

---

# Approche multi-modèle pour les Systèmes à Événements Discrets : application à un préhenseur pneumatique

Oulaid KAMACH — Laurent PIÉTRAC — Éric NIEL

*INSA de Lyon, Laboratoire d'Automatique Industrielle (LAI)*

*Bat. Antoine de St-Exupery*

*27, Av. Jean Capelle*

*69621 Villeurbanne cedex. France*

*prénom.nom@insa-lyon.fr*

---

*RÉSUMÉ. Le travail présenté dans ce papier porte sur la gestion des modes de fonctionnement des SED. Notre contribution se base sur la théorie de contrôle par supervision initiée par les travaux de Ramadge et Wonham. Notre objectif est d'enrichir cette théorie par des mécanismes permettant l'étude des modes de fonctionnement. Nous proposerons dans un premier temps une approche, dite multi-modèle, qui permet de représenter les SED évoluant dans différents modes de fonctionnement par un ensemble de modèles simples décrivant chacun le comportement du système dans un mode donné. L'alternance de modes est assurée par un mécanisme de suivi. Ensuite, nous associons à chaque modèle du procédé un modèle de spécification, puis dans le même esprit nous aborderons le mécanisme de commutation entre les modèles de spécifications. Dans un second temps, nous montrerons l'application de notre approche sur un exemple de préhenseur pneumatique.*

*ABSTRACT. This work deals with operating mode management applied to Discrete Event Systems (DES). Our contribution is based on the Supervisory Control Theory (SCT). We aim at extending the SCT by introducing mechanism for studding the different operating modes of the system to be controlled. Our approach is multi-model, involves representing a complex system by a set of simple models, each of which describes the system in a given operating mode. A global monitoring mechanism ensures the correct switching between modes. Then, a unique specification model can be associated to each plant model. Specification models are also subject to switch, according to changes occurring in the plant model dynamics, and the control actions to be taken. After we illustrated our approach on an example.*

*MOTS-CLÉS : SED, Supervision, Théorie de Contrôle par Supervision, multi-modèle, gestion des modes de fonctionnement.*

*KEYWORDS: DES, Supervisory, Supervisory Control Theory, Multi-Model, Operating Mode Management.*

---

## 1. Introduction

Depuis plus que 20 ans, la théorie de contrôle par supervision a connu un essor sans précédent. Basée sur des concepts simples – l'état et l'événement – elle s'inscrit naturellement dans le cadre de l'étude des Systèmes à Événements Discrets (SED). Son originalité est de proposer un cadre formel permettant l'étude de problèmes – classiques en automatique continue mais appliqués ici à l'automatique discrète – tels que la synthèse de trajectoires dynamiques, la vérification de propriétés de contrôlabilité, d'atteignabilité ou d'observabilité. Ces particularités ont permis un développement important de cette théorie. De nombreux travaux ont ainsi participé à ce développement par la prise en compte de structures décentralisée [LIN 88], [RUDIE 92] [YOO 02] modulaire [RAMADGE 87a], hiérarchique [ZHONG 92] ou hiérarchique distribuée [WONG 00]. Certains de ces travaux [NOURELFATH 97] [CHAFIK 00] ont permis de montrer que la prise en compte des modes de fonctionnement du système nécessite, au delà du problème de l'explosion combinatoire, l'étude de structures de commande évoluées. Cependant cette prise en compte rend l'applicabilité des structures de commande très difficile, puisque :

- 1) les prescriptions (spécifications) imposées dans chaque mode de fonctionnement peuvent être contradictoires et entraîner un blocage du système ;
- 2) le procédé global peut être composé d'éléments qui ne sont pas tous exploités dans chacun des modes de fonctionnement.

Dans le cadre de cet article nous allons présenter une proposition d'extension de la théorie du contrôle par supervision permettant la prise en compte des différents modes de fonctionnement.

Le développement de ce papier se fera de la façon suivante : dans la section 2, l'approche multi-modèle sera définie pour ce cas généralisé à  $m$  modes de fonctionnement ( $m > 2$ ). Dans la section 3, nous définissons les mécanismes de construction des spécifications étendues de chaque mode. L'exemple retenu pour illustrer, dans la section 4, les résultats théoriques des sections 2 et 3 est un préhenseur pneumatique proposé par le groupe de travail COSED « Commande opérationnelle des Systèmes à Événements Discrets » qui fait maintenant partie du groupe INCOS<sup>1</sup> du Groupe de Recherche MACS<sup>2</sup> du CNRS. Enfin la conclusion sera présentée dans la section 5.

## 2. Commutation entre les modèles des procédés

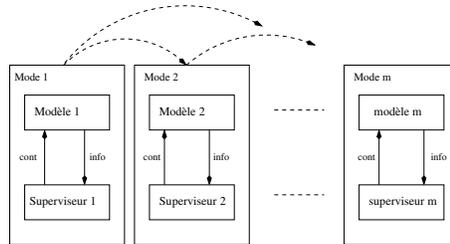
Nous considérons qu'un système ne peut être engagé que dans un mode de fonctionnement à la fois. Durant cet engagement la structure et la tâche du système restent figées. C'est à l'occurrence d'un événement conduisant à la modification de la structure (engagement ou désengagement de ressources) ou de la tâche à réaliser qu'il sera envisagé d'engager le nouveau système dans un mode différent. Un mode résulte

---

1. « Ingénierie de la COMmande et de la Supervision »

2. « Modélisation Analyse et Conduite des Systèmes dynamiques »

donc de l'activation d'un modèle du système auquel seront associées des spécifications relevant de la nouvelle tâche attendue. La démarche globale de notre proposition commence par l'étude du mode initialement sélectionné. Dans ce mode le système est caractérisé par un modèle du procédé  $G_1$  auquel est associé un modèle de spécification  $E_1$ . Ensuite nous caractérisons chaque mode de fonctionnement  $i$  par un modèle du procédé  $G_i$  auquel est associé un modèle de spécification  $E_i$  (figure 1). Nous nous intéressons par la suite au problème de la commutation entre les différents



**Figure 1.** Architecture de commande proposée

modèles en construisant dans un premier temps les modèles étendus des procédés et dans un second temps les modèles étendus des spécifications. L'étape de la synthèse des contrôleurs étendus se déduit à partir des modèles étendus des procédés et des spécifications.

Dans cette section, nous allons nous intéresser à la problématique du modèle du procédé représenté ici par un automate à états finis. L'événement de commutation permet de commuter d'un mode à un autre donc d'un modèle du procédé à un autre modèle. Le problème majeur lors de la commutation entre les différents modèles consiste à identifier l'état correct de départ (resp. l'état de retour) du modèle nouvellement activé (resp. renouvellement activé). Cet état dit de départ (car il ne correspondra pas forcément à l'état initial du procédé) doit être compatible avec l'état à partir duquel l'événement de commutation s'est produit : un état  $q$  du mode  $i$  est dit compatible avec un état  $q'$  du mode  $j$  si l'ensemble des ressources communes entre les deux modes ont la même activité dans les deux états considérés.

### 2.1. Définitions préliminaires

Le mode 1 sera toujours considéré comme le premier mode sélectionné, ainsi  $G_1$  sera le seul modèle du procédé initialement activé. L'activation/désactivation de modèles est consécutive à l'occurrence d'un événement de commutation. La désactivation d'un modèle signifie sa mise « en silence » : dans cet état particulier le modèle désactivé est passif à l'écoute des événements qui défilent. De ce fait, nous proposons d'intégrer au modèle d'origine un état inactif correspondant au fait que le modèle ne peut plus (étant désactivé) être réceptif aux événements présentés. Nous augmentons ainsi chaque modèle  $G_i$  de cet état particulier  $q_{i,in}$ , appelé état inactif, inspiré

du concept d'état puits proposé par [NOURELFATH 97] et de l'état non significatif proposé par [DANGOUMAU 00]. Comme un seul mode de fonctionnement est retenu à la fois, le modèle étendu  $G_{i,et}$  activé à un instant donné est le seul modèle ne se trouvant pas dans son état inactif. Réciproquement l'état de tous les modèles non activés est leur propre état inactif. Afin de prendre en considération plusieurs modes de fonctionnement, nous adopterons les notations suivantes. L'ensemble des modes de fonctionnement est noté  $I = \{1, 2, \dots, m\}$  où  $m \in \mathbb{N}$  et  $m > 2$ . A chaque mode de fonctionnement  $i$  est associé un modèle automate  $G_i = (Q_i, \Sigma_i, \delta_i, q_{i,0}, Q_{i,m})$ . L'ensemble  $\Sigma'$  des événements de commutation est défini par  $\bigcup_{i,j,i \neq j}^m \{\alpha_{i,j}\}$  où  $\alpha_{i,j}$  représente l'événement assurant la commutation du mode  $i$  vers le mode  $j$ . Ces nombreux événements de commutation signifient que plusieurs commutations sont possibles : passage du mode 1 au mode 2, passage du mode 1 au mode  $i$ , du mode 2 au mode  $k$ , etc. Toutes ces commutations induisent une étape de mémorisation de traces plus imposante. Prenons le cas où dans un mode  $i$  il existe une commutation vers le mode  $j$ , puis à partir de celui-ci une commutation vers le mode  $k$ . Dans ce cas, il est nécessaire de mémoriser l'historique du procédé  $G_i$  dans le mode  $i$ , avant la première commutation, puis celui du procédé  $G_j$  dans le mode  $j$ , avant la seconde commutation. Toutes ces mémorisations des historiques sont nécessaires à la détermination des états de départ dans chacun des modes vers lequel aboutit une commutation. ces mémorisations sont réalisées par des canaux d'information notés  $\pi_{i,j}$ .

Formellement, le canal d'information  $\pi_{i,j}$  est une fonction de projection. Cette fonction de projection est définie par :

**Definition 2.1** (Projection étendue)

$$\begin{aligned} \text{Soit } \pi_{i,j} : \Sigma_i^* &\longrightarrow \Sigma_j^* \text{ telle que } \forall \sigma \in \Sigma_i \text{ et } \forall s \in \Sigma_i^* : \\ \pi_{i,j}(\varepsilon) &= \varepsilon \\ \pi_{i,j}(s\sigma) &= \begin{cases} \pi_{i,j}(s)\sigma & \text{si } \sigma \in \Sigma_i \cap \Sigma_j \\ \pi_{i,j}(s) & \text{si } \sigma \in \Sigma_i \setminus \Sigma_j \end{cases} \end{aligned}$$

Cette nouvelle définition de projection ne contraint pas les alphabets  $\Sigma_i$  et  $\Sigma_j$ , c'est pourquoi nous l'appelons projection étendue. Dans le cas particulier où  $\Sigma_j \subseteq \Sigma_i$ , elle correspond à la projection naturelle classiquement utilisée dans la théorie de contrôle par supervision. La projection  $\pi_{i,j}$  permet d'observer uniquement l'occurrence des événements appartenant à l'intersection  $\Sigma_i \cap \Sigma_j$  et donc de ne suivre que les ressources communes entre les modèles  $G_i$  et  $G_j$ .

Le modèle étendu du procédé pour chacun des modes de fonctionnement  $i \in I$  est donné par le modèle automate  $G_{i,et}$  formellement défini par :

$$G_{i,et} = (Q_{i,et}, \Sigma_{i,et}, \delta_{i,et}, q_{i,0,et}, Q_{i,m,et}) \text{ avec :}$$

- $Q_{i,et} = Q_i \cup \{q_{i,in}\}$  ;
- $\Sigma_{i,et} = \Sigma_i \cup \Sigma'_i$ , où  $\Sigma'_i$  est l'ensemble des événements permettant de quitter ou de revenir au mode  $i$  ;

$$- q_{i,0,et} = \begin{cases} q_{i,0} & \text{si } i = 1 \\ q_{i,in} & \text{si } i \neq 1 \end{cases}$$

$$- Q_{i,m,et} = Q_{i,m}.$$

– La fonction étendue de la transition  $\delta_{i,et}$  est définie comme suit :

$$1) \forall q \in Q_i \text{ et } \forall \sigma \in \Sigma_i \text{ si } \delta_i(q, \sigma) \text{ existe alors } \delta_{i,et}(q, \sigma) := \delta_i(q, \sigma);$$

2)  $\forall q \in Q_i$  à partir duquel l'événement de commutation  $\alpha_{i,j}$  peut être généré alors  $\delta_{i,et}(q, \sigma) := q_{i,in}$  ;

–  $\delta_{j,et}(q_{j,in}, \alpha_{i,j})$ , avec  $((i, j) \in I \times I \text{ et } i \neq j)$  sera définie ultérieurement.

L'adjonction d'un état inactif unique dans le modèle  $G_{j,et}$  conduit malheureusement à un indéterminisme qu'il s'agit maintenant de lever. En effet, à partir d'un état inactif  $q_{j,in}$ , il existe, pour le même événement de commutation  $\alpha_{i,j}$ , plusieurs états d'arrivée selon l'historique enregistré avant la commutation. Pour lever cet indéterminisme, nous définissons un ensemble d'événements permettant, par labellisation, de distinguer dans le modèle  $G_{j,et}$  les différentes occurrences de l'événement de commutation  $\alpha_{i,j}$  :

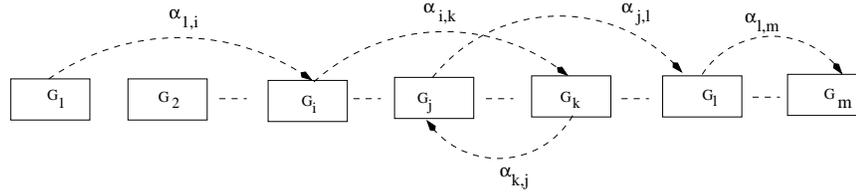
**Proposition 2.2** *Afin de lever l'indéterminisme des modèles, nous associons à l'événement de commutation  $\alpha_{i,j}$  un ensemble défini par :*

$$\alpha_{i,j} = \alpha_{i,j,k} \text{ si } \delta_j(q_{j,0}, \pi_{i,j}(s)) = q_{j,k}$$

## 2.2. Formalisation du mécanisme de commutation

Avant de déterminer l'ensemble des états de raccordement d'une commutation donnée, nous précisons que pour un mode donné plusieurs événements de commutation peuvent être considérés. De plus, à partir d'un mode  $i$  un événement de commutation peut mener vers un autre mode  $k$  où  $i$  et  $k$  ne sont liés par aucune relation d'ordre particulière. En d'autres termes, la numérotation des modes ne représente absolument pas un ordre d'activation des modes. Ainsi l'occurrence d'un événement de commutation  $\alpha_{i,k}$  (avec  $i \neq k$ ) à partir d'un modèle  $G_i$  peut mener vers un modèle  $G_k$ . Ensuite l'occurrence d'un événement de commutation  $\alpha_{k,j}$  peut conduire vers un modèle  $G_j$  où  $j$  peut aussi bien être inférieur que supérieur à  $k$  (avec également  $j \neq k$ ). Ce mécanisme de commutation est illustré par la figure 2.

Pour déterminer les états de départ de  $G_m$  (figure 2), nous devons déterminer ceux de  $G_l$ . Les états de départ de  $G_l$  sont déterminés en fonction de ceux de  $G_j$  qui sont eux aussi fonction de ceux de  $G_k$ , eux-mêmes déterminés à partir des états de départ de  $G_i$ . Enfin les états de départ de  $G_i$  sont déterminés à partir du modèle  $G_1$  initialement actif. Ces déterminations successives impliquent de déterminer le suivi des évolutions des ressources communes à deux modes successifs. Le principe consiste à calculer d'abord les états de départ des modes pouvant être sélectionnés à partir du mode initial, puis ceux pouvant être sélectionnés à partir de ces modes et ainsi de suite. Afin d'éviter de refaire les mêmes calculs il faut en outre vérifier au fur et à mesure s'il reste des



**Figure 2.** Commutation entre les modèles des procédés

commutations à traiter. D'une manière formelle l'ensemble des états de départ dans le mode  $m$  sera donné par la proposition généralisée suivante.

**Proposition 2.3** Soient les modèles  $G_1, G_2, \dots, G_m$  caractérisant chaque mode de fonctionnement.

1) Déterminer  $C$  une fonction partielle définissant les commutations possibles.  $i \rightarrow j$  appartient à  $C$  si et seulement si il existe une commutation du mode  $G_i$  vers  $G_j$ .

2)  $I = \{1\}$ .  $I$  représente l'ensemble des indices des modes à partir desquels nous allons considérer les événements de commutation. Nous partons du mode initial.

3) Tant que  $I \neq \{\}$  faire :

a)  $L = \{\}$ .  $L$  est un ensemble temporaire permettant de déterminer les indices des modes à partir desquels seront envisagées les commutations au pas suivant.

b) Pour chaque  $i \in I$  : soit  $L_i$  l'ensemble des modes tel que quel que soit  $j$  appartenant à  $L_i$ , la commutation  $i \rightarrow j$  fasse partie de  $C$ .

c) Pour tout  $G_j$  tel que  $j \in L_i$  :

d) Déterminer l'ensemble des états de départ en appliquant :

$$\delta_{j,et}(q_{j,in}, \alpha_{i,j}) = \delta_j(q_{j,0}, \pi_{i,j}(K_{q,q'}))$$

où  $K_{q,q'}$  est le langage contenant toutes les séquences ayant comme état d'origine un état de départ  $q$  du modèle  $G_i$  et comme état final un état de départ  $q'$  de ce modèle ( $\forall s \in K_{q,q'}, \alpha_{i,j} \in \text{suiv}(s)^3$ ). Ceci doit être fait pour tous les langages  $K_{q,q'}$  (il y a plusieurs états  $q$  et  $q'$  possibles). Pour chaque état de départ de  $G_j$  déterminé, l'occurrence de l'événement  $\alpha_{i,j}$  sera labellisée suivant la règle :

$$\alpha_{i,j} = \alpha_{i,j,k} \text{ si } \delta_{j,et}(q_{j,in}, \alpha_{i,j}) = q_{j,k}$$

e)  $C = C - \{i \rightarrow j\}$ ;

$i \rightarrow j$  représente la commutation du mode  $i$  vers le mode  $j$ .

$$f) L = (L \cup L_i) \cap \text{dom}(C)^4$$

4) faire  $I = L$

3.  $\text{suiv}(s)$  est l'ensemble des événements qui suivent la séquence d'événements  $s$ .

4.  $\text{dom}(C)$  représente le domaine de la fonction  $C$ , c'est-à-dire l'ensemble des indices  $i$  tel que  $i \rightarrow j$  appartient à  $C$ .

### 3. Un modèle de spécification par mode

Dans le même esprit que dans la section 2 nous abordons dans cette section le mécanisme de commutation entre les modèles de spécifications. Il s'agira de rechercher les états à partir desquels ces modèles doivent être initialisés afin de garantir une cohérence entre la dynamique courante du procédé et les actions de contrôle souhaitées (les actions d'interdiction ou d'autorisation). Puisque chaque modèle de spécification est associé à un modèle du procédé, un changement de mode de fonctionnement permet d'obtenir des dynamiques distinctes. N'oublions pas que nous devons activer chaque modèle de spécification depuis un état permettant d'ajuster les actions de contrôle compatibles avec la nouvelle dynamique du procédé. Le mécanisme de commutation utilisé précédemment pour les modèles des procédés ne peut pas être appliqué pour les modèles de spécification. En effet, lorsqu'il y a une commutation d'un modèle  $G_i$  à un modèle  $G_j$ , la trace générée dans  $G_i$  peut comporter des événements qui sont interdits à partir de l'état initial de la spécification  $E_j$ , et donc qui n'apparaissent pas dans cette spécification.

Soit  $E_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i,m})$  la spécification associée au procédé  $G_i = (Q_i, \Sigma_i, \delta_i, q_{i,0}, Q_{i,m})$ . Initialement, le mode 1 est supposé être sélectionné donc l'automate de procédé  $G_1$  se trouve dans l'état initial  $q_{1,0}$ . Le modèle de la spécification  $E_1$  se trouve également activé, son modèle est dans l'état initial  $x_{1,0}$ . A l'occurrence de l'événement de commutation  $\alpha_{1,j}$ , le modèle  $G_{1,et}$  sera conduit vers son état inactif  $q_{1,in}$ . La spécification  $E_1$  ne jouant plus de rôle, son modèle doit être étendu par un état inactif  $x_{1,in}$  atteint par  $\alpha_{1,j}$ . Cet état sera conservé jusqu'à l'occurrence d'un événement de commutation  $\alpha_{k,1}$  lui permettant ainsi de rejoindre un état  $x \in X_1$ . Rappelons que lorsque le procédé  $G_1$  se trouve dans un état  $q \in Q_1$  ( $q \neq q_{1,in}$ ), les modèles  $G_i$  ( $i \neq 1$ ) sont forcément plongés dans leurs états inactifs. De façon similaire, lorsque la spécification  $E_1$  se trouve dans un état  $x \in X_1$ , les spécifications  $E_i$  ( $i \neq 1$ ) se trouvent dans leurs états inactifs. Ce phénomène de commutation reste néanmoins un problème nouveau. En effet, si le modèle du procédé  $G_i$  possède plusieurs états de raccordement (départ et/ou retour), la spécification correspondante doit posséder également plusieurs états de départ et/ou de retour afin de suivre correctement l'évolution de son modèle et par conséquent agir convenablement sur celui-ci.

Soit  $q_{i,m}$  l'état de départ du procédé  $G_{i,et}$ . Pour déterminer l'état de départ de la spécification  $E_{i,et}$  adéquat avec l'état de départ du procédé  $G_i$  nous caractérisons dans un premier temps le langage désiré  $K_{i,q}$  élaboré à partir de l'état  $q_{i,m}$ . Dans un second temps nous recherchons l'état compatible de la spécification  $E_{i,et}$ . La notion de compatibilité est définie ici de telle façon que l'intersection du langage de la spécification avec celui du procédé soit le langage désiré  $K_{i,q}$ . La proposition ci-dessous présente le cadre formel de cette détermination.

**Proposition 3.1** *Soit  $q_{i,m}$  un état de départ du procédé  $G_i$ . Il existe un état unique  $x_{i,l}$  de la spécification  $E_i$  tel que  $L(G_i, q_{i,m}) \cap L(S_i, x_{i,l}) = K_{i,m}$ , avec  $K_{i,m}$  langage désiré élaboré à partir de  $q_{i,m}$ ,  $L(G_i, q_{i,m})$  est le langage du procédé  $G_i$  ayant comme*

état initial l'état  $q_{i,m}$  et  $L(S_i, x_{i,l})$  est le langage de spécification  $S_i$  ayant comme état initial l'état  $x_{i,l}$ .

Pour de plus amples informations concernant le mécanisme permettant de construire les modèles étendus des spécifications, le lecteur pourra se référer à [KAMACH 04].

## 4. Application sur l'exemple

### 4.1. Description du système

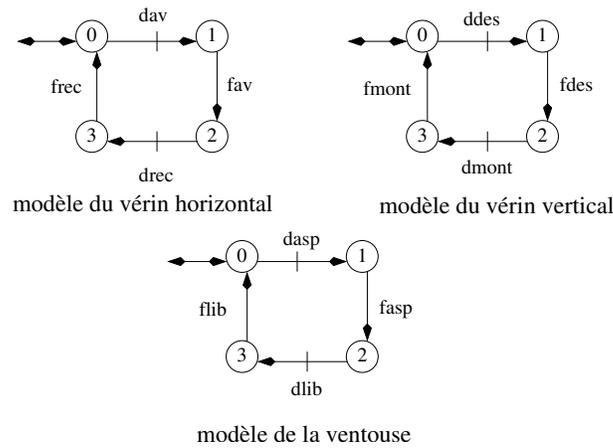
Le système à étudier est un préhenseur pneumatique déplaçant un produit d'un poste de prise à un poste de dépose suivant un cycle en « U ». Il est composé de trois chaînes fonctionnelles qui assurent le mouvement vertical, horizontal et l'aspiration. Les déplacements horizontaux et verticaux sont assurés par deux vérins « double effet » pilotés respectivement par un distributeur 5/2 bistable et un distributeur 5/2 monostable. Chacun d'entre eux est équipé de capteurs de fin de course magnétiques. La préhension est assurée par un système Venturi.

Deux modes de fonctionnement ont été retenus pour ce système : le mode nominal (noté  $n$ ) et le mode dégradé (noté  $d$ ). Dans le mode nominal le système remplit la fonctionnalité pour laquelle il a été conçu (voir paragraphe suivant). Au cours du fonctionnement nominal, l'occurrence d'un événement d'exception (ici de défaillance) engagera le système en fonctionnement dégradé dans lequel il peut accomplir une partie de sa tâche en dépit de la défaillance considérée. Initialement le système est dans le mode nominal. Le préhenseur est en position haute, à gauche (position d'attente) et il n'y a pas d'aspiration.

### 4.2. Procédé dans le mode nominal $G_n$

La modélisation du procédé consiste d'abord à décomposer le système en sous-systèmes ou ressources considérées comme élémentaires, c'est-à-dire des ressources dont le concepteur construit directement le modèle. Il s'agit ici des trois chaînes fonctionnelles présentées précédemment<sup>5</sup>. Le mode nominal du système sera ensuite décrit par un seul modèle automate  $G_n$  obtenu par la composition parallèle des modèles de chacun des trois ressources. Les modèles automates correspondant respectivement au vérin vertical (VV), vérin horizontal (VH) et à la ventouse (V) sont donnés par la figure 3. Le modèle du procédé en mode nominal  $G_n$  comporte 64 états et 192 transitions. Vu le nombre conséquent d'états et de transitions, ce modèle ne sera pas représenté ici pour des raisons de lisibilité.

5. Par abus de langage nous parlerons dans la suite de cet article de vérin horizontal, de vérin vertical et de ventouse plutôt que de chaînes fonctionnelles assurant le mouvement horizontal, le mouvement vertical et l'aspiration.



**Figure 3.** Modèles des ressources dans le mode nominal

Événements contrôlables		Événements incontrôlables	
dav	Demande d'avance	fav	Fin d'avance
drec	Demande de recul	frec	Fin de recul
ddes	Demande de descente	fdes	Fin de descente
dmont	Demande de montée	fmont	Fin de montée
dasp	Demande d'aspiration	fasp	Fin d'aspiration
dlib	Demande de libération	flib	Fin de libération

**Tableau 1.** Liste des événements contrôlables et incontrôlables

**4.3. Spécifications du mode nominal  $E_n$**

Les spécifications nominales de commande peuvent être les suivantes :

- 1) L'avance ne peut se faire que si le vérin vertical est en haut et si une pièce est aspirée ;
- 2) Le recul ne peut se faire que si le vérin vertical est en position haute et qu'une pièce a été libérée ;
- 3) Une pièce ne peut être saisie que si le vérin vertical est en position basse et le vérin horizontal en position rentrée ;
- 4) Une pièce ne peut être libérée que si le vérin vertical est en position basse et le vérin horizontal est complètement sorti ;
- 5) Le vérin vertical ne monte que si une pièce est aspirée ou libérée ;
- 6) Après être monté, le vérin vertical ne peut descendre que si le vérin horizontal est complètement sorti.

#### 4.4. Vérification de la contrôlabilité

Maintenant que nous avons d'une part défini le modèle non étendu du procédé dans le mode nominal et d'autre part les modèles des spécifications nominales, nous effectuons le calcul du contrôleur non étendu en mode nominal. Le modèle du procédé sous contrôle (figure 4) est obtenu par le produit de  $G_n$  et de  $E_n$ . Il comporte 16 états et 16 transitions, et sa construction à l'aide du logiciel « TCT »<sup>6</sup> nous permet de vérifier que ce modèle est contrôlable.

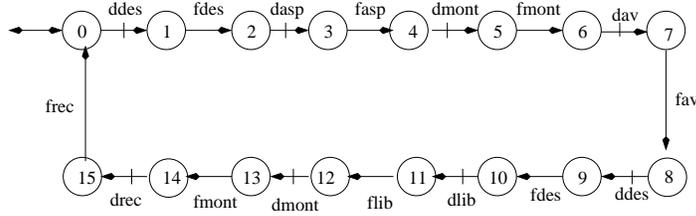


Figure 4. Contrôleur du mode nominal

#### 4.5. Mode dégradé

Afin de prendre en considération le mode de fonctionnement dégradé et plus tard l'alternance du mode nominal vers le mode dégradé, le modèle du vérin horizontal doit être augmenté par des événements de défaillance et de réparation ( $p$  et  $r$ ) (figure 5). Dans notre exemple, les événements  $p$  et  $r$  sont des événements de commutation.

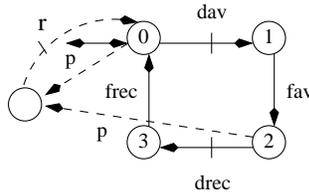


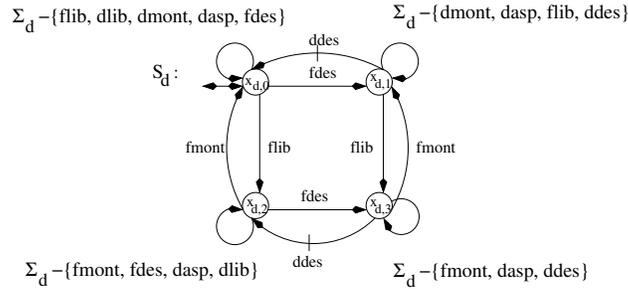
Figure 5. Modèle étendu du vérin horizontal

##### 4.5.1. Procédé dans le mode dégradé $G_d$

Le mode dégradé est enclenché suite à la perte fonctionnelle du vérin horizontal. La nouvelle configuration fonctionnelle du système est réduite alors à la ventouse et

6. « TCT » est un outil permettant la mise en oeuvre de la théorie RW. Il peut être récupéré sur le site Web du professeur Wonham à l'université de Toronto : <http://www.control.utoronto.ca/people/profs/wonham/>





**Figure 7.** Spécification globale du mode dégradé

rectement l'alternance et la cohérence du mode nominal vers le mode dégradé, nous avons besoin du canal d'information  $\pi_{n,d}$  intercalé entre les deux modèles des procédés.

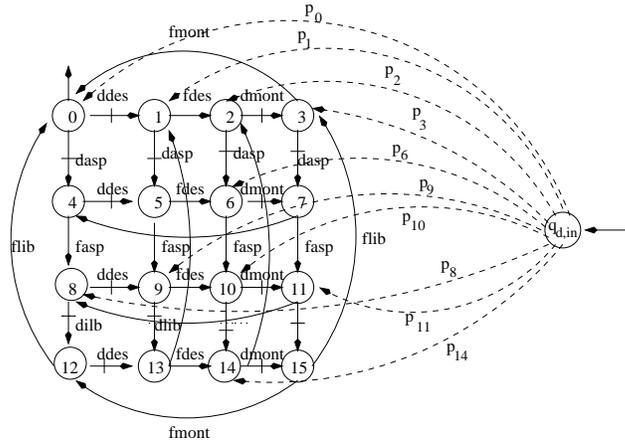
La panne survenant dans l'état 2 ou l'état 0 du composant vérin horizontal peut donc survenir entre les états 0 et 6 ou entre les états 8 et 14 (inclus) du procédé supervisé (figure 4). Supposons que cette panne arrive dans l'état 8 du graphe de la figure 4. Pour déterminer l'état de départ du procédé  $G_d$  en mode dégradé nous cherchons le langage  $K_{n,8}$  qui part depuis l'état initial et mène vers l'état 8. En utilisant la règle d'Arden, nous trouvons :  $K_{n,8} = (\text{d des.f des.d asp.f asp.d mont.f mont.d av.f av.d des.f des.d lib.f lib.d mont.f mont.d rec.f rec})^* . (\text{d des.f des.d asp.f asp.d mont.f mont.d av.f av})$ . Donc l'état de départ suite à  $K_{n,8}$  est

$$\delta_{d,et}(q_{d,in}, p) = \delta_d(q_{d,0}, \pi_{n,d}(K_{n,8})) = q_{d,8}.$$

L'exemple donné précédemment correspondait à une panne arrivant dans l'état 8 du procédé. Le même raisonnement peut être adopté si cette panne survient dans un autre état. Pour construire le procédé étendu du mode dégradé, nous appliquons notre proposition sur tous les langages  $K$  du procédé supervisé du mode nominal qui mènent vers un état où l'événement de panne peut se produire et nous obtenons le modèle étendu du mode dégradé de la figure 8. L'événement de panne sera représenté par des événements significatifs de type  $p_i : p = p_i$  si  $\delta_{d,et}(q_{d,in}, p) = q_{d,i}$ . La même démarche doit être appliquée à l'événement de réparation. Le modèle de la figure 8 n'est pas complet. En effet, l'objectif de cette figure est de mettre l'accent sur les états de départ du modèle à partir de l'état inactif  $q_{d,in}$ . L'événement de réparation  $r$  n'apparaît donc pas dans le modèle, puisque cet événement ne fait que conduire le modèle vers l'état inactif  $q_{d,in}$ .

#### 4.5.4. Modèle du procédé étendu $G_{n,et}$

Avec la même démarche que pour l'alternance du mode nominal vers le mode dégradé, nous déterminons l'ensemble des états de retour dans le modèle du procédé en mode nominal.



**Figure 8.** Représentation partielle du modèle étendu du mode dégradé

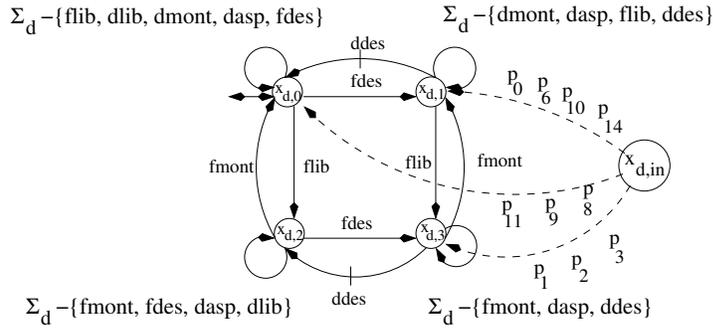
4.5.5. Spécification étendue en mode dégradé  $E_{d,et}$

Pour établir le modèle étendu de la spécification du mode dégradé, il faut caractériser la fonction de transition  $\xi_{d,et}(x_{d,in}, p)$ . Ceci permettra de déterminer l'ensemble des états de départ de la spécification  $E_d$ . L'état de départ de la spécification est relatif à l'état de départ du procédé  $G_d$ . Supposons qu'après la génération de la trace  $s = ddes.fdes.dasp.fasp.dmont.fmont.dav.fav$ , l'événement de panne  $p$  s'est produit, dans ce cas l'état de départ du procédé dégradé est  $q_{d,8}$  (paragraphe précédent). Depuis cet état le système se trouve au-dessus du poste de dépose avec une pièce saisie par la ventouse et que le vérin vertical est en position haute. Le langage désiré depuis cet état est  $K_{d,8} = \{d des, d des f des, d des f des d lib, d des f des d lib f lib, d des f des d lib f lib d mont, \dots\}$ .

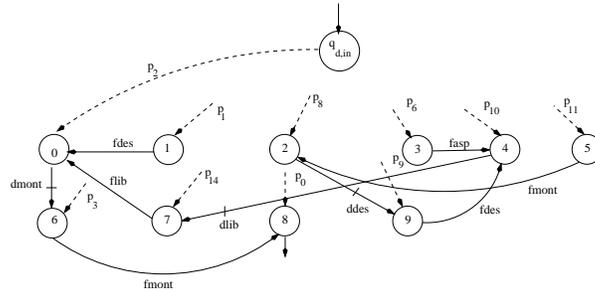
Pour déterminer l'état de départ de la spécification  $E_d$ , nous rappelons la proposition 3.1. L'état de départ de la spécification  $E_d$  qui correspond à  $L(G_d, q_{d,8}) \cap L(E_d, x_j) = K_{d,8}$  est  $x_{d,0}$ . Le même raisonnement peut être adopté pour déterminer les autres états de départ. Le modèle étendu de spécification du mode dégradé est donné par la figure 9.

4.6. Contrôleur étendu en mode dégradé  $C_{d,et}$

La contrôlabilité et le non-blocage du contrôleur résultant étendu sont vérifiés. Le contrôleur résultant dans le mode dégradé admet 11 états et 16 transitions (figure 10) l'événement de réparation pourrait apparaître dans ce modèle, il conduit l'automate vers l'état inactif  $q_{d,in}$ .



**Figure 9.** Spécification étendue en mode dégradé



**Figure 10.** Représentation partielle du contrôleur étendu du mode dégradé

**4.7. Spécification étendue  $E_{n,et}$**

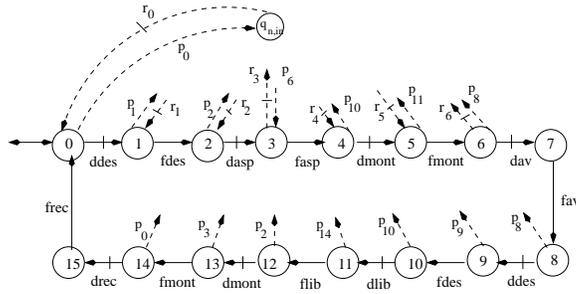
Pour établir le modèle étendu de spécification  $E_n$  en mode nominal, nous devons déterminer l'ensemble des états de retour de cette spécification. Ces états doivent être compatibles avec la nouvelle dynamique du procédé. Ils sont obtenus en résolvant l'équation suivante :

$$L(G_n, q) \cap L(E_n, x_j) = K_{n,q}, \text{ où } K_{n,q} \text{ représente le langage désiré compatible avec la nouvelle dynamique du procédé en mode nominal } L(G_n, q).$$

On peut adopter la même démarche pour déterminer les états de retour de la spécification  $E_n$ .

**4.8. Contrôleur étendu en mode nominal  $C_{n,et}$**

Le contrôleur étendu en mode nominal est obtenu par le produit des modèles étendus de procédé  $G_{n,et}$  et de spécification  $E_{n,et}$ , il est donné par la figure 11



**Figure 11.** *Contrôleur étendu du mode nominal*

A partir de cet exemple, les deux modes de fonctionnement retenus sont le mode nominal et le mode dégradé. Le fonctionnement en mode nominal est assuré par trois éléments du système. Au cours du fonctionnement nominal, l'occurrence de l'événement de défaillance  $p$  engage le système dans un mode dégradé. A chaque mode correspond un modèle du procédé et un modèle des spécifications. Au départ les événements de défaillance  $p$  et de réparation  $r$  (événements de commutation) ne sont pas considérés dans les modèles des procédés et des spécifications du fonctionnement nominal. Ceux-ci sont considérés dans une seconde étape où sont définis les modèles étendus du procédé et des spécifications en mode dégradé. Ces modèles sont construits à partir du premier type d'alternance qui est caractérisé par le canal d'information  $\pi_{n,d}$ . Enfin en troisième étape le deuxième type d'alternance (dégradé vers le nominal), caractérisé par le canal d'information  $\pi_{d,n}$ , permet de déterminer les états à partir desquels on reprend le fonctionnement en mode nominal et donc de construire en compatibilité les modèles étendus du procédé et de spécification. La synthèse des contrôleurs en mode nominal et en dégradé est basée sur ces différents modèles étendus (procédés et spécifications). Les contrôleurs étendus du mode nominal et dégradé sont contrôlables, non-bloquants et simples.

## 5. Conclusion

Le travail présenté dans cet article constitue une extension de la théorie de contrôle par supervision pour la gestion des modes de fonctionnement. Les modes de fonctionnement représentent l'ensemble des comportements admissibles du système pour une configuration donnée des ressources. Le problème qui se pose est alors d'obtenir un contrôleur par mode. Il s'agit dans un premier temps de caractériser et de représenter les différents modes possibles d'un système et dans un second temps de gérer correctement l'alternance entre ces modes. Notre approche est une approche dite multi-modèle, elle consiste à représenter un système complexe par un ensemble des modèles simples. Chaque modèle du procédé commandé représente le comportement exigé du système dans un mode de fonctionnement donné. Le passage d'un mode à l'autre est

dû à l'occurrence d'un événement, dit événement de commutation. Le principal avantage de cette approche est d'une part, elle permet de matérialiser très simplement le mécanisme de suivi entre les modèles des procédés et de spécifications. D'autre part, elle permet de contourner et de maîtriser l'explosion combinatoire de l'espace d'états en considérant un ensemble de modèles simples et distincts au lieu de se focaliser sur un modèle unique représentant le système.

## 6. Bibliographie

- [LIN 88] LIN F., WONHAM W. M., « Decentralized supervisory control of discrete event systems », *Information Sciences*, vol. 25, 1995, p. 1202-1208.
- [RUDIE 92] RUDIE, K., WONHAM W. M., « Think globally, act locally : Decentralized supervisory control », *IEEE transactions on automatic control*, vol. 37, 1994, p. 1692-1708.
- [YOO 02] YOO, S., LAFORTUNE, S., « Decentralized supervisory control : a new architecture with a dynamic decision fusion rule », *6th International Workshop On Discrete fusion rule*, Saragosse, Espagne, 2002, p. 11-17.
- [RAMADGE 87a] RAMADGE, P. J. G., WONHAM W. M., « Modular feedback logic for discrete event systems », *SIAM Journal of Control and optimisation*, vol. 25, n° 5, 1987, p. 1202-1218.
- [ZHONG 92] ZHONG, H., « Hierarchical control of discrete event systems Contribution à la modélisation des câbles », Thèse de doctorat, Université de Toronto, Canada, 1992.
- [WONG 00] WONG, K., THISTLE, J. G., MALHAME, R., HOANG, H. « Supervisory Control of distributed Systems : conflict resolution », *Discrete Event Dynamic Systems : Theory and applications*, vol. 10, 2000, p. 131-186.
- [NOURELFATH 97] NOURELFATH, M., « Modular supervisory control of an experimental manufacturing system », Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 1997.
- [CHAFIK 00] CHAFIK, S., « Proposition d'une structure de contrôle par supervision hiérarchique et distribuée : application à la coordination », Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2000.
- [DANGOUMAU 00] DANGOUMAU, M., « Contribution à la gestion des modes des systèmes automatisés de production », Thèse de doctorat, Université des Sciences et Technologiques de Lille, 2000.
- [KAMACH 04] KAMACH, O., « Approche Multi-Modèle pour les Systèmes à Événements Discrets : application à la gestion des modes de fonctionnements », Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2004.