

**Optimisation de la méthode de synthèse de commande  
par supervision des systèmes à événements discrets  
basée sur les invariants de marquage**

Mohaman Gonza, Laurent Bitjoka

► **To cite this version:**

Mohaman Gonza, Laurent Bitjoka. Optimisation de la méthode de synthèse de commande par supervision des systèmes à événements discrets basée sur les invariants de marquage. 2017. <hal-01590999>

**HAL Id: hal-01590999**

**<https://hal.inria.fr/hal-01590999>**

Submitted on 20 Sep 2017

**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.



---

# Optimisation de la méthode de synthèse de commande par supervision des systèmes à événements discrets basée sur les invariants de marquage

Mohaman GONZA, Laurent BITJOKA

LESIA (Laboratoire d'Energie, Signal, Imagerie et Automatique), ENSAI, Université de N'Gaoundéré, Cameroun

[mohaman.gonza@univ-ndere.cm](mailto:mohaman.gonza@univ-ndere.cm), [lbitjoka@univ-ndere.cm](mailto:lbitjoka@univ-ndere.cm)

---

*RESUME*— La théorie de commande par supervision des systèmes à événements discrets (SED) a été initialement développée à base des automates à états finis (AFD), avec pour objet de garantir à priori un fonctionnement conforme aux spécifications qui se déclinent généralement en états interdits. Les automates représentent un puissant formalisme de modélisation des SED, malgré leur grande sensibilité au problème d'explosion combinatoire du nombre d'états. Cet inconvénient est minimisé lorsque la modélisation est effectuée par les réseaux de Petri (RdP). Cependant, cet avantage en concision est contrebalancé par la perte de l'optimalité des résultats fournis par les automates. Toutefois, parmi de nombreuses méthodes de synthèse de commande par supervision à base RdP la méthode des invariants est la plus utilisée pour le calcul du contrôleur. Malheureusement, cette approche ne garantit pas en général une solution optimale si elle est utilisée de manière classique, et surtout dans le cas où la synchronisation des RdP du SED est réalisée par des transitions incontrôlables. La solution consiste à identifier les contraintes admissibles adéquates à la synthèse du contrôleur maximal permissif (optimal). Nous avons proposé une approche fusionnelle basée sur l'Algorithme de Kumar afin de déterminer les contraintes liées aux états interdits frontières, pour le calcul du contrôleur. Mais, la nécessité d'explorer l'espace d'états du RdP, à savoir le graphe de marquage nous a conduit à représenter le graphe de marquage par sa fonction de transition, à savoir, la matrice de transition. La codification des éléments de la matrice, selon la spécification du SED nous a permis d'identifier les états constituant l'hyperplan (fonction de décision) et les états interdits frontières adéquats.

*ABSTRACT*— The discrete event system (DES) control theory was initially developed using finite state automata, with the aim of guaranteeing a priori operation in conformity with the specifications which are generally decline as forbidden states. The automata represent a powerful modeling formalism of the DESs, despite their great sensitivity to the combinatorial

*explosion problem of the number of states. This disadvantage is minimized when the modeling is carried out by the Petri nets (PN). However, this advantage in concision is counterbalanced by the loss of the optimality of the results provided by the automata. However, among many methods of synthesis of supervisory control by PN, the invariants method is the most used for the However, among many methods of synthesis of supervisory control by PN, the invariants method is the most used for computing the controller. Unfortunately, this approach does not generally guarantee an optimal solution if it is used in a conventional way, and especially in the case where the synchronization of the PNs of the DES is carried out by uncontrollable transitions. The solution consists in identifying the admissible constraints suitable for the synthesis of the maximum permissive (optimal) controller. We proposed a fusion approach based on the Kumar algorithm to determine the constraints related to the forbidden boundary states, for computing the controller. But the need to explore the state space of the PN, namely the marking graph, led us to represent the marking graph by its transition function, namely, the transition matrix. The coding of the elements of the matrix, according to the DES specification enabled us to identify the states constituting the hyperplane (decision function) and the appropriate boundary states*

*MOTS CLES : Commande par Supervision, Système à Evénements Discrets, Réseau de Petri , Automates à états finis, Graphe de marquage, Matrice de transition*

*KEYWORDS: Supervisory Control, Discret Event Systems, Petri Net, Finite Automaton, reachability marking graph, Transition matrix*

---

## **1. Introduction**

Le besoin de méthodes formelles et d'outils puissants capables de synthétiser les lois de contrôle des systèmes d'événements discrets (SED) a donné naissance à la théorie de commande par supervision basée sur l'utilisation d'automates et de langues à états finis (Ramadge et Wonham, 1983 ; Cassandras et Lafortune, 2008 ; Komenda *et al.*, 2009). Cependant, le manque de structure dans les automates et l'explosion combinatoire de l'espace d'états, induit par les opérations de composition (Kumar, 1991), limitent le développement des méthodes de synthèse simples et efficaces, applicables dans l'industrie (Uzam et Wonham, 2006). Afin de profiter de la structure des SED, deux solutions ont été développées: (i) l'utilisation de structures de contrôle décentralisées, modulaires et hiérarchiques (deQueiroz et Cury, 2002 ; Yoo et Lafortune, 2002 ; Wonham, 2003), (ii) l'utilisation de formalismes de modélisation autres que les automates à états finis (Moody *et al.*, 1998). En ce qui concerne les formalismes de modélisation, les réseaux de Petri (RdP) ont une grande similitude avec les automates puisque ces derniers représentent explicitement les fonctions de transition des SED (Zhou et DiCesare, 1993 ; David et Alla, 2010) et une large classe de langues (Wonham, 2011 ; Giua et Seatzu, 2007). Dans ce domaine de recherche, plusieurs approches formelles bien fondées de synthèse des contrôleurs ont été développées. Par exemple, la théorie des

régions (Badouel *et al.*, 1995, Ghaffari *et al.*, 2003), la théorie de Holloway et Krogh (Holloday et Krogh, 1990), l'approche basée sur le RdP colorée et synchronisée (Godon, 1996). Cependant, elles ne sont pas optimales et/ou générales en raison de l'exploration fastidieuse de l'espace d'état accessible ou de la complexité de l'algorithme de calcul du contrôleur. Une autre approche, basée sur la théorie des invariants de marquage (Yamalidou *et al.*, 1996), est simple et efficace pour calculer un contrôleur optimal si l'ensemble adéquat des contraintes admissibles lui est fourni (Kattan, 2004 ; Vasiliu 2011). Le rôle du contrôleur est d'imposer le respect des contraintes définies par la spécification, en permettant / interdisant l'occurrence d'événements selon le langage "admissible" pour le SED. Toutefois, l'existence d'événements incontrôlables, que le contrôleur ne peut interdire, pose un véritable problème de contrôlabilité (Gaudin et Marchand, 2005).

Notre travail consiste à explorer cette méthode mathématique afin de proposer une méthode formelle telle que, le passage de la synthèse à l'implantation soit systématique (Nourelfath et Niel, 2000). Pour atteindre notre objectif, nous considérerons la classe de réseau de Petri synchronisé (Section 2) qui permet d'exprimer une spécification sous forme d'états interdits ou de séquences d'événements (Boel *et al.*, 1995). En premier lieu, nous proposons une approche fusionnelle (Section 3) qui associe le concept d'invariants de marquages du réseau de Petri (Yamalidou *et al.*, 1996) à la puissance de calcul des automates à états finis (Hopcroft *et al.*, 2007). Cette approche nous permet de déterminer les états interdits frontières, nécessaires et suffisants pour le calcul du contrôleur (Vasiliu *et al.*, 2009). Le problème de cette approche est la nécessité d'explorer l'espace d'états (graphe de marquages) du SED. En second lieu, nous cherchons à résoudre ce problème, en proposant une technique de séparation des états du graphe de marquages du RdP du SED en boucle fermée (Section 4). La séparation est effectuée par un ensemble d'états constituant l'hyperplan de séparation (Ehrenfeucht et Rozenberg, 1990 ; Falk *et al.*, 2011). Cette solution simple est partiellement structurelle à cause de la construction du graphe de marquages. Cependant, elle permet de comprendre que les états interdits peuvent être générés si la synchronisation des RdP est réalisée par les transitions associées aux événements incontrôlables (Dideban et Alla, 2008).

## **2. Commande par supervision des SED**

### ***2.1. Outils de modélisation***

Un système à événements discrets (SED) est un système dynamique dont les variables d'état évoluent spontanément d'un état à l'autre, en fonction de l'occurrence d'événements instantanés (Seatzu *et al.*, 2012). Les outils de modélisation les plus utilisés dans la synthèse de commande par supervision des SED sont les automates à états finis et les réseaux de Petri « synchronisés » (Hopcroft *et al.*, 2007).

### 2.1.1. Automate à états finis et langage généré

**Définition 1 :** Un automate à états finis  $A$  est un 5-uplet  $A = (Q, \Sigma, \delta, q_0, Q_m)$ , où :  $Q$  est l'ensemble fini d'états ;  $\Sigma$  est un ensemble fini événements, appelé alphabet,  $\delta$  est la fonction de transition d'états de  $Q \times \Sigma \rightarrow Q$ ,  $q_0 \in Q$  est l'état initial ;  $Q_m \subseteq Q$  est l'ensemble d'états marqués. Le comportement d'un automate est caractérisé par des séquences de mot ou chaînes de mot  $\sigma$ , permettant de définir les langages selon que l'on implique les états marqués ou non (Seatzu *et al.*, 2012) :

Le langage généré  $L(A)$  qui décrit l'ensemble de chaînes  $\sigma$  qui permettent de rejoindre un état de l'automate à partir de l'état initial,

$$L(A) = \{\sigma \in \Sigma^* \mid \delta(q_0, \sigma) \in Q\}, \text{ avec } \Sigma^* \text{ est la fermeture de Kleene} \quad (1)$$

Langage marqué  $L_m(A)$  qui décrit l'ensemble de chaînes  $\sigma$  qui permettent de rejoindre un état marqué à partir de l'état initial,

$$L_m(A) = \{\sigma \in \Sigma^* \mid \delta(q_0, \sigma) \in Q_m\} \quad (2)$$

### 2.1.2. Réseau de Petri synchronisé et langage généré

Il est commode de modéliser les SED par la classe de réseau de Petri « synchronisé » (Peterson, 1981) c'est-à-dire comme un 4-uplet  $R = (N, \ell, M_0, F)$  avec

- $N = (P, T, W, W^+)$  la structure d'un RdP ;
- $\ell : T \rightarrow E$  est la fonction d'étiquetage qui assigne à chaque transition l'étiquette à partir de l'alphabet des événements  $E \leftrightarrow \Sigma$  ;
- $M_0$  est le marquage initial ;
- $F$  est l'ensemble des marquages finaux

Le réseau de Petri « synchronisé » est relié à automate à états finis dans le sens qu'ils représentent explicitement la fonction de transition et offre la possibilité de conditionner le franchissement d'une transition à l'occurrence d'un événement (Pocci *et al.*, 2011, Dorsaf *et al.*, 2012).

**Définition 2.** Un réseau de Petri synchronisé (RdP-S) défini sur l'alphabet  $E$  est une structure  $R = (P, T, E, W, W^+, \ell, M_0)$  où :  $P$  est un ensemble fini de place ;  $T$  est un ensemble fini de transitions ;  $E$  est l'ensemble d'événements associés aux transitions ;  $W$  et  $W^+$  les matrices d'incidence de  $N$  indexé par  $P \times T \rightarrow \mathbb{N}$  ;  $\ell : T \rightarrow E$  est la fonction d'étiquetage de transitions. Une séquence de transitions est  $\sigma = t_1 t_2 \dots t_m \in T^*$  où  $\ell(\sigma) = \ell(t_1) \ell(t_2) \dots \ell(t_m)$  ;  $M_0 : P \rightarrow \mathbb{N}$  est le marquage initial.

Dans la suite, le RdP-S est simplement désigné par  $R = (N, E)$  avec  $N = (P, T, W, W^+, \ell, M_0)$ .

L'état d'un SED modélisé par un RdP-S est déterminé par le marquage  $M_k : P \rightarrow \mathbb{N}$  des places à un instant donné.

$$M_k = [M_k(P_1), M_k(P_2), \dots, M_k(P_n)] \text{ où } n \text{ est le nombre de places.} \quad (3)$$

A partir d'un marquage  $M_i$ , on peut atteindre un marquage  $M_k$  via une séquence de franchissement des transitions  $\sigma$ . Cette dynamique, dénoté  $M_i[\sigma]M_k$ , peut être calculée par l'équation fondamentale

$$M_k = M_i + W\sigma^T, \text{ avec } W = W^+ - W^- \quad (4)$$

L'ensemble de marquages accessibles tient lieu d'espace d'états  $A(R, M_0)$  et peut être représenté sous forme de graphe des marquages accessibles, s'il est fini (Olive, 2011).

$$A(R, M_0) = \{M_i \mid \exists \sigma \in E^* \text{ tel que } M_0[\sigma]M_i\} \quad (5)$$

**Définition 3.** Le graphe de marquage accessible est un 4-uplet  $G = \{M, E, \delta, M_0\}$  où  $M = A(R, M_0)$ , est l'ensemble (fini) des états ;  $E$  est l'ensemble des événements associés aux transitions ;  $\delta : M \times E \rightarrow M$  est la fonction de transition d'état ;  $M_0 \in M$  est l'état initial.

#### ❖ Produit synchrone des RdP

Soient  $R_1 = (N_1, E_1)$ ,  $N_1 = (P_1, T_1, W_1, W_1^+, \ell_1, M_{10})$ , défini sur  $E_1$  et  $R_2 = (N_2, E_2)$ ,  $N_2 = (P_2, T_2, W_2, W_2^+, \ell_2, M_{20})$ , défini sur  $E_2$ . Le produit synchrone of  $R_1$  et  $R_2$ , désigné  $R = R_1 \parallel R_2$ , est défini sur l'alphabet  $E$ , tel que :  $P = P_1 \cup P_2$  ;  $T = T_1 \cup T_2 - T_{12}$  avec  $T_{12} := \{t_i \in T_1 \mid \exists t_j \in T_2 \text{ tel que } \ell(t_i) = \ell(t_j)\}$  ;  $W(p, t) := \{W_1(p, t) \text{ si } p \in P_1 \text{ ou } W_2(p, t) \text{ si } p \in P_2\}$  ;  $W^+(p, t) := \{W_1^+(p, t) \text{ si } p \in P_1 \text{ ou } W_2^+(p, t) \text{ si } p \in P_2\}$  ;  $M_0(p) := \{M_{10}(p) \text{ si } p \in P_1 \text{ ou } M_{20}(p) \text{ si } p \in P_2\}$  ;  $\ell := T \rightarrow E$  est défini pour tout  $t \in T$  par

$$\ell(t) = \begin{cases} \ell_1(t) & \text{si } t \in T_1, \ell_1(t) \notin E_1 \cap E_2 \\ \ell_2(t) & \text{si } t \in T_2, \ell_2(t) \notin E_1 \cap E_2 \\ \ell_1(t) = \ell_2(t) & \text{si } t \in T_{12}, \ell(t) \in E_1 \cap E_2 \end{cases}$$

#### ❖ Langages générés

Un RdP-S peut définir des langues générés et acceptés avec une expression régulière (Hopcroft *et al.*, 2007; Sogbohossou et Vianou, 2016).

**Définition 4.** Soit  $L(R)$  le langage généré par  $R = (N, E)$  sur l'alphabet  $E$ , alors il existe un graphe de marquage  $G$  avec un langage  $L(G)$ , tel que :

$$L(R) = L(G) = \{\sigma \in E^* \mid M_0[\sigma]M_k\}, \text{ avec } \sigma \in T^* \rightarrow E^* \quad (6)$$

**Définition 5.** Considérons le RdP-S  $R$  qui reconnaît le langage  $L(R)$ . Soit  $F$  l'ensemble d'état final, qui correspond à la réalisation d'un certain objectif. Le langage accepté  $L_m(R)$  est

$$L_m(R) = \{\sigma \in E^* \mid \exists (M_k \in F); M_0[\sigma] M_k\} \Rightarrow L_m(R) \subseteq L(R). \quad (7)$$

Le langage généré par  $R$  à partir de tout état  $M_k$  est

$$L(R, M_k) = \{\sigma \in E^* \mid \exists M_k \text{ et } \sigma \in L(R)\} \quad (8)$$

Le produit synchrone est généralement défini sur les langages et nécessite les opérations de projection et de projection inverse (Seatzu *et al.*, 2012).

**Définition 6.** Soient  $E_1$  et  $E_2$  deux alphabets. Nous définissons la projection naturelle  $P_i: (E_1 \cup E_2)^* \rightarrow E_i^*$ ,  $i = 1, 2$  et l'application inverse  $P_i^{-1}: E_i^* \rightarrow (E_1 \cup E_2)^*$ ,  $i = 1, 2$  par:

$$P_i(\varepsilon) := \varepsilon \text{ et } P_i(\sigma) := \begin{cases} \sigma & \text{si } \sigma \in E_i, \\ \varepsilon & \text{si } \sigma \notin E_i, \end{cases} \quad (9)$$

Nous désignons à présent  $\sigma \in E^*$  comme une séquence générée par  $R_1 \parallel R_2$ , les projections de  $\sigma$  sur les alphabets  $E_1$  et  $E_2$  seront les séquences  $\sigma_1 \in E_1^*$  et  $\sigma_2 \in E_2^*$ . Soient  $\sigma_1 \in L(R_1)$  et  $\sigma_2 \in L(R_2)$  les langages générés par  $R_1$  et  $R_2$ . Soit  $L(R_1 \parallel R_2) \subseteq E^*$  le langage généré par le produit synchrone de  $R_1$  et  $R_2$ . Pour toute séquence  $\sigma \in L(R_1 \parallel R_2)$ ,

$$L(R_1 \parallel R_2) = \{\sigma \in E^* \mid P_1(\sigma) := \sigma_1 \in L(R_1), P_2(\sigma) := \sigma_2 \in L(R_2)\} \quad (10)$$

**Définition 7.** Le produit synchrone de deux langages  $L(R_1) \subseteq E_1^*$  et  $L(R_2) \subseteq E_2^*$  est

$$L(R_1) \parallel L(R_2) = \{\sigma \in E^* \mid P_1(\sigma) := \sigma_1 \in L(R_1), P_2(\sigma) := \sigma_2 \in L(R_2)\} = L(R_1) \parallel L(R_2). \quad (12)$$

Si les alphabets des deux langages  $L(R_1)$  et  $L(R_2)$  sont les mêmes

$$L(R_1 \parallel R_2) = \{\sigma \in E^* \mid P_1(\sigma) := \sigma \in L(R_1), P_2(\sigma) := \sigma \in L(R_2)\} = L(R_1) \cap L(R_2). \quad (13)$$

En somme, le produit synchrone permet d'obtenir le modèle de commande du SED en boucle fermée et de minimiser le phénomène d'explosion combinatoire par entrelacement des événements concurrents.

## 2.2. Principe de la commande par supervision

L'objectif de la commande par supervision défini par Ramadge et Wonham (Ramadge et Wonham, 1989) est de synthétiser un superviseur (contrôleur) à base d'automates tel que, le fonctionnement du SED couplé à ce dernier (figure 1)

respecte les spécifications en se basant sur le concept de contrôlabilité (Cassandras et Lafortune, 2008).

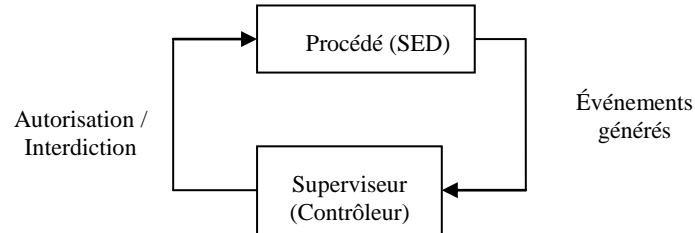


Figure 1 : Schéma de principe de la commande par supervision

**Définition 8.** Un langage de spécification  $K$  est dit contrôlable par rapport à un langage  $L(R)$  si :  $K \Sigma_u \cap L(P) \subseteq K$

Où  $\Sigma_u$  représente les événements incontrôlables ;  $K$  le préfixe-clôture et  $L(R)$  le langage du SED.

La théorie de Ramadge et Wonham permet de déterminer directement le langage contrôlable maximal permissif à partir de l’algorithme de Kumar (Kumar, 1991).

Soit  $P$  et  $S_{spec}$  les modèles automates du procédé (SED) et de la spécification, l’algorithme de Kumar permet de vérifier la contrôlabilité du langage de spécification  $L(S_{spec})$  et de synthétiser un modèle automate du langage suprême contrôlable du fonctionnement désiré  $supC(L_D)$  ; lorsque  $L(S_{spec})$  n’est pas contrôlable. L’algorithme est basé sur les 4 pas suivants:

**Pas 1.** On construit le composé synchrone  $D$  de  $P$  et de  $S_{spec}$ , c’est-à-dire,  $D = P \parallel_{S_{spec}}$ . Son langage est noté  $L_D$ .

**Pas 2.** On détermine l’ensemble des *états interdits* : Tout état de  $D$  tel qu’il existe un événement incontrôlable définie dans  $P$  mais non définie dans  $S_{spec}$ .

**Pas 3.** On détermine l’ensemble des *états faiblement interdits* : Tout état de  $D$  qui n’est pas un état interdit et tel qu’il existe une séquence événements incontrôlables qui conduit à un état interdit  $D$ .

**Pas 4.** On supprime de  $D$  l’ensemble des états interdits ainsi que l’ensemble des états faiblement interdits et l’ensemble des états non accessibles depuis l’état initial.

### 3. Méthode des invariants de marquages

Parmi les méthodes de synthèse par RdP, la méthode des invariants de marquages est la plus simple et efficace lorsque l’ensemble adéquat des contraintes



admissibles lui est fourni. L' invariant de marquages d'un RdP est une propriété structurelle qui est déterminé par les vecteurs  $X$ , calculés à partir de :

$$X^T W = 0, \text{ où } X \text{ est appelé P-semi-flot} \quad (14)$$

L'invariant de marquage déduit de tout P-semi-flot  $X$ , tel que  $X^T = [l_1, \dots, l_i, \dots, l_n, 1]$  est donné par la relation :

$$X^T M = X^T M_0 \quad (15)$$

La méthode permet de calculer un contrôleur à base des RdP (Iordache et Antsaklis, 2006 ; Yamalidou *et al.*, 1996) . Soit un RdP de commande du SED obtenu par synchronisation structurelle du RdP du procédé avec le RdP de la spécification. Le rôle du contrôleur est de forcer le SED à respecter des contraintes de type :

$$\sum_{i=1}^n l_i M(P_i) \leq \beta \text{ où } l_i, \beta \text{ sont des nombres constants entiers positifs.} \quad (16)$$

L'inégalité de ce type peut être transformée en une égalité en ajoutant une variable positive et entière  $M(P_C)$ , et la contrainte devient :

$$\sum_{i=1}^n l_i M(P_i) + M(P_C) = \beta \quad (17)$$

La structure contrôleur, dénoté  $C$ , sera calculée en observant que l'introduction de la variable  $M(P_C)$  introduit un invariant de marquage. Le nombre de places du RdP du contrôleur est égale au nombre des contraintes imposées. La matrice d'incidence du SED sous contrôle  $W$  est composée de deux matrices :  $W_R$  et  $W_C$ . Les arcs reliant les places du contrôleur avec les transitions du RdP de commande du SED,  $R$  sont calculés par l'équation des invariants de marquages (eq. 10). Les contraintes sont écrites sous forme matricielle :

$$L M_R + M_C = b \quad (18)$$

Et l'invariant de marquage doit satisfaire :

$$X^T W = [L \quad I] \begin{bmatrix} W_R \\ W_C \end{bmatrix} = 0 \Rightarrow L W_R + W_C = 0$$

$$W_C = -L W_R \quad (19)$$

Le marquage initial du RdP du contrôleur  $M_{C0}$  peut être calculé à partir de l'équation :

$$L M_{R0} + M_{C0} = b \Rightarrow M_{C0} = b - L M_{R0} \quad (20)$$

**Exemple:**

Nous considérons un système composé de deux machines et d'un robot (Figure 1). La machine M1 fabrique des pièces de type I et la machine M2 fabrique des pièces de type II. Lorsque la machine Mi a fini son usinage (événement incontrôlable  $f\_USi$ ), elle peut être libre. Le robot transporte la pièce brute du stock vers la machine libre suivant l'ordre de fabrication (événement contrôlable  $of\_Pi$ ). Et lorsque la pièce est sur la machine Mi (événement incontrôlable  $Ps\_Mi$ ), après la fin du transport (événement incontrôlable  $f\_Tri$ ), le robot revient à son état initial. Les spécifications sont imposées par le robot.

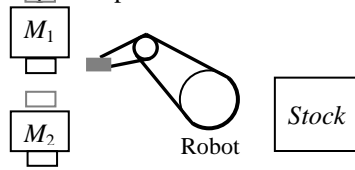
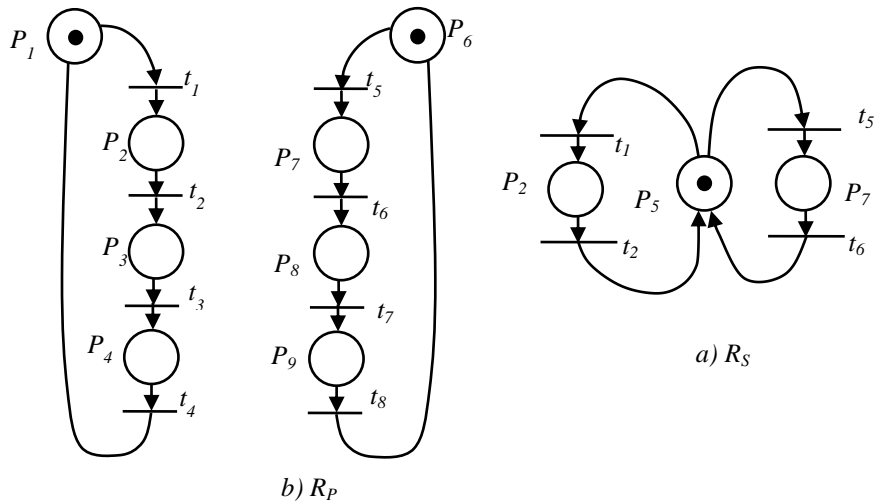


Figure 2. Système manufacturier

La figure suivante donne les modèles RdP avant leur produit synchrone (synchronisation structurelle)



$P_1$	$M1\_libre$	$t_1$	$of\_P1$
$P_2$	$TRI (Transport Robot-M1)$	$t_2$	$f\_TRI$
$P_3$	$Pièce sur M1$	$t_3$	$Ps\_M1$

$P_4$	US1 (usinage M1)	$t_4$	$f\_US1$
$P_5$	Robot libre	$t_5$	$of\_P2$
$P_6$	M2_libre	$t_6$	$f\_TR2$
$P_7$	TR2 (Transport Robot-M1)	$t_7$	$Ps\_M1$
$P_8$	Pièce sur M2	$t_8$	$f\_US2$
$P_9$	US1 (usinage M2)		

Figure 3. Modèles RdP ( $R_p$  et  $R_s$ ) du SED avant synchronisation

Supposons la spécification suivante : Le robot est entrain de transporter la pièce vers la machine M1 (marquage de la place  $P_2$ ), et la machine M2 a déjà démarré et soit arrivé à la fin de l'usinage (marquage de la place  $P_9$ ). Dans cet état, il est nécessaire que le robot vienne pour transporter une nouvelle pièce vers M2. Comme ce n'est pas le cas, la production est ralentie. C'est donc un état interdit qui ne doit jamais être atteint.

Il se traduit par la contrainte :  $M(P_2) + M(P_9) \leq 1$

Le vecteur de contrainte associé est :  $L = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$

La matrice d'incidence du fonctionnement désiré système en boucle fermée est :

$$W_R = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Ainsi, la matrice incidence du contrôleur est :

$$W_C = -LW_R = [-1 \ 1 \ 0 \ 0 \ 0 \ 0 \ -1 \ 1]$$

Le marquage initial du SED est :  $M_{R0} = [1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0]^T$  et  $b = [1]$

Donc, le marquage initial du contrôleur est :  $M_{C0} = b - LM_{R0} = [1]$

La place de contrôle ont pour transitions amonts les transitions de signe + dans  $W_C$  et transitions avals les transitions de signe - dans  $W_C$  (Moody et Antsaklis, 1998).

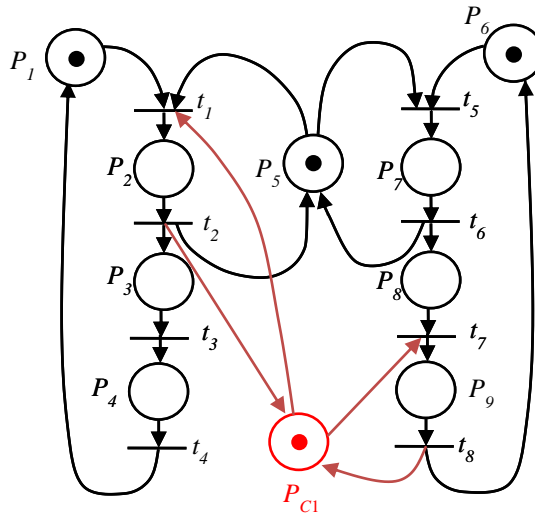


Figure 4. Modèle RdP de fonctionnement désiré du SED en boucle fermée

Cette solution simple n'est pas maximal permmissif (optimal) car elle ne prend pas en compte les états faiblement interdits défini par l'algorithme de Kumar (Kumar, 1991) ou les interdits supplémentaires qui peuvent être générés si la synchronisation structurelle des RdP est réalisée via les transitions (événements) incontrôlables (Dideban et Alla, 2008). Par ailleurs les situations de blocage ou de contrôlabilité ne sont pas évaluées lorsque que cette méthode est utilisée de cette manière.

#### 4. Contrôlabilité du RdP de commande du SED en boucle fermée

Le comportement du procédé en isolation ne respecte pas la spécification désirable pour l'exploitation. C'est pourquoi, il faut réaliser la produit synchrone des modèles  $R_P$  et  $R_S$  pour obtenir le modèle  $R = R_P || R_S$  de commande du SED en boucle fermée.

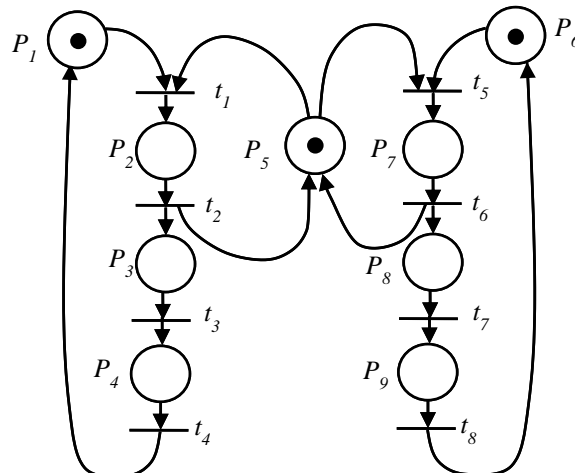


Figure 5. Modèle  $R = R_p || R_S$  de commande du SED en boucle fermée

Il est possible de déterminer si le modèle de commande  $R = R_p || R_S$  est non bloquant et contrôlable. Pour cela nous devons construire son graphe de marquages accessibles.

Le graphe de marquage accessible présenté à la figure 6 montre immédiatement que le modèle de commande du SED de la figure 5 est bloquant et incontrôlable.

En effet, l'état  $M_2$  est bloquant parce que à partir de lui, il est impossible d'atteindre l'état initial  $M_0$ . L'état  $M_1$  est dangereux car  $M(P_6) = 1$ , c'est-à-dire, la transition  $t_5$  est permise dans  $R_1$ , alors que  $M(P_5) = 0$ , c'est-à-dire, la transition  $t_5$  n'est pas permise dans  $R_2$ . Si  $t_5$  est incontrôlable alors la situation ne permet pas de respecter la spécification (l'état  $M_{11}$  ( $P_2P_9$ ) qui ne doit jamais être atteint) et génère des états interdits supplémentaires qu'il faut identifier.

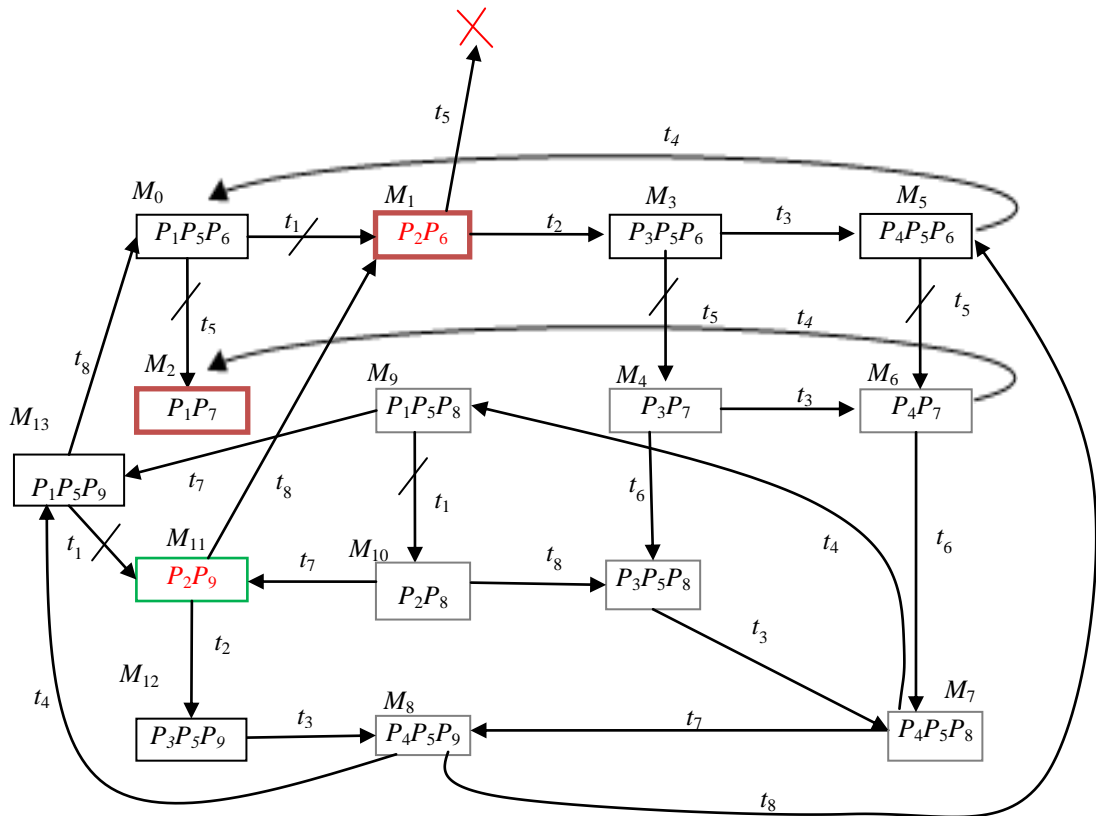


Figure 6 : Graphe de marquage accessible du système manufacturier

### 5. Détermination des contraintes admissibles par l'algorithme de Kumar

Soient  $R_P = (N_P, E_P)$ ,  $N_P = (P_P, T_P, W_P, \ell_P, M_{P0})$  et  $R_S = (N_S, E_S)$ ,  $N_S = (P_S, T_S, W_S, \ell_S, M_{S0})$  les RdP du procédé et de la spécification d'un SED. Le modèle en boucle fermée du SED à contrôler est obtenu en réalisant le produit synchrone (Dideban et Alla, 2008) des deux :  $R = R_P || R_S$ . A cause de la synchronisation avec des événements incontrôlables  $E_u$ , le procédé ne peut pas respecter la spécification ; il apparait donc un problème d'états interdits qu'il faut distinguer des états autorisés dans le graphe de marquages.

**Définition 9.** Soit  $T_u \in T$  l'ensemble des transitions associées aux évènements incontrôlables de  $R = R_P || R_S$ , un état  $M \in A(R, M_0)$  est interdit s'il existe une transition incontrôlable qui est permise par  $M$  dans  $R_P$ , mais non permise par  $M$  dans  $R_S$ . Le modèle de commande  $R$  est admissible si aucun état interdit n'est accessible.

Soient  $M_P$  et  $M_S$  les marquages des places du procédé  $R_P$  et ceux de la spécification  $R_S$  ; l'ensemble des états interdits  $M_I$  est ainsi défini :

$$M_I = \{(M_P, M_S) \mid \exists t \in T_u, \forall P_i \in R_P, M(P_i) = k \text{ et } \exists P_j \in R_S, M(P_j) = n ; k > n\} \quad (21)$$

**Définition 12.** Soit  $M_A$  l'ensemble des états autorisés par la spécification, l'ensemble des états faiblement interdits  $M_F$  sera ainsi défini :

$$M_F = \{M_i \mid M_i \in M_A, M_j \in M_U \text{ et } \sigma \in E_u, M_i[\sigma]M_j\} \quad (22)$$

A partir des ensembles  $M_F$  et  $M_I$ , on peut construire deux autres ensembles d'états très utiles dans la synthèse de commande par supervision :

- L'ensemble des états interdits frontières  $M_{IF}$  correspondant aux états faiblement interdits accessibles par occurrence d'évènements contrôlables.

$$M_{IF} = \{M_j \mid M_i \in M_A, M_j \in M_I \text{ ou } M_F \text{ et } \sigma \in E_c, M_i[\sigma]M_j\} \quad (23)$$

- L'ensemble des états autorisés critiques  $M_{AC}$  correspondant aux états à partir desquels l'occurrence d'évènements contrôlables mène à un état frontière.

$$M_{AC} = \{M_i \mid M_i \in M_A, M_j \in M_{IF} \text{ et } \sigma \in E_c, M_i[\sigma]M_j\} \quad (24)$$

L'espace d'états accessibles pertinent de commande par supervision est constitué de l'ensemble d'états autorisés  $M_A$  et de l'ensemble d'états interdits frontières  $M_{IF}$ .

L'application du **Pas 3** de l'algorithme de Kumar qui permet de calculer formellement l'ensemble des états interdits et d'identifier ensuite les états interdits

(frontières), associés aux contraintes admissibles nécessaires à la méthode des invariants de marquages.

Ainsi pour notre exemple, l'application du **Pas 3** de l'algorithme de Kumar au graphe de marquages (figure 4) donne les états interdits frontières suivants :  $\{M_{10} = P_2P_8, M_{13} = P_1P_5P_9\}$ .

Ils se traduisent par les contraintes suivantes :  $M(P_2) + M(P_8) \leq 1$  et  $M(P_1) + M(P_5) + M(P_9) \leq 2$

Le vecteur de contrainte associé est :  $L = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$

La matrice incidence du contrôleur est:

$$W_C = -LW_R \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 2 & 0 & 0 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

Le marquage initial du SED est :  $M_{R0} = [1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0]^T$  et  $b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Le marquage initial du contrôleur est :  $M_{C0} = b - LM_{R0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

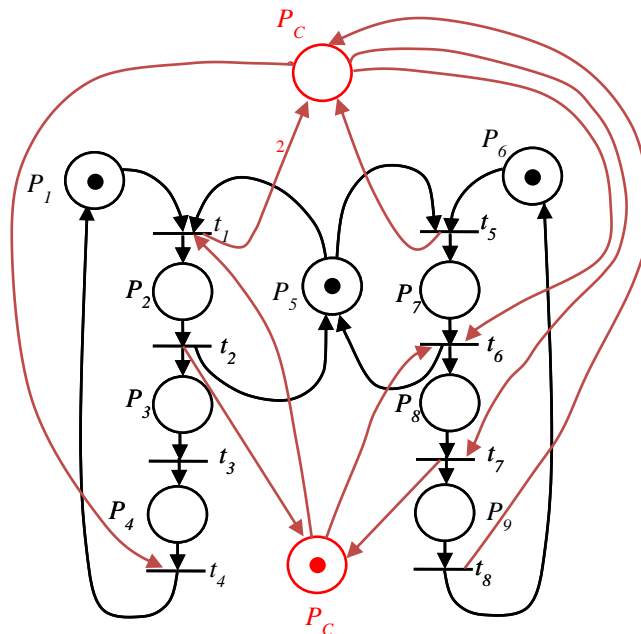


Figure 7. Modèle RdP de fonctionnement désiré du SED en boucle fermée

**Remarque :** Le problème, ici, est de trouver l'ensemble complet des contraintes admissibles qui prend en compte les états interdits par la synchronisation incontrôlable sans avoir à explorer les branches du graphe de marquages.

## 6. Détermination des contraintes admissibles par séparation des états du graphe de marquages

### 6.2. Matrice de transition associée au graphe de marquages

La fonction de transition du graphe de marquages de  $R = R_p || R_s$  est  $\delta : M \times E \rightarrow M$ . Elle fait le bilan de l'action de toutes les liaisons de  $R$  en assurant l'intégrité des informations vis-à-vis du graphe de marquages  $G$ , dont les états sont désignés  $M_i$  et les transitions associées aux événements sont désignées  $t_j$ . Nous associons à  $G$  une matrice  $\Delta = [\delta_{ij}]$  appelée matrice de transition définie par :

$$\delta_{ij} = M_i[t_j] \quad (25)$$

Nous choisissons de représenter cette écriture sous forme de table (Tableau 1),

Tableau 1. Matrice de transition d'un graphe de marquages

$\delta$	$t_1$	$\dots$	$t_m$
$M_0$	$\delta_{00}$	$\dots$	$\delta_{0m}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$M_n$	$\delta_{n1}$	$\dots$	$\delta_{nm}$

Les éléments  $\delta_{ij} = M_i[t_j]$  de la matrice de transitions peuvent être injectivement codés (Badouel *et al.*, 1994) de façon à permettre la séparation des états suivant leurs caractères interdit ou autorisé par la spécification. La codification consiste à attribuer une valeur algébrique définie par une fonction associative notée :

$$g : \delta_{ij} \rightarrow \{-1, 1\} \quad (26)$$

Nous obtenons ainsi une matrice de transition codifiée  $A = [a_{ij}] \in IR^{n \times m}$  :

$$a_{ij} = \begin{cases} 1 & ; \text{si } M_i \text{ est autorisé par la spécification} \\ -1 & ; \text{si } M_i \text{ est interdit par la spécification} \\ 0 & ; \text{sinon} \end{cases} \quad (27)$$



## 6.2. Séparation des états du Graphe de Marquages

La séparation consiste à associer à chaque état  $M_i$  du graphe de marquages son ensemble interdit  $M_I$  ou autorisé  $M_A$  vis-à-vis d'un ensemble d'états constituant l'hyperplan de séparation, noté  $H$ . Nous associons à  $H$  une fonction de décision  $g(M_i)$  définie sur l'ensemble des états  $M = M_A \cup M_I$  par :

$$g(M_i) = \beta + \langle \alpha_i, M_i \rangle ; \alpha_i \in \mathbb{R} \text{ et } \beta = 0 \text{ un vecteur du mot vide } \varepsilon \quad (28)$$

$H$  sépare strictement  $M_A$  et  $M_I$  si et seulement si :

$$\begin{cases} g(M_i) > 0 \text{ pour tout } M_i \in M_A \\ g(M_i) < 0 \text{ pour tout } M_i \in M_I \\ i = 0, \dots, n \end{cases} ; \quad (29)$$

Si  $H$  ne sépare pas strictement  $M$ , alors il existe un ensemble  $M_H$  tel que :

$$M_H = \{M_i : g(M_i) = 0\} \quad (30)$$

A cet effet, l'ensemble  $M_H$  des états constituant l'hyperplan peut appartenir soit à l'ensemble d'états interdits frontières; soit à l'ensemble d'états autorisés critiques.

Considérons  $T = t_1 \dots t_m$  une base canonique de  $g(M_i)$  telle que la matrice de transition codifiée dans cette base soit la matrice  $\mathbf{A}(t_1 \dots t_m)$ , notée  $\mathbf{A} = [\alpha_i]$ .

$$g(M_i) = [a_{ij}] \cdot [t_j]^T \cdot M_i = \sum a_{ij} \cdot M_i = \alpha_i \cdot M_i, \text{ avec } \alpha_i = \sum a_{ij} \quad (31)$$

Comme les marquages sont toujours positifs ( $M_i \geq 0$ ), la séparation va dépendre de  $\alpha_i = \sum_{j=1}^m a_{ij}$  :

- $\alpha_i < 0$  :  $M_i$  est un état interdit par la synchronisation des transitions incontrôlables
- $\alpha_i > 0$  :  $M_i$  est un état autorisé par la spécification
- $\alpha_i = 0$  :  $M_i$  est un état appartenant à l'ensemble  $M_H$  constituant l'hyperplan de séparation.

## 6.3. Application

Nous appliquons cette approche à notre exemple en partant du graphe de marquage obtenu dans la figure 6



$M_3$			1		1				2
$M_4$			1			1			2
$M_5$				1	1				2
$M_6$				1		1			2
$M_7$				1			1		2
$M_8$				1				1	2
$M_9$	1						1		2
$M_{10}$							-1	1	0
$M_{11}$		-1						-1	-2
$M_{12}$			1						1
$M_{13}$	-	1						1	0
$M_{14}$			1						1

L'application de notre méthode de séparation des états au graphe de marquages de la figure 6 à partir de la matrice de transition codifiée (Tableau 3) donne le résultat suivant :

- Etats autorisés :  $\{M_0, M_1, M_2, M_4, M_5, M_7, M_8, M_9, M_{12}, M_{14}\}$
- Etats interdits frontières :  $\{M_{10}, M_{13}\} \in M_H$
- Etats interdits :  $M_{11}$  ( $M_2$  est un état bloquant)

Les états interdits frontières sont identiques à ceux déterminés par l'algorithme de Kumar Etats blocage et donc le contrôleur obtenu par la méthode des invariants est idem à la figure 7

## 7. Conclusion

La synthèse de la commande par supervision des SED simple et efficace qu'offre la méthode des invariants est une réponse idoine au réel besoin des industriels. La fusion de cette méthode avec l'algorithme de Kumar apporte permet de l'optimiser et résoudre le problème de mise en œuvre de la commande par supervision basée sur les automates à états finis. Mais, l'idéale est de s'absoudre dans la mesure du possible de l'algorithme de Kumar afin de rester dans le contexte intégrale du RdP. C'est pourquoi, nous avons proposé une méthode simple de séparation du graphe de marquages, outil d'analyse du RdP du SED. Mais, cette approche reste partiellement structurelle parce qu'elle requière la construction du graphe de marquage. Notre

perspective est de contourner cet obstacle en proposant une synthèse complètement structurelle. En effet, des états interdits peuvent être générés lorsque la synchronisation entre le modèle du procédé et celui de la spécification est réalisée via des transitions incontrôlables et les inclusions de support entre les états interdits et les états autorisés peuvent entraîner l'interdiction de comportement désiré du SED.

### Références Bibliographiques

- Badouel E., Darondeau and Bernardinello, L. (1995). Polynomial algorithms for the synthesis of bounded nets. Proceedings Caap 95, Lecture notes in computer science. Springer. Vol 915, pp. 364–378.
- Boel R.K., Ben-Naoum L. and Van Breusegem, V. (1995). On forbidden state problems for a class of controlled Petri nets. *IEEE Transactions on Automatic Control*, Vol. 40, no 10, pp. 1717-1731.
- Cassandras C., Lafortune S. (2008). Introduction to Discrete Event Systems, 2<sup>nd</sup> Edition, Springer - Verlag.
- David R., Alla H. (2010). Discrete, Continuous, and Hybrid Petri Nets, Springer - Verlag.
- Dideban A., Alla H. (2008). Reduction of constraints for controller synthesis based on safe Petri nets. *Automatica*, Vol 44, no 7, pp. 1697–1706.
- Dorsaf E-B., Haddad S., and Hennicker R. (2012). Refinement and asynchronous composition of modal Petri Nets. Lecture Notes in Computer Science LNCS 6900, K. Jensen, S. Donatelli, and J. Kleijn (Eds.), Springer, pp. 96–120.
- Ehrenfeucht A. and Rozenberg G. (1990). Partial 2-structures, Part II: State Spaces of Concurrent Systems. *Acta Informatica*, Vol. 26, pp. 343–368
- Falk J. E., Bandurova Y. and Yeganova L. (2011). Sets separation problems and global optimization. Third world congress of nonlinear analysts, Elsevier science Ltd, Vol 47, pp.1857-1867.
- Ghaffari A., Rezg N., Xie, X. (2003). Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE Trans. Robot. Autom.*, Vol 19, no 1, pp. 137–141.
- Giua, A., Seatzu, C. (2007). A systems theory view of Petri nets, in Bonivento C., Marconi L., Rossi C., Isidori A. eds, Advances in Control Theory and Applications, *Lecture Notes in Control and Information Sciences*, Springer, Vol. 353, pp. 99–127.
- Godon, A. (1996). *Contribution à la commande des systèmes à événements discrets par réseaux de Petri*. Thèse de doctorat en Automatique Industrielle. Université d'Angers
- Holloway L. and Krogh H. (1990). Synthesis of feedback control logic for a class of controlled Petri nets. *IEEE Trans. Automatic control*, Vol. 35, no 5, pp. 514-523
- Hopcroft J., Motwani R., Ullman J. (2007). Introduction to Automata Theory, Languages, and Computation. 3rd Edition, Prentice Hall.

- Iordache M., Antsaklis P. (2006). *Supervisory Control of Concurrent Systems-A Petri Net Structural Approach*, Birkhäuser, Boston.
- Kattan B. (2004). *Synthèse structurelle d'un contrôleur basée sur le Grafcet*. Thèse de doctorat PhD, Université Joseph Fourier, Grenoble
- Komenda J., Lahaye S. et. Boimond J-L. (2009). Le produit synchrone des automates (max, +). *JESA-MSR*, Vol. 43, pp. 1033-1047
- Kumar R. (1991). *Supervisory Synthesis Techniques for Discrete Event Dynamical Systems*. Thesis for the degree of Doctor of Philosophy, University of Texas
- Marchand H. and Gaudin B. (2002). Supervisory Control Problems of Hierarchical Finite State Machines. *41th IEEE Conference on Decision and Control*, Las Vegas, USA
- Moody J., Antsaklis P. (1998). *Supervisory Control of Discrete Event Systems Using Petri Nets*, *Kluwer Academic Publishers*, Boston.
- Nourelfath M. et Niel E. (2000). Relaxation du concept de contrôlabilité pour une synthèse de la commande des systèmes de production. Première Conférence Internationale Francophone d'Automatique (CIFA), Lille, pp. 526-531
- Peterson J.L. (1981). *Petri Net Theory and the Modelling of Systems*. *Prentice-Hall*
- Pocci et al., 2011, : M. Pocci, I. Demongodin, N. Giambiasi et A. Giua (2011). Synchronizing sequences on not strongly connected Petri nets. In *Symposium On Theory of Modelling and Simulation (DEVS/TMS'11)*, Boston, USA
- deQueiroz M.H and Cury J.E.R. (2002). Synthesis and implementation of local modular supervisory control for a manufacturing cell. *Proceeding of sixth International Workshop on Discrete Event Systems*, pp 377-382, 2002
- Ramadge P.J., Wonham W.M. (1989). The control of discrete event systems. *Proceedings of the IEEE*, Vol. 77, no 1, pp. 81-98
- Ramadge P.J., Wonham W.M. (1983). Supervisory control of a class of discrete event processes. *Lecture Notes in Computer Science (LNCIS)*, *Springer-Verlag*, Germany, Vol. 63, pp. 477-498
- Seatzu C., Silva M. J. and van Schuppen H. (2012). Control of discrete-event systems, Automata and Petri net Perspectives. *Lecture Notes in Control and Information Science*, *Springer*, Vol. 433
- Sogbohossou M. et Vianou A. (2016). Un dépliage par processus pour calculer le préfixe complet des réseaux de Petri. *Proceedings of CARI*, pp. 97
- Uzam M., Wonham W. (2006). A hybrid approach to supervisory control of discrete event systems coupling RW supervisors to Petri nets. *Int. J. of Adv. Manuf. Technol.* Vol 28, no 7 - 8, pp. 747 - 760.
- Vasiliu A., Alla H. (2011). Border Forbidden States and Constraints for Optimal Controller Synthesis using Generalized Petri Nets, *Automatica*.
- Vasiliu A., Dideban A., Alla, H. (2009). Control synthesis for manufacturing systems using non-safe Petri nets. *J. of Control Eng. and Applied Informatics* Vol 11, no 2, pp. 43 -50.

- Wonham W. M. (2011). Supervisory control of discrete event systems. *Technical report*, University of Toronto, Dept. of Electrical and Computer Engineering. <http://www.control.utoronto.ca/~wonham/>.
- Wonham W. M. (2003). Notes on control of discrete-event systems. *Technical report*, no ECE 1636F/1637S, University of Toronto, Dept. of Electrical and Computer Engineering. <http://www.control.utoronto.ca/~wonham/>
- Yamalidou K., Moody J. O., Lemmon M., Antsaklis P. (1996). Feedback Control of Petri Nets Based on Place Invariants. *Automatica*, Vol. 32, no. 1, pp. 15-28
- Yoo T.-S. and S. Lafortune (2002). A general architecture for decentralized supervisory control of discrete-event systems. *Discrete Event Dynamic Systems: Theory & Applications*, Vol. 12, no. 3, pp. 335–377
- Zhou M.C. and DiCesare F. (1993). Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. *Kluwer Academic Publishers*, Boston