automates à états finis comme outils de modélisation des SEDTS: (Ostroff, 1990) (Brandin et Wonham, 1994), (Alur et Dill, 1994), (Sava et Alla, 2006), (Khousi et Ouedraogo, 2010), et (Zhang et al., 2013). Dans ces travaux le temps est modélisé par deux façons:

- Le temps discret, pour lequel la progression du temps est discrète et mesurée par une horloge digitale qui s'incrémente d'une unité à la fois.

- le temps dense, pour lequel la progression du temps est continue et mesurée par des horloges prenant des valeurs réelles positives.

Cependant, la prise en compte du passage du temps rend l'étape de modélisation ainsi que la synthèse de la commande par supervision plus complexes. Pour pallier ces difficultés, nous préconisons une extension de notre démarche de synthèse de la commande en exploitant sa conception distribuée (Qamsane et al., 2014 a,b) et en y introduisant le facteur temps dès les premières étapes de modélisation. L'approche proposée est constituée des étapes suivantes (Figure 1):

1. Modélisation modulaire et temporisée de la partie opérative (PO).
2. Définition des contraintes locales temporelles et non-temporelles de sécurité et de vivacité à partir des spécifications, et application locale de l'algorithme de synthèse (Tajer et al., 2013).
3. Première agrégation locale non temporelle avec mise en avant des ordres autorisés et interdits (Qamsane et al., 2014 a,b).
4. Deuxième agrégation locale temporelle avec mise en avant des activations et désactivations des horloges.
5. Prise en compte des contraintes globales de sécurité et de vivacité pour déterminer des contrôleurs distribués.

6. Traduction des contrôleurs distribués en langage normalisé IEC 61131-3.
7. Implémentation dans un Automate Programmable Industriel (API).

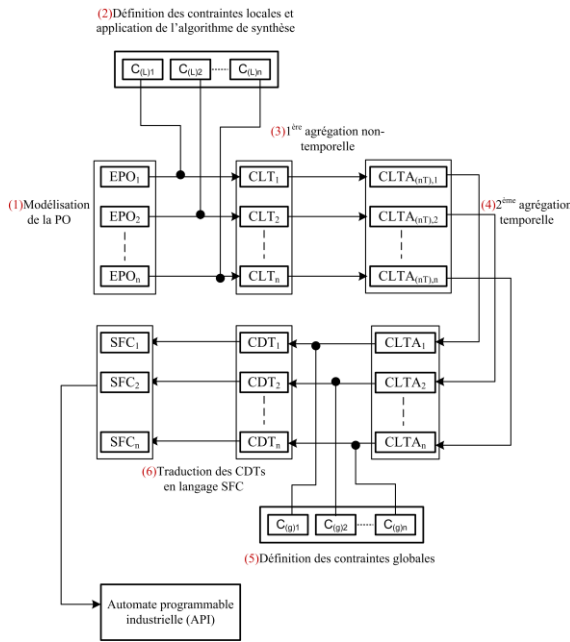


Figure 1: Structure de l'approche distribuée temporisée

Nous présentons dans ce papier les résultats de nos travaux en décrivant chaque étape de l'approche en section 2. La section 3 illustre nos propos sur un benchmark académique. Enfin, nous concluons et présentons nos perspectives de travaux en section 4.

## 2 CONCEPT DE BASE DE L'APPROCHE DISTRIBUEE DES SEDTS

### 2.1 Modélisation de la partie opérative (PO)

L'approche se base sur un modèle de la Partie Opérative (PO) décrivant le plus respectueusement possible le comportement du système réel. Cette modélisation des Eléments de la PO (EPO) est issue d'une extension du modèle pratique de la PO présenté dans (Philippot, 2006) par l'introduction du temps. Un EPO étant obtenu par composition synchrone de modèles de détecteurs, de préactionneurs et d'actionneurs. Chaque EPO ne dispose pas toujours d'un ensemble de détecteurs suffisant pour être décrit. Des modèles des détecteurs temporisés noté  $G_{DT}$ , sont donc introduits. Il convient alors d'intégrer une horloge dans le modèle des actionneurs  $G_{AT}$ , où l'occurrence des événements d'activations des horloges est considérée comme un événement non contrôlable. Un exemple de cette modélisation sera présenté dans la section 3.2.

Le modèle pratique de l'EPO<sub>i</sub> est obtenu en effectuant la composition synchrone entre le modèle des détecteurs temporisés et celui de l'actionneur temporisé ( $G_T = G_{DT} \parallel$

$G_{AT}$ ). Le modèle de l'EPO est présenté par un automate:  $G_T = (Q, \Sigma, H, T, \delta, q_0, Q_m)$  avec:

- **Q**: Ensemble fini des états
- **$\Sigma$** : Ensemble fini des événements tel que:  $\Sigma = \Sigma_c \cup \Sigma_{uc}$ . Avec:  
 $\Sigma_c = \uparrow\downarrow Z \cup \downarrow H$  et  $\Sigma_{uc} = \uparrow\downarrow E \cup \uparrow H$ .  
 $\uparrow\downarrow Z$ : Ensemble des activations et des désactivations des ordres de la commande  $z_i$  (sorties actionneurs).  
 $\uparrow\downarrow E$ : Ensemble des fronts montants et descendants des variables d'entrées  $e_i$  (entrées capteurs).  
 $\uparrow\downarrow H$ : Ensemble des activations et désactivations des horloges  $h_i$ . Les activations sont définies non-contrôlables puisque on ne connaît pas au préalable le moment de déclenchement des horloges. Par contre les désactivations sont considérées contrôlables car nous maîtrisons la durée d'activation des horloges qui précède la désactivation.  
Avec:  $\Sigma_{nT} = \uparrow\downarrow Z \cup \uparrow\downarrow E$ , ensemble des événements non-temporisés, et  $\Sigma_T = \uparrow\downarrow H$ , ensemble des événements temporisés.
- **H**: Ensemble fini des horloges, tel que:  $H = \uparrow h_i \cup \downarrow h_i$ .
- **T**: Ensemble fini des durées  $t_i$  associées à chaque horloge  $h_i$ , tel que  $T = \{t_1, t_2, \dots, t_n\}$ .
- **$\delta$** : Fonction de transition. Une transition est définie par:  $\delta(q_1, \sigma) = q_2$  tel que  $q_1$  est l'état de départ et  $q_2$  est l'état d'arrivée.  $\sigma$  est l'occurrence d'un événement temporisé ou non de  $\Sigma$ .
- **q<sub>0</sub>**: Etat initial.
- **Q<sub>m</sub>**: Ensemble des états marqués.

### 2.2 Synthèse et agrégation des contrôleurs locaux temporisés

La synthèse des Contrôleurs Locaux Temporisés (CLTs) est basée sur l'utilisation de l'algorithme de synthèse (Tajer et al, 2013), dont les contraintes du fonctionnement locale sont représentées par des équations logiques. Cependant, nous y introduisons des contraintes temporelles soit sur l'activation/désactivation de l'horloge, soit sur sa durée. Et pour but d'appliquer les contraintes globales de sécurité et de vivacité aux contrôleurs locaux temporisés, une démarche d'agrégation des CLTs<sub>i</sub> est réalisée en deux étapes:

**Etape 1:** Associer les états atteints par les événements contrôlables ( $z_i \in \Sigma_c$ ) dans des macro-états liés par les événements incontrôlables ( $e_i \in \Sigma_{uc}$ ), les événements temporisés  $\{\uparrow h_i \in \Sigma_{nc}, \downarrow h_i \in \Sigma_c\}$  et les transitions tem-

porisées associées à  $t_i$ . Une transition temporisée entre deux états est conditionnée par une garde ( $t = t_i$ ) sur les horloges  $h_i$ . Si l'état du  $CLT_i$  est associé à un front montant d'un événement contrôlable, alors l'ordre est autorisé et appartient à l'ensemble  $Ord$ . Par contre s'il est associé à un front descendant de cet événement alors l'ordre est inhibé et appartient à l'ensemble  $Inh$ .

L'algorithme formel d'obtention de la première agrégation  $CLTA_{i(nT)}$  est défini comme suit: (figure 2).

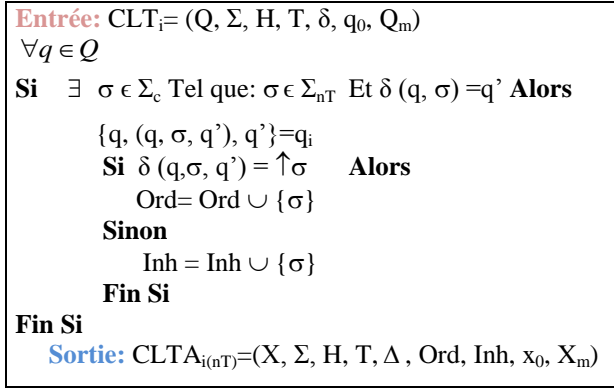


Figure 2: Algorithme d'obtention de la 1<sup>ère</sup> agrégation non-temporelle

Un exemple d'automate constitué de 6 états est présenté dans la figure 2. On distingue 2 ensembles:  $\Sigma_{nT} = \{\uparrow e_1, \downarrow e_1, \uparrow z_1, \downarrow z_1\}$  et  $\Sigma_T = \{\uparrow h_1, \downarrow h_1\}$ . Par application de la première agrégation on obtient le deuxième automate à 4 états et 5 transitions.

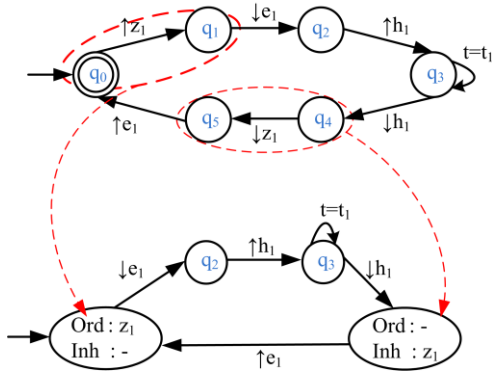


Figure 3: Exemple d'un  $CLTA_{i(nT)}$

**Etape 2:** Agréger les événements temporisés  $\uparrow h_i$  dans des macro-états liés par les événements incontrôlables ( $e_i \in \Sigma_{uc}$ ) et les transitions temporisées conditionnées par ( $t = t_i$ ). Si l'état du  $CLTA_{i(nT)}$  est atteint par un événement qui correspond à une activation de l'horloge, alors ce dernier appartient un ensemble noté  $A_{clock}$ . S'il est atteint par un événement correspondant à la désactivation de l'horloge, cet événement sera remplacé par la condition ( $t = t_i$ ) qui a été déjà exprimé par une self-loop en supprimant cette dernière. L'automate résultant est le contrôleur local agrégé temporisé noté  $CLTA_i$  (figure 4).

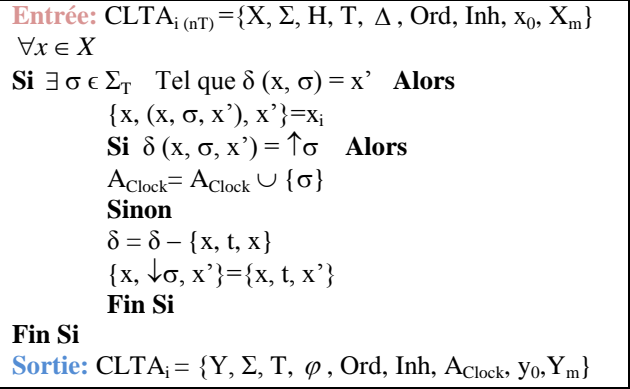


Figure 4: Algorithme d'obtention de la 2<sup>ème</sup> agrégation temporelle

La deuxième agrégation appliquée depuis le modèle de  $CLTA_{i(nT)}$  de la figure 3 permet d'obtenir le modèle  $CLTA_i$  de la figure 5.

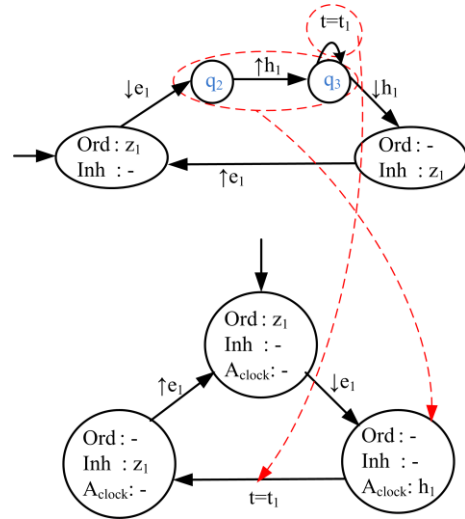


Figure 5: Exemple d'un  $CLTA_i$

### 2.3 Etablissement du contrôleur distribué temporisé CDT

Pour établir les contrôleurs distribués temporisés  $CDT_i$ , il faut que les contraintes liées au fonctionnement globale du système manufacturier soient prises en compte. Un contrôleur distribué temporisé est défini par l'automate suivant :  $CDT = (S_T, \Sigma, T, \delta_T, E, C, s_0)$ , avec:

- **S<sub>T</sub>**: Ensemble fini des états. A chaque état est associé des éléments des ensembles E et C.
- **Σ**: Ensemble fini des événements.
- **δ<sub>T</sub>**: Fonction de transition, une transition peut être associée à un événement  $\sigma$  qui peut être un événement non temporisé et non contrôlable  $e_i$  ou à une condition sur l'horloge ( $t = t_i$ ).
- **E**: Ensemble  $\{Ord, Inh, A_{clock}\}$  associé à chaque état de  $S_T$ .

- **C**: Ensemble fini des contraintes globales liées au fonctionnement du système.
- **s<sub>0</sub>**: Etat initial.

La figure 6 présente un exemple d'un CDT. Le passage de l'état initial à l'état suivant n'est possible que si la contrainte sur le fonctionnement global de l'EPO (dcy = 1) ou la contrainte sur le fonctionnement local (c<sub>0</sub> = 1) et la transition liée à l'événement  $\downarrow e_1$  sont vérifiées.

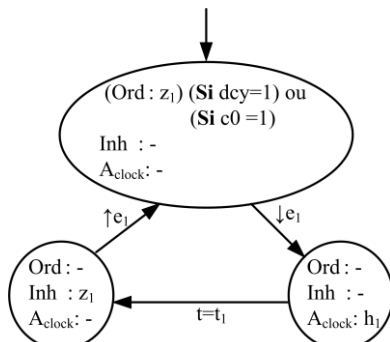


Figure 6: Exemple d'un CDT

### 2.4 Traduction des CDTs en langage SFC

Les Contrôleurs Distribués Temporisés sont transformés en langage SFC. Le Sequential Function Chart (SFC) est un langage de programmation des APIs normalisé (IEC 61131-3). Il est notamment très proche du GRAFCET (IEC 60848) dans sa compréhension.

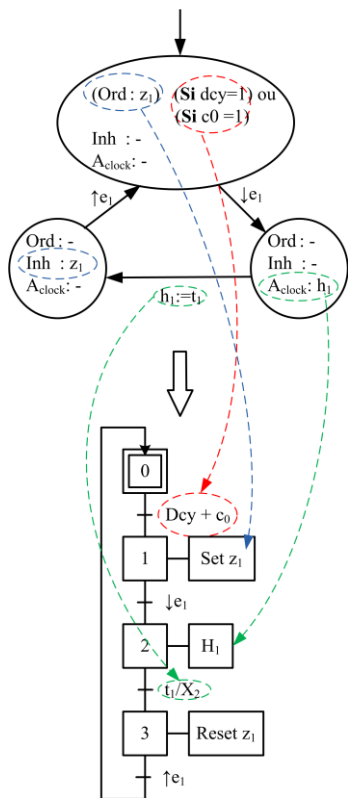


Figure 7: Transformation d'un CDT au SFC

Cette transformation consiste à commencer par l'état initial du CDT qui sera traduit par une étape initiale au niveau du SFC. Ensuite, l'envoi des ordres (Ord: z<sub>i</sub>) est traduit par des actions maintenues Set (mise à 1), l'inhibition de ces ordres (Inh: z<sub>i</sub>) est traduite par une action Reset (mise à 0) de l'action. Les contraintes globales ou locales qui conditionnent ou non l'autorisation de l'ordre sont transformées en réceptivité de la transition juste avant l'étape associée à l'action correspondante. L'activation de l'horloge (A<sub>clock</sub>: h<sub>i</sub>) est traduite par une temporisation H<sub>i</sub>, et la durée t<sub>i</sub> sur l'action est transformée en durée par rapport à l'étape antérieure du SFC qui est associée à l'activation de h<sub>i</sub>. Un exemple de cette transformation est illustré dans la figure 7.

## 3 ILLUSTRATION

### 3.1 Description du système

Pour illustrer notre contribution, nous considérons un cas d'étude portant sur un système de production vu comme un réseau d'interactions de composants à événements discrets temporisés (figure 8). C'est un système de transfert de pièces constitué de:

- Deux poussoirs sous forme de vérins simples effets, le premier poussant les pièces devant le second, et ce dernier les transférant vers le convoyeur d'évacuation.
- Deux convoyeurs à bandes, afin de transporter les pièces devant le premier poussoir, et pour les évacuer vers le stock. Le convoyeur d'évacuation est considéré toujours en marche.
- Cinq capteurs: c<sub>0</sub> et c<sub>3</sub> capteurs de présence des pièces devant les poussoirs, c<sub>1</sub> et c<sub>2</sub> capteurs associés aux poussoirs et correspondent aux fins de course des vérins en position rentrée, et c<sub>4</sub> capteur qui détecte le passage des pièces sur le convoyeur d'évacuation.
- Un bouton poussoir de mise en marche: Bp<sub>Dcy</sub>.

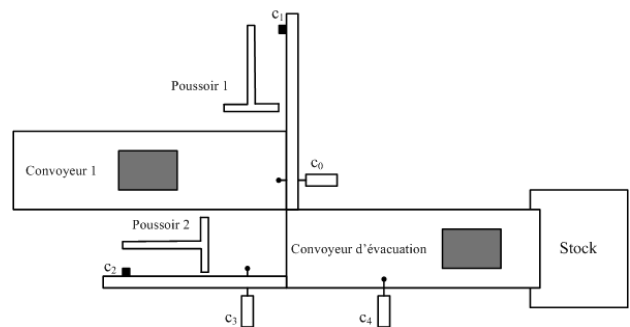


Figure 8: Système de transfert des pièces

### 3.2 Modélisation des éléments de la partie opérative (EPOs)

Les EPOs du système étudié sont obtenus comme suit:

**EPO<sub>1</sub>:** Convoyeur 1 associé au capteur  $c_0$  placé à l'extrémité droite du convoyeur. L'activation du convoyeur est assurée par l'ordre  $Conv_1$ . On distingue deux ensembles d'événements non temporisés:  $\Sigma_c = \{\uparrow Conv_1, \downarrow Conv_1\}$  et  $\Sigma_{uc} = \{\uparrow c_0, \downarrow c_0\}$ .

Le composé synchrone du modèle actionneur avec le modèle des détecteurs génère l'automate présentant toutes les situations possibles de cet élément de la partie opérative (figure 9).

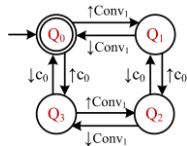


Figure 9: Modèle de l'EPO<sub>1</sub> du convoyeur 1

Le fonctionnement souhaité de cet EPO est décrit par la séquence suivante: A l'état initial si l'ordre  $conv_1$  est envoyé, le convoyeur 1 sera à l'état marche jusqu'à l'activation du capteur  $c_0$ . Une fois la pièce est transférée du convoyeur, le capteur sera désactivé et on se retrouve à l'état initial.

**EPO<sub>2</sub>:** Poussoir 1 associé au capteur  $c_1$ . La sortie de la tige est assurée par l'ordre  $P_1$  pendant une durée  $t_1$  de l'horloge  $h_1$ . On distingue deux ensembles d'événements temporisés et non temporisés:  $\Sigma_{nT} = \{\uparrow P_1, \downarrow P_1, \uparrow c_1, \downarrow c_1\}$  et  $\Sigma_T = \{\uparrow h_1, \downarrow h_1\}$ , avec  $\Sigma_c = \{\uparrow P_1, \downarrow P_1, \downarrow h_1\}$  et  $\Sigma_{uc} = \{\uparrow h_1, \uparrow c_1, \downarrow c_1\}$ . Le composé synchrone du modèle actionneur temporisé figure 10-a) avec le modèle des détecteurs temporisé (figure 10-b) génère l'automate présentant toutes les situations possibles de cet élément de la partie opérative (Figure 10-c).

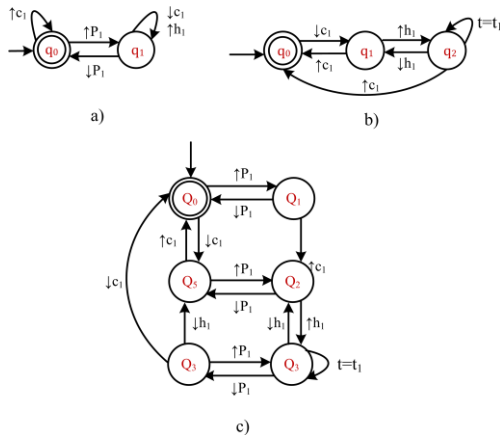


Figure 10: Modèle de l'EPO<sub>2</sub> du poussoir 1

L'une des séquences possible par cet EPO est la suivante: Si l'ordre de sortie de la tige du vérin 1 est envoyé, le capteur de fin de course  $c_1$  sera désactivé. En-

suite l'horloge  $h_1$  sera activée pendant  $t_1$ . Une fois cette dernière est écoulée, l'horloge se désactive et entraîne la désactivation de la sortie de la tige. Le retour de cette tige est assuré par le rappel du ressort jusqu'à l'activation du capteur  $c_1$ , le vérin revient alors à son état initial.

**EPO<sub>3</sub>:** Poussoir 2 associé au capteur  $c_2$ . La sortie de la tige est assurée par l'ordre  $P_2$  pendant une durée  $t_2$  de l'horloge  $h_2$ . Deux ensembles d'événements temporisés et non temporisés sont distingués:  $\Sigma_{nT} = \{\uparrow P_2, \downarrow P_2, \uparrow c_2, \downarrow c_2\}$  et  $\Sigma_T = \{\uparrow h_2, \downarrow h_2\}$ , avec  $\Sigma_c = \{\uparrow P_2, \downarrow P_2, \downarrow h_2\}$  et  $\Sigma_{uc} = \{\uparrow h_2, \uparrow c_2, \downarrow c_2\}$ . Le composé synchrone du modèle actionneur temporisé avec le modèle des détecteurs temporisé génère le même modèle que celui de la figure 10, en changeant  $P_1$  par  $P_2$ ,  $c_1$  par  $c_2$ ,  $h_1$  par  $h_2$  et  $t_1$  par  $t_2$ .

### 3.3 Définition des contraintes locales et application locale de l'algorithme de synthèse

Les contraintes locales temporelles et non temporelles de vivacité à appliquer à chaque EPO sont présentées dans le tableau 1. Par exemple la contrainte de vivacité temporelle « $t_1$  AND  $\downarrow h_1 = 1$ » traduit le fait que si la durée  $t_1$  est écoulée, l'horloge  $h_1$  doit être désactivée.

Type	EPO	id	Contraintes
Vivacité	1	(1)	$\uparrow Conv_1$ AND $\uparrow c_0 = 1$
		(2)	$\downarrow Conv_1$ AND $\downarrow c_0 = 1$
	2	(3)	$t_1$ AND $\downarrow h_1 = 1$
		(4)	$\uparrow P_1$ AND $\downarrow c_1 = 1$
		(5)	$\downarrow P_1$ AND $\uparrow c_1 = 1$
	3	(6)	$t_2$ AND $\downarrow h_2 = 1$
		(7)	$\uparrow P_2$ AND $\downarrow c_2 = 1$
		(8)	$\downarrow P_2$ AND $\uparrow c_2 = 1$

Tableau 1: Contraintes locales de vivacité

Pour obtenir les contrôleurs locaux temporisés CLT<sub>i</sub> (figure 11), les contraintes locales définies dans le tableau ci-dessus doivent être appliquées sur les modèles des EPOs.

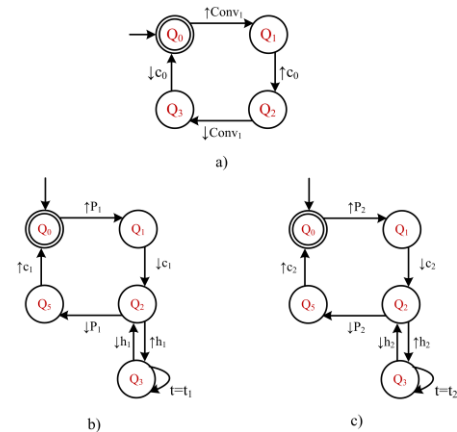


Figure 11: Contrôleurs locaux (a. CL<sub>Conv1</sub>), (b. CLT<sub>P1</sub>) et (c. CLT<sub>P2</sub>)



### 3.4 Agrégation des contrôleurs locaux temporisés et non temporisés

#### 3.4.1 Agrégation du contrôleur local $CL_{Conv1}$

Dans cette étape, l'envoi de l'ordre  $Conv_1$  à partir de l'état initial  $Q_0$ , est autorisé ( $\uparrow Conv_1$ ). Par conséquent les deux états  $Q_0$  et  $Q_1$  sont agrégés en un seul état défini par (Ord:  $Conv_1$ ). De même, les deux états  $Q_2$  et  $Q_3$  sont agrégés en un seul état caractérisé par l'inhibition de l'ordre  $Conv_1$  (Inh:  $Conv_1$ ).

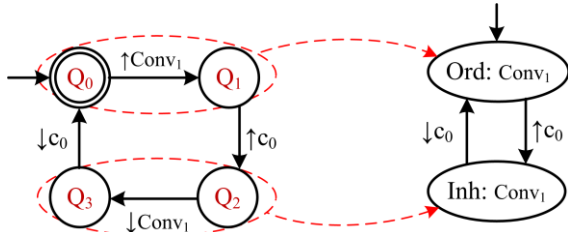


Figure 12: Contrôleur local agrégé  $CLA_{Conv1}$  du convoyeur 1

#### 3.4.2 Agrégation du contrôleur local temporisé $CLT_{P1}$

L'agrégation du contrôleur local décrivant le fonctionnement souhaité du poussoir 1 se fait selon 2 étapes. La première est l'agrégation des événements non temporisés contrôlables  $\uparrow P_1$  et  $\downarrow P_1$  dans des macro-états pour obtenir un  $CLTA_{P1(nT)}$  (figure 13). La deuxième consiste à agréger l'événement temporisé  $\uparrow h_1$  dans un macro-état pour obtenir un  $CLTA_{P1}$  (figure 14).

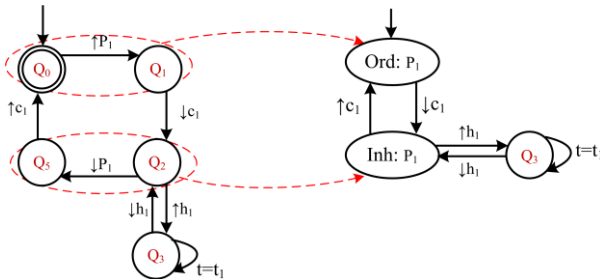


Figure 13: Première agrégation du  $CLT_{P1}$

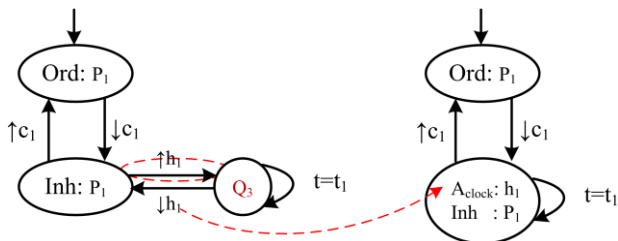


Figure 14:  $CLTA_{P1}$  du poussoir 1

#### 3.4.3 Agrégation du contrôleur local $CL_{P2}$

En suivant les mêmes démarches d'agrégation du contrôleur local temporisé agrégé du poussoir 1, nous obtenons celui du deuxième poussoir (figure 15).

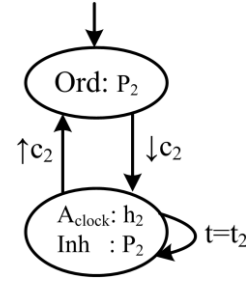


Figure 15:  $CLTA_{P2}$  du poussoir 2

### 3.5 Détermination des contrôleurs distribués temporisés et non temporisés

Pour pouvoir obtenir les contrôleurs distribués correspondants à chaque EPO, les contraintes globales doivent être prises en compte (tableau 2).

id	Contraintes
1	<b>Si</b> ( $B_{p_{Dcy}}=1$ ET $c_1=1$ ET $c_2=1$ ) OU ( $c_4=1$ ) <b>Alors</b> $Conv_1=1$
2	<b>Si</b> ( $c_0=1$ ) ET ( $c_2=1$ ) <b>Alors</b> $P_1=1$
3	<b>Si</b> ( $c_3=1$ ) ET ( $c_1=1$ ) <b>Alors</b> $P_2=1$

Tableau 2: Contraintes globales de vivacité et de sécurité

Par exemple la contrainte «1» permet l'autorisation de l'ordre  $Conv_1$  (Ord:  $Conv_1$ ) à la condition d'avoir ( $B_{p_{Dcy}}=1$  et  $c_1=1$  et  $c_2=1$ ) ou ( $c_4=1$ ). Le contrôleur distribué résultant est présenté en figure 16.

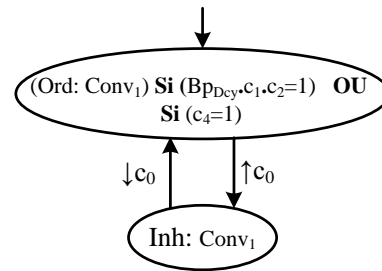


Figure 16: Contrôleur distribué  $CD_{Conv1}$  du convoyeur 1

En suivant la même démarche d'affectation des contraintes globales à chaque autorisation d'ordre, nous obtenons les deux autres contrôleurs distribués temporisés (figure 17).

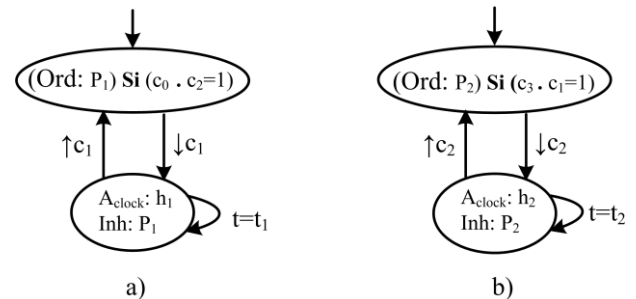


Figure 17: Contrôleurs distribués temporisés des deux poussoirs (a.  $CDT_{P1}$ ) et (b.  $CDT_{P2}$ )

### 3.6 Traduction des CDs et CDTs en langage SFC

Pour traduire le  $CD_{Conv1}$  en langage SFC, l'état initial du contrôleur distribué  $CD_{Conv1}$  est traduit par une étape initiale au niveau du SFC (figure 18). Les conditions sur cet état exprimant le comportement global sont transformées en réceptivité de la transition vers l'étape 1. L'envoi de l'ordre  $Conv_1$  est traduit par une action (set  $Conv_1$ ) associée à l'étape 1 du SFC. L'inhibition de cet ordre est traduite par une action (reset  $Conv_1$ ) associée à l'étape 2.

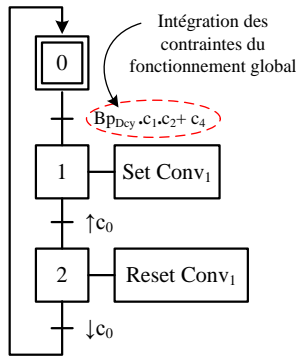


Figure 18: SFC du contrôleur distribué  $CD_{Conv1}$  du convoyeur 1

Pour traduire le  $CDT_{P1}$  en langage SFC, son état initial est traduit par une étape initiale (étape 10) au niveau du SFC. Les contraintes sur cet état exprimant le comportement global sont transformées en réceptivité de la transition vers l'étape 10 du SFC. Cette dernière est associée à une action «Set  $P_1$ ». L'activation de l'horloge « $h_1$ » est traduite par une action temporisée associée à l'étape 12. La transition temporisée exprimée par une self-loop au niveau du  $CDT_{P1}$  est traduite par une transition de l'étape 12 vers l'étape 13. L'inhibition de l'ordre  $P_1$  dû à l'écoulement de  $t_1$  est traduite par une action mise à zéro de  $P_1$  «Reset  $P_1$ » associée à l'étape 13 du SFC (figure 19).

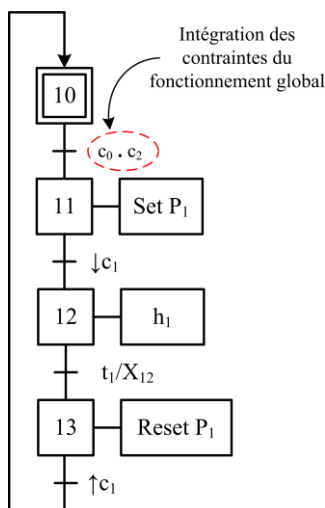


Figure 19: SFC du contrôleur distribué temporisé  $CDT_{P1}$  du poussoir 1

De la même façon, nous obtenons le dernier SFC correspondant au poussoir 2.

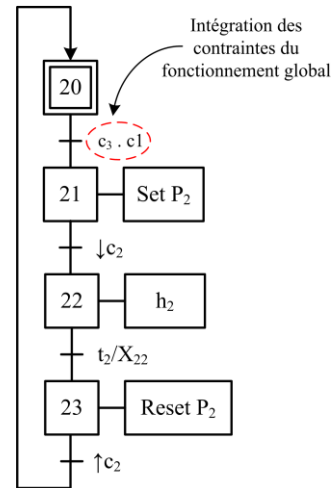


Figure 20: SFC du contrôleur distribué temporisé  $CDT_{P2}$  du poussoir 2

Les SFCs résultants de tous les contrôleurs distribués non-temporisés et temporisés sont implémentés simultanément dans un API.

## 4 CONCLUSION ET PERSPECTIVES

Dans ce travail nous avons proposé une contribution à la synthèse de la commande des SEDs en adoptant une approche distribuée et en intégrant le facteur temps. Par conséquent nous avons réussi à élaborer une démarche structurée de synthèse allant d'une modélisation modulaire de la partie opérative prenant en compte le temps réel associé à chaque action du système manufacturier étudié, jusqu'à l'implémentation des différents contrôleurs distribués en langage SFC pour exécution.

Nos travaux futurs consistent à généraliser les résultats établis pour certaines classes de systèmes complexes et non seulement aux systèmes manufacturiers et les exploiter pour vérifier l'information sur les éléments d'un processus quelconque en cas de défaut. Par exemple s'il est programmé de détecter une caisse devant un capteur lors de l'écoulement d'une durée «  $t$  » et que ce dernier n'est pas activé, cela peut signifier qu'un défaut au niveau de ce capteur est présent. Il est également envisagé d'associer cette information temporelle comme une information redondante permettant ainsi de définir un algorithme de reconfiguration des systèmes complexes.

## REFERENCES

- Alur, R and D. L. Dill, 1994. A theory of timed automata, *Theoretical Computer Science* 126 (1994): 183–235.
- Brandin, A and M. Wonham, 1994. Supervisory control of timed discrete-event systems, *IEEE Transactions on Automatic Control* 39(2) (1994): 329–342.
- Cai, K, and W. M. Wonham, 2010. Supervisor Localization: A Top-Down Approach to Distributed



- Control of Discrete-Event Systems. *IEEE transactions on automatic control*, vol. 55, NO. 3, March 2010.
- Cassandras, C.G and S. Lafortune, 2008. *The second edition of Introduction to Discrete Event Systems*, 2008.
- IEC 60848, 2002. *International Electrotechnical Commission*, "GRAFCET specification language for sequential function charts", 2002.
- IEC 61131-3, 2003. Programmable controllers, part 3: Programming languages. IEC.
- Khoumsi, A and L. Ouedraogo, 2010. New Results in Modeling and Supervisory Control of Real-Time Discrete Event Systems. *18th Mediterranean Conference on Control & Automation Congress* Palace Hotel, Marrakech, Morocco June 23-25, 2010.
- Khoumsi, A and H. Chakib, 2014. Decentralized Supervisory Control of Discrete Event Systems Moving Decisions Closer to Actions. *11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2014.
- Leduc, R. J, D., Pengcheng, and S. Raoguang, 2009. Synthesis Method for Hierarchical Interface-Based Supervisory Control. *IEEE transactions on automatic control*, vol. 54, NO. 7, July 2009.
- Ostroff, J. S, 1990. Deciding properties of timed transition models, *IEEE Trans. Parallel Distributed Systems* 1 (2) (1990) 170–183K.
- Philippot, A, 2006. *Contribution au diagnostic décentralisé des SED: application aux systèmes manufacturiers*. Thèse de Doctorat, Université de Reims Champagne Ardenne, France.
- Qamsane, Y, A. Tajer, and A. Philippot, 2014a. Synthèse de contrôle distribué pour un système manufacturier. *10ème Conférence Francophone de Modélisation, Optimisation et Simulation- MOSIM'14, 2014 - Nancy - France*.
- Qamsane, Y., A. Tajer, and A. Philippot, 2014b. Synthesis and implementation of distributed control for a flexible manufacturing system. *In Second World Conference on Complex Systems (WCCS)*, Agadir, Morocco pp. 323–329. 2014.
- Ramadge, P.J and W. M. Wonham, 1989. The control of discrete event systems. *Proceedings IEEE, Special Issue on DEDS*, 77(1):81-98.J.
- Sava, A.T and H. Alla, 2006. A control synthesis approach for time discrete event systems, *Mathematics and Computers in Simulation* 70 (2006) 250–265.
- Tajer, A., A. Philippot, and V. Carré-Ménétrier, 2011. Distributed Optimal Controller for Manufacturing Systems through Synthesis Approach. *International Conference on Communication, Computing and Control Applications (CCCA'11)*, IEEE, Hammamet, Tunisia.
- Tajer, A., A. Philippot et V. Carré-Ménétrier, 2013. Centralised controller for manufacturing systems through liveness extraction approach. *Pedrycz Witold, Inderscience publishers, International Journal of Systems, Control and Communications*, 5(3):189-213.
- Zhang, R., K. Cai, Y. Gan, Z. Wang, and W. M. Wonham, 2013. Supervision localization of timed discrete-event systems. *Automatica* 49(9): 27862794 (2013).