

Commandes multi-contrôleurs et multi-modèle pour les SED

¹*Oulaid KAMACH,*² *Laurent PIETRAC,*² *Eric NIEL*

¹Ecole Nationale des Sciences Appliquées de Tanger (ENSAT)
BP. 1818

Tanger Principal

²Institut National des Sciences Appliquées de Lyon (INSA)

okamach@ensat.ac.ma

RESUME :

Les travaux présentés dans cet article portent sur la synthèse de contrôleurs pour la gestion des modes de fonctionnement des Systèmes à Événements Discrets (SED). Notre contribution se base sur la théorie de contrôle par supervision. Notre objectif est d'enrichir cette théorie par des mécanismes permettant l'étude des modes de fonctionnement tout en gardant des modèles de taille raisonnable. Nous proposons 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 chacun des modèles des procédés un modèle unique de spécification. Comme les modèles des procédés, ces modèles de spécification sont soumis à un mécanisme de commutation : ils doivent être activés depuis un état permettant d'ajuster la nouvelle dynamique du procédé avec les actions de contrôle à mener. Enfin nous présenterons un exemple pour illustrer notre approche.

Mots-clés : SED, automates à états, synthèse de contrôleurs, gestion des modes de fonctionnement.

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.

Keywords : DES, Supervisory, Supervisory Control Theory, Multi-Model, Operating modes management.

1. Introduction

S'agissant de la synthèse de contrôleurs [1] [2], [3], il se révèle parfois très difficile, en s'appuyant sur des méthodes de contrôles classiques (contrôleur unique associé au système), de commander convenablement un système complexe [8]. Les solutions proposées consistent à mettre en oeuvre un ensemble de superviseurs élémentaires agissant sur une fraction de l'espace d'états complet. Rappelons que le comportement désiré du procédé dans un mode de fonctionnement est décrit par des spécifications qui lui sont propres. Elles peuvent donc être conflictuelles, et entraîner un blocage du système. Ceci justifie la proposition d'une architecture de commande employant un ensemble de superviseurs, où chacun d'entre eux est élaboré spécifiquement à un modèle du procédé (figure 1). Le rôle de chacun de ces superviseurs consiste à faire respecter les contraintes imposées dans chacun des modes retenus. Cette approche pour la synthèse des contrôleurs est donc cohérente avec l'approche multi-modèle retenue pour le procédé. Nous montrons dans cet article que les techniques de synthèse employées permettent d'obtenir des contrôleurs simples et performants (optimalité, non-blocage,...) et que les contrôleurs résultants restent de taille admissible. La problématique considérée ici doit prendre en compte, au-delà du mécanisme de commutation des procédés, celui des superviseurs. Le problème majeur de la synthèse de contrôleurs dans l'approche multi-modèle proposée ici consiste à assurer le raccordement entre la dynamique (le comportement) du procédé et l'action de contrôle à mener.

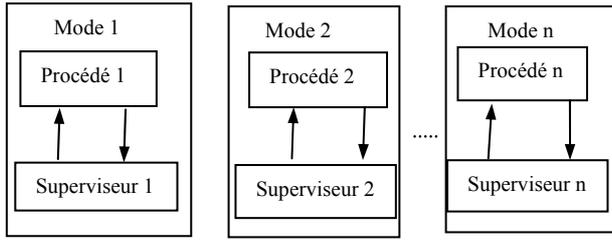


Figure 1 : Architecture de commande proposée

2. Approche multi-modèle

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 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 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 succinctement 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 $q_{in,i}$, appelé état inactif, inspiré du concept d'état puits proposé par [7] et de l'état non significatif proposé par [8]. 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. L'ensemble Σ' des événements de commutation est défini par $\cup_{i,j} \alpha_{i,j}$ 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

$$\pi_{i,j} : (\Sigma_i)^* \rightarrow (\Sigma_j)^* \text{ telle que } \forall s \in (\Sigma_i)^* \text{ et } \forall \sigma \in \Sigma_i$$

$$\pi_{i,j}(\varepsilon) = \varepsilon$$

$$\pi_{i,j}(s\sigma) = \begin{cases} \pi_{i,j}(s\sigma) & \text{si } \sigma \in \Sigma_i \cap \Sigma_j \\ \pi_{i,j}(s) & \text{sinon} \end{cases}$$

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 Σ'_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 :

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

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

- $\delta_{i,et}(q_{j,in}, \alpha_{i,j})$ avec $((i, j) \in I \times I$ 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 1

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

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

2.2. Formalisation du mécanisme de commutation

D'une manière formelle l'ensemble des états de départ dans le mode n sera donné par la proposition généralisée suivante.

Proposition 2

Soient les modèles G_1, G_2, \dots, G_n caractérisant chaque mode de fonctionnement.

- 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 .

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

- Tant que $I \neq \{\}$ faire :

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

- 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 .

- Pour tout G_j tel que $j \in L_i$:

- 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 ($\forall s \in K_{q,q'}, \alpha_{i,j} \in \text{suiv}(s)$, avec $\text{suiv}(s)$ est l'ensemble des événements qui suivent la séquence d'événements s). Ceci doit être fait pour tous les langages $K_{q,q'}$ (il y a plusieurs états q et q' possible). Pour chaque état de départ de G_j

déterminé, l'occurrence de l'événement $\alpha_{i,j}$ sera labellisée suivant la règle :

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

- $C = C - i \rightarrow j$;

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

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

- faire $I=L$. \square

3. Un Modèle de spécification par mode

Dans le même esprit que la section précédente 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 (section précédente) 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. D'où l'impossibilité d'utiliser le mécanisme de suivi du procédé pour la 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 son 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 $\alpha_{i,j}$, 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 $\alpha_{i,j}$. 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 autres modèles G_j ($j \neq 1$) sont forcément plongés dans leurs états inactifs respectifs. De façon similaire, lorsque la spécification E_1 se trouve dans un état $x \in X_1$, les spécifications autres que E_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. De ce fait, les deux sections ci-après présentent une méthode formelle pour la détermination des états de raccordement de la spécification nouvellement activée par l'occurrence d'un événement de commutation.

3.1 Etat de départ du modèle des spécifications E_i

Tout d'abord les modèles de spécifications de chaque mode doivent être étendus afin de garantir la sélection d'un seul mode à la fois. Soit la spécification

$E_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i,m})$, son modèle étendu est :

$E_{i,et} = (X_{i,et}, \Sigma_{i,et}, \xi_{i,et}, x_{i,0,et}, X_{i,m,et})$

avec :

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

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

- $X_{i,m,et} = X_{i,m}$

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

- $\forall x \in X_i$ et $\forall \sigma \in \Sigma_i$ si $\xi_i(x, \sigma)$ existe alors $\xi_{i,et}(x, \sigma) := \xi_i(x, \sigma)$
- $\forall x \in X_i$ à partir duquel l'événement de commutation $\alpha_{i,j}$ peut être généré alors $\xi_{i,et}(x, \alpha_{i,j}) := q_{i,in}$
- $\xi_{i,et}(x_{j,in}, \alpha_{i,j})$ avec $((i, j) \in I \times I$ et $i \neq j)$ est à déterminer.

Soit $q_{j,m}$ l'état de départ du procédé $G_{j,et}$. Pour déterminer l'état de départ de la spécification $E_{j,et}$ adéquat avec l'état de départ du procédé $G_{j,et}$ nous caractérisons dans un premier temps le langage désiré $K_{j,q}$ élaboré à partir de l'état $q_{j,m}$. Dans un

second temps nous recherchons l'état compatible de la spécification $E_{j,et}$.

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 $K_{j,q}$. Afin de bien appréhender la portée de notre proposition, supposons que le procédé G_j admette deux états de départ $q_{j,1}$ et $q_{j,2}$, alors à partir de l'automate G_j nous pourrions définir deux automates $G_{1,j}$ et $G_{2,j}$ ayant respectivement l'état de départ $q_{j,1}$ et $q_{j,2}$:

$G_{1,j} = (Q_j, \Sigma_j, \delta_j, q_{j,1}, Q_{j,m})$ et $G_{2,j} = (Q_j, \Sigma_j, \delta_j, q_{j,2}, Q_{j,m})$. Pour chaque état de départ et donc pour chaque automate nous définissons un langage désiré $K_{j,1}$. Par conséquent si l'état de départ du procédé G_j est $q_{j,1}$, il faut rechercher un état x_1 de la spécification E_j vérifiant

$L(G_j, q_{j,1}) \cap L(E_j, x_1) = K_{j,1}$, où $L(G_j, q_{j,1})$ est le langage du procédé dégradé ayant comme état initial l'état $q_{j,1}$. Le théorème ci-dessous présente le cadre formel de cette détermination.

Théorème 1 :

Soit q_m un état de départ du procédé G_j . Il existe un état unique x_1 de la spécification E_j tel que $L(G_j, q_{j,1}) \cap L(E_j, x_1) = K_{j,m}$.

$L(S_{j,x_1}) = K_{j,m}$, avec $K_{j,m}$ le langage désiré compatible avec q_m . Ce langage désiré $K_{j,m}$ est donné par rapport à l'ensemble des séquences d'événements $s \in L(S_i/G_i)$ tel que $\delta_i \times \xi_i((q_{i,0}, x_{i,0}), s) = q$, où $\alpha_{i,j} \in \text{suiv}(s)$ et $\delta_{j,et}(q_{j,0}, \pi_{i,j}(s)) = q_m$. \square

3.2 Etat de retour du modèle des spécifications E_i

La démarche proposée dans le théorème 1, pour déterminer les états de raccordement des spécifications, exige un calcul de tous les langages de tous les états de spécification. Ceci rend cette démarche complexe. Nous proposons donc un algorithme permettant de déterminer l'ensemble des états de raccordement d'une manière moins complexe. Avant de présenter notre proposition, nous devons tout d'abord établir un ensemble de notations qui seront utilisées dans l'algorithme.

Notations :

- $\Sigma(q)$: représente l'ensemble des événements physiquement possible à partir de l'état q du procédé;
- $\Sigma_a(x)$: représente l'ensemble des événements autorisés depuis l'état x de la spécification;
- $\text{Re}(x, E)$: est l'ensemble des états de la spécification E atteignables depuis l'état x ;
- $\text{Re}(q, G)$: est l'ensemble des états de procédé G accessibles depuis l'état q .

Proposition 3

Soit $\{q_1, q_k, \dots, q_n\}$ l'ensemble des états de départ du procédé G_i .

- Déterminer pour chaque état de départ q_i le langage désiré K_{q_i} élaboré à partir de cet état.

- Faire $H = X$. Initialement H est l'ensemble de tous les états de la spécification E_i associée au procédé G_i .

- Pour chaque q_i faire :

- Calculer $\Sigma(q_i) \cap K_{q_i} \Sigma(q_i) = \Sigma(q_i) \cap K_{q_i}$ représente l'ensemble des événements physiquement possibles depuis l'état q_i du procédé et qui appartient au langage désiré K_{q_i} .

- Pour chaque $x \in H$ faire :

- Calculer $\Sigma_a(x)$.

- Calculer $\Sigma_a(x) \cap \Sigma(q_i)$ est l'ensemble des événements physiquement possibles depuis l'état q_i et qui sont autorisés à partir de l'état de la spécification.

- Si $\Sigma(q_i) \cap \Sigma_a(x) \neq \Sigma(q_i) \cap K_{q_i}$ alors $H = H - \{x\}$. $H - \{x\}$ est l'ensemble H dépourvu de tous les états x qui ne vérifient pas la condition.

- Tant que $\text{card}(H) \neq 1$ ($\text{card}(H)$ représente le nombre d'éléments qui existent dans H) faire :

- Calculer $\text{Re}(x, E)$.

- Calculer $\text{Re}(q_i, G_i)$.

- Si quel que soit $x' \in \text{Re}(x, E)$ et quel que soit $q' \in \text{Re}(q_i, G_i)$, il existe une séquence d'événements vérifiant $\delta_i(q_i, s) = q'$ et $\xi_i(x, s) = x'$, telle que $s \Sigma(q_i) \cap K_{q_i} \neq s(\Sigma(q_i) \cap \Sigma_a(x))$ alors $H = H - \{x\}$.

- L'état x vérifiant que $\text{card}(H)=1$ est par conséquent le seul état de départ qui correspond à l'état q_i .

4. Exemple D'application

Pour illustrer l'approche proposée, nous considérons un système manufacturier composé de deux machines identiques M_1 et M_2 , d'une machine redondante M_3 et d'un stock intermédiaire S , conformément à la figure 2

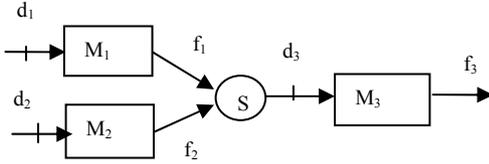


Figure 2 : Systèmes manufacturiers à trois machines et un stock intermédiaire.

Les deux machines M_1 et M_2 fonctionnent de façon indépendante, puisent des pièces brutes en amont et rejettent des pièces usinées en aval. La machine M_3 est en redondance avec la machine M_1 .

La machine M_1 puise des pièces brutes d'un stock amont de capacité supposée infinie. Cette action est modélisée par l'événement contrôlable d_1 . Après traitement par la première machine, la pièce est ensuite déposée dans le stock intermédiaire S . Ceci est modélisé par l'événement incontrôlable f_1 . Lorsque la machine M_1 est en marche, l'occurrence de l'événement p_1 conduit la machine dans un état de panne. La pièce en cours de traitement est alors supposée perdue. La réparation de la machine défaillante M_1 est modélisée par l'événement contrôlable r_1 . La deuxième machine M_2 commence son fonctionnement (événement contrôlable d_2) par le retrait d'une pièce du stock S . La pièce traitée par la seconde machine sera déposée (événement incontrôlable f_2) dans un stock aval dont la capacité est supposée également infinie. La machine M_3 est redondante avec la machine M_1 .

Afin de réduire notre étude à deux modes de fonctionnement, nous ne considérons pas les pannes des machines 2 et 3. Ceci nous permet de considérer un mode nominal dans lequel les machines 1 et 2 produisent et un mode dégradé dans lequel la machine 3 remplace la machine 1 qui est en panne. La machine M_2 est donc la ressource commune des deux modes.

Initialement, le cycle $M_1 \rightarrow M_2 \rightarrow M_1 \dots$ traite normalement les pièces brutes. C'est l'objectif pour lequel le système a été conçu (mode nominal). La commutation du mode nominal vers le mode dégradé (resp. du mode dégradé vers le mode nominal) est engendrée par l'occurrence de l'événement de commutation p_1 (resp. r_1). Le mode dégradé correspond alors à un second cycle $M_3 \rightarrow M_2 \rightarrow M_3 \dots$

Les modèles automates G_1, G_2 et G_3 respectivement des machines M_1, M_2 et M_3 sont donnés dans la figure 3. Les flèches en pointillés représentent les événements de commutation entre les modes nominal et dégradé.

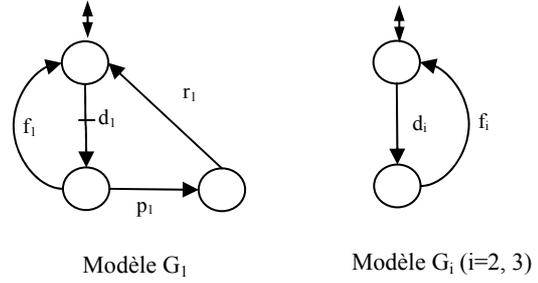


Figure 3 : Modèles automates des machines M_1, M_2 et M_3 .

L'alphabet global $\Sigma = \{d_1, d_2, d_3, f_1, f_2, f_3, p_1, r_1\}$ est scindé en trois alphabets : l'alphabet du mode nominal $\Sigma_n = \{d_1, d_2, f_1, f_2\}$, l'alphabet du mode dégradé $\Sigma_d = \{d_3, d_2, f_3, f_2\}$ et l'alphabet des événements de commutation $\Sigma' = \{p_1, r_1\}$. Ce système possède deux modes de fonctionnement, le mode nominal n et le mode dégradé d . Afin de mettre en oeuvre l'approche proposée nous exprimons pour chaque mode de fonctionnement un modèle du procédé auquel est associé un modèle de spécification associé (voir figure 4). La spécification E_n est ainsi associée au procédé G_n en mode nominal et la spécification E_d est associée au procédé G_d en mode dégradé. Ces spécifications sont écrites en fonction de l'état initial du procédé associé : le fait que le mode considéré n'est pas sélectionné au départ n'entre donc pas en compte dans l'écriture de l'automate de la spécification non étendue. Pour de plus amples informations concernant le mécanisme permettant de construire les modèles étendus des procédés dans les deux modes nominal et dégradé, le lecteur pourra se référer à [6].

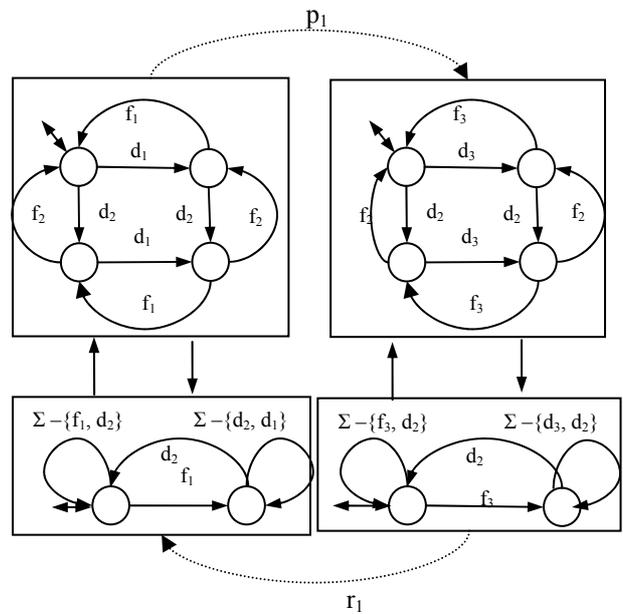


Figure 4 : Modèles non étendus du procédé et des spécifications

4.1 Construction de $E_{d,et}$

L'occurrence de l'événement de commutation p_1 conduit l'automate de la spécification $E_{n,et}$ vers l'état inactif $x_{n,in}$. En revanche, ce même événement conduit l'automate de la spécification $E_{d,et}$ depuis son état inactif $x_{d,in}$ vers un état $x \in X_d$ dit état de départ. Le modèle du procédé dans le mode dégradé peut posséder deux états de départ notés $q_{d,0}$ et $q_{d,2}$. Par conséquent ce modèle admet deux dynamiques différentes, $L(G_{d,qd,0})$ et $L(G_{d,qd,2})$ avec :

$$L(G_{d,qd,0}) = \{s \in \Sigma_d^* \mid \delta_d(q_{d,0}, s) \neq !\} = \{d_2, d_3, d_2f_2, d_3f_3, \dots\}.$$

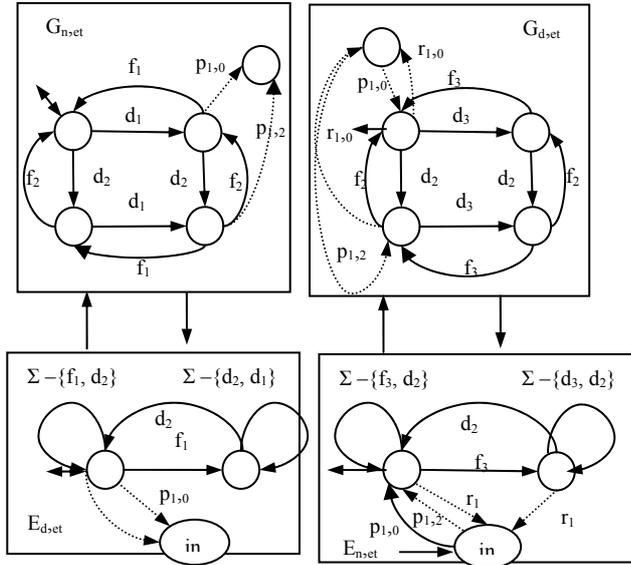
$$L(G_{d,qd,2}) = \{s \in \Sigma_d^* \mid \delta_d(q_{d,2}, s) \neq !\} = \{f_2, d_3, f_2d_2, d_3f_3, \dots\}.$$


Figure 5 : Procédés et spécifications étendus du mode nominal et dégradé.

Supposons maintenant qu'en cours du mode nominal, l'événement de panne se produise après la séquence d'événements $s=d_1$ (événement $p_{1,0}$, où $p_{1,0}$ est l'événement de panne défini à partir de p_1 par l'expression suivante $p_1 = p_{1,i}$ si $\delta_{d,et}(q_{d,in}, p_1) = q_{d,i}$). L'état de départ du modèle $G_{d,et}$ en appliquant la proposition 2 est $q_{d,0}$. Ce qui correspond à un stock vide. Le langage désiré en boucle fermée depuis cet état doit prendre en considération l'état vide du stock, tel que $K_{d,0} = \{d_3, d_3f_3, d_3f_3d_2, \dots\}$. Maintenant, nous appliquons la proposition 3 pour déterminer l'état de départ de la spécification E_d . Initialement $H = \{x_{d,0}, x_{d,1}\}$.
Pour $x_{d,0} \in H$ faire :

Calculons $\Sigma_a(x_{d,0}), \Sigma_a(x_{d,0}) = \{d_3, f_2, f_3\}$,
Calculons $\Sigma_a(x_{d,0}) \cap \Sigma(q_{d,0}), \Sigma_a(x_{d,0}) \cap \Sigma(q_{d,0}) = \{d_3\}$
Calculons $K_{d,0} \cap \Sigma(q_{d,0}), K_{d,0} \cap \Sigma(q_{d,0}) = \{d_3\}$
Test $\Sigma_a(x_{d,0}) \cap \Sigma(q_{d,0}) = K_{d,0} \cap \Sigma(q_{d,0})$ $H=H$.

Pour $x_{d,1} \in H$ faire :

Calculons $\Sigma_a(x_{d,1}), \Sigma_a(x_{d,1}) = \{d_2, f_2, f_3\}$,
Calculons $\Sigma_a(x_{d,1}) \cap \Sigma(q_{d,0}), \Sigma_a(x_{d,1}) \cap \Sigma(q_{d,0}) = \{d_2\}$

Calculons $K_{d,0} \cap \Sigma(q_{d,0}), K_{d,0} \cap \Sigma(q_{d,0}) = \{d_3\}$

Test $\Sigma_a(x_{d,1}) \cap \Sigma(q_{d,0}) \neq K_{d,0} \cap \Sigma(q_{d,0})$ $H=H - \{x_{d,1}\} = x_{d,0}$.

Ainsi, la solution de l'équation $L(S_{d,x_j}) \cap L(G_{d,qd,0}) = K_{d,0}$ est $x_j = x_{d,0}$.

De façon similaire, on peut montrer qu'à l'occurrence de l'événement de commutation $p_{1,2}$ depuis l'état $q_{n,3}$ en mode nominal, l'état de départ de la spécification étendue du mode dégradé est $x_{d,0}$. Le modèle global étendu de la spécification du mode dégradé est donné par la figure 5. Sur cette figure, le modèle étendu du mode nominal ne comporte que les transitions évidentes menant à l'état inactif. Les autres transitions sont déterminées dans la section suivante. Le procédé sous contrôle étendu du mode dégradé, obtenu par synthèse, est présenté figure 6. Il contient 7 états et 12 transitions.

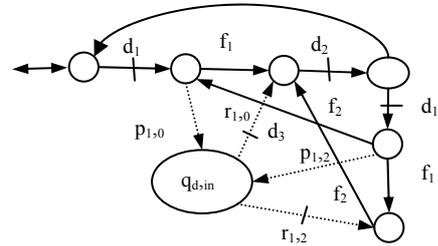


Figure 6 : Contrôleur étendu du mode dégradé.

4.2 Construction de $E_{n,et}$

De la même façon, on peut montrer que l'occurrence de l'événement de commutation r_1 conduit l'automate de la spécification $E_{n,et}$ depuis son état inactif $x_{n,in}$ vers l'état $x_{n,1}$ (voir figure 7).

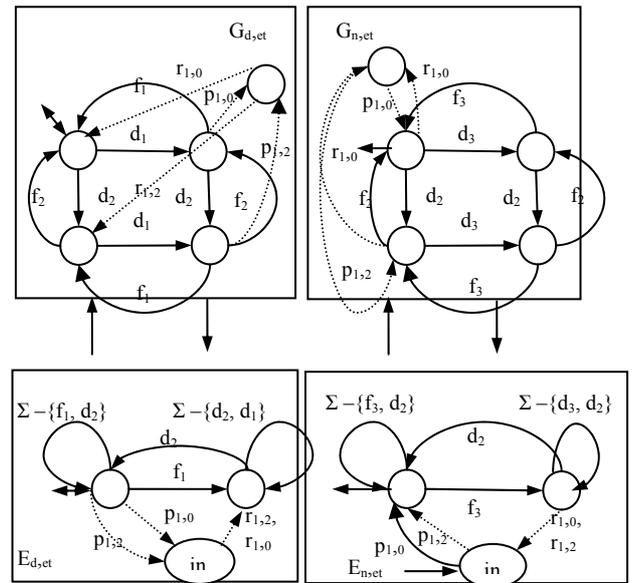


Figure 7 : Modèles étendus du mode nominal et dégradé

Le contrôleur étendu du mode nominal est donné par la figure 8. Il comporte 7 états et 12 transitions.

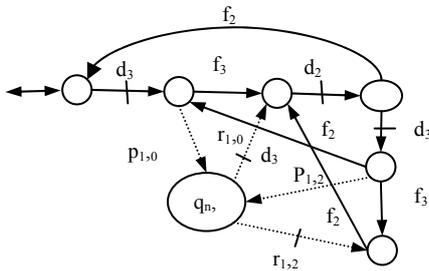


Figure 8 : Contrôleur étendu du mode nominal.

5. Conclusion

Cet article nous a permis de présenter l'extension de la théorie de contrôle par supervision par la prise en compte de différents modes de fonctionnement. A partir d'un exemple simple nous avons montré l'utilisation de différents modèles du procédé et proposé un mécanisme systématique de détermination de l'état départ/retour d'un modèle. Nous avons également justifié et présenté des modèles étendus permettant la prise en compte de différents états de départ possibles dans un mode donné, évitant ainsi de construire autant de modèles que d'états de départ/retour possibles. Nous avons aussi étendu les modèles des spécifications de chaque mode de fonctionnement afin de prendre en compte le mécanisme de commutation entre les modèles des procédés et ceux des spécifications. Les contrôleurs étendus du mode nominal et dégradé sont contrôlables et non-bloquants.

BIBLIOGRAPHIE.

1. P. Ramadge, W. Wonham. *The control of Discrete Event Systems*, In Proc. of the IECE, Vol. 77, p. 81-98. 1987.
2. W. M. Wonham. *Notes on control of discrete-event systems*, notes de cours, department of Electrical and Computer Engineering, University of Toronto,

<http://www.control.toronto.edu/people/profs/wonham/>, 2002.

3. F. Lin, W. M. Wonham. *Decentralized supervisory control of discrete-event systems*, Information Sciences, Vol. 41, no. 2, pp. 199-224, 1988.
4. G. C. Cassandras, S. Lafortune. *Introduction to Discrete Event Systems*, 1st edition. Boston : Kluwer Academic Publishers, pp. 822, 1999.
5. K. Rudie, W. M. Wonham. *Think globally, act locally : Decentralized supervisory control*, IEEE transaction on automatic and Control, 37(11):1692-1708, November 1992.
6. O. Kamach. *Approche Multi-Modèle pour les Systèmes à Événements Discrets : application à la gestion des modes de fonctionnement*. Thèse de Doctorat : Institut National des Sciences Appliquées de Lyon, p 177, France, 2004.
7. M. Nourelfath. *Extension de la théorie de la supervision à la surveillance et à la commande des systèmes à événements discrets : application à la sécurité opérationnelle des systèmes de production*, Thèse de Doctorat : Institut National des Sciences Appliquées de Lyon, p 145, France, 1997.
8. N. Dangoumau. *Contribution à la gestion des modes des systèmes automatisés de production*. Thèse de Doctorat : Université des Sciences et Technologiques de Lille, p 181, France 2000.