République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

MEMOIRE DE MAGISTER

En Automatique

Option : Automatique des Systèmes Continus et Productiques

Présentée par :

Hayet HARROUCHE

Thème :

Synthèse de superviseur basée sur les réseaux de Petri

Jury :

MELAH Rabah, Maître de Conférences Classe A, UMMTO, Président

DJENNOUNE Saïd, Professeur, UMMTO, Rapporteur

MAIDI Ahmed, Maître de Conférences Classe A, UMMTO, Examinateur

KARA Redouane, Maître de Conférences Classe A, UMMTO, Examinateur

PDF created with pdfFactory Pro trial version www.pdffactory.com

Table des matières

PDF created with pdfFactory Pro trial version www.pdffactory.com

Table des Matières

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 2 : Méthodes de synthèse de commande des SED

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 3 : Synthèse de superviseur en présence de transitions non contrôlables.

Chapitre 4 : Supervision par invariance.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Introduction Générale

PDF created with pdfFactory Pro trial version www.pdffactory.com

Introduction générale :

Par opposition aux systèmes dynamiques dont l'évolution est continue dans le temps et peut être décrite par des équations différentielles, les systèmes à événements discrets (SED) sont des systèmes dynamiques dont les transitions entre les états sont associées à des événements.

Des théories et des modèles spécifiques à cette classe de systèmes sont nécessaires pour pouvoir faire de la modélisation, de l'analyse de performances et de la commande. La modélisation est la première étape pour la commande des SED, et elle peut être réalisée avec différents outils tels que : les automates à états finis et les langages formels, l'algèbre max-plus, les chaînes de Markov, le grafcet et les réseaux de Petri.

Le problème général de la commande d'un système à événements discrets consiste, à établir une loi de commande adéquate à partir de la connaissance d'un modèle du système et du fonctionnement souhaité (objectif à atteindre). Dans les SED, la commande a pour effet de restreindre l'ensemble des états possibles du système à un sous-ensemble d'états admissibles.

Dans la théorie de la supervision de Ramadge et Wonham, l'outil utilisé pour modéliser les SED est les automates et les langages. Bien que cet outil fournit une structure générale pour établir les propriétés fondamentales du problème de contrôle des SED. L'inconvénient majeur de son utilisation est celui de l'explosion combinatoire : en effet, pour des systèmes complexes le nombre d'états utilisés pour modéliser le système croît exponentiellement, ce qui limite la possibilité de développer des algorithmes efficaces pour l'analyse et la synthèse de la commande.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Ces limitations ont poussé à l'utilisation de modèles plus évolués, tels que les réseaux de Petri. Cet outil est très puissant en raison de sa capacité graphique et algébrique essentiellement appliqué aux SED. Les RDP présentent plusieurs avantages par rapport aux automates :

-Les états des RDP sont représentés par le marquage de ses places. Ainsi le RDP donne une description plus compacte car la structure du réseau peut être maintenue petite même si le nombre de marques augmente.

-Au lieu d'utiliser les descriptions textuelles ambiguës ou les notations mathématiques qui sont difficiles à comprendre, le système et les spécifications peuvent être représentés par un graphe facile à comprendre en utilisant les RDP.

-Finalement, en utilisant les RDP, le modèle peut être utilisé pour l'analyse des propriétés et l'évaluation des performances et la construction systématique du superviseur.

Le concept de la supervision des SED peut s'énoncer comme suit :

-Il existe deux types d'événements qui peuvent se produire dans les SED : les événements contrôlables, qui peuvent être contrôlés par l'action de la commande, et les événements non contrôlables qui ne peuvent pas être inhibé par l'action d'un contrôle.

-Etant donné un modèle SED et un comportement désiré du système à contrôler, l'objectif est de synthétiser un superviseur pour satisfaire ce comportement désiré.

-Le superviseur doit être non bloquant et maximum permissif.

Comme la synthèse de superviseur est une question essentielle dans la commande des SED, des méthodes efficaces ont été proposées pour la conception de superviseur. C'est la théorie de RW qui est à l'origine de toutes ces méthodes.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Le problème principal de la théorie de supervision est l'existence des événements non contrôlables. Lors de synchronisation d'un événement non contrôlable du modèle du procédé avec celui de la spécification, le modèle final ne peut pas respecter cette synchronisation. Dans ce cas il apparaîtra un ensemble d'états interdits. Pour les approches basées sur les automates, on peut résoudre ce problème par l'algorithme proposé par Kumar. L'inconvénient principal est que le contrôleur est donné sous forme d'automate dont la taille peut être inexploitable.

La contribution principale de ce travail de Magister, est la proposition d'une approche qui nous donne l'ensemble des transitions à interdire pour garantir le respect des spécifications.

Notre objectif dans ce travail est d'étudier les SED en vue de leur commande. Pour cela il nous a paru nécessaire de commencer par une présentation générale des SED. C'est l'objet du premier chapitre. Tout d'abord nous évoquons quelques notions relatives aux langages formels. Puis nous présentons les outils les plus usuels dans le champ de commande des SED tels que les automates, et les Réseaux de Petri.

Le deuxième chapitre présente dans un premier temps les principales méthodes existantes dans la littérature sur la commande des systèmes à événements discrets. La théorie de la supervision des SED proposée par Ramadge & Wonham est d'abord présentée en détail. Elle est à l'origine de la commande par supervision. Puis les approches de synthèse de lois de commande basées sur le Réseaux de Petri sont décrites.

Nous présentons au chapitre trois le cas où les transitions sont non contrôlables en explicitant l'algorithme de paramètrisation linéaire des moniteurs proposé par Basile

PDF created with pdfFactory Pro trial version www.pdffactory.com

et al, 2009. Une application de cet algorithme à un système de production modélisé par réseau de Petri est effectuée.

Le chapitre quatre est consacré à la présentation d'une approche de commande proposée dans le cadre de ce mémoire. Cette contribution est basée sur la représentation des spécifications sous forme d'un ensemble polyèdral, ensemble des contraintes. Un certain nombre de résultats sont obtenu en appliquant la propriété d'invariance positive bien connue dans le cas des systèmes continus. Il s'agit en fait, de garantir que le franchissement d'une transition ne fait pas sortir le système de cet ensemble polyèdral.

Enfin, nous terminons notre mémoire par une conclusion et des perspectives.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 1

Généralités sur les systèmes à événements discrets

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 1

Généralités sur les systèmes à événements discrets

1.1. Introduction

Avec les progrès de la technologie, l'homme s'est mis à construire des systèmes de plus en plus complexes dont l'enchaînement dynamique des tâches provient de phénomènes de synchronisation, d'exclusion mutuelle ou de compétition dans l'utilisation des ressources communes. Ce qui nécessite une politique d'arbitrage ou de priorité. Ce type de dynamique échappe totalement à la modélisation par équations différentielles.

Au début des années 80, le monde de l'automatique a pris en compte ces systèmes, on les a alors appelé systèmes à événements discrets. Le mot **'discret'** ne signifie ni temps discret, ni état discret, mais se réfère au fait que la dynamique est composée d'événements qui peuvent avoir une évolution continue, ce qui nous intéresse, c'est le début et la fin de ces événements, dans la mesure où les fins conditionnent de nouveaux débuts.

La théorie des SED peut être divisée en deux grandes approches :

L'approche qualitative (logique) qui ne s'intéresse qu'à l'occurrence des événements ou l'impossibilité de cette occurrence (impasse ou blocage), et à la succession de ces événements.

PDF created with pdfFactory Pro trial version www.pdffactory.com

L'approche quantitative qui s'intéresse à l'aspect évaluation des performances (nombre d'événements survenant dans un laps de temps donné), et à l'optimisation de ces performances.

1.2. Différents outils de modélisation des SED

Les réseaux de Petri, l'algèbre des dioïdes, les langages et les automates, sont les trois outils de base utilisés pour la modélisation, la commande et l'analyse des SED.

1. Langages et automates

Ils permettent le traitement mathématique des problèmes relatifs aux SED, essentiellement du point de vue logique (analyse qualitative).
Une présentation plus détaillée est donnée dans [5], [11], [19].

1.1. Langages :

Définitions de base

Un alphabet est un ensemble fini de symboles par exemple {a, b, c…,z}. On suppose l'alphabet fini et on le note **Σ**. Un mot est une suite finie de lettre dans l'alphabet de référence, exemple : $w= \sigma_1\sigma_2\sigma_3.....\sigma_k$, $\sigma_{i\in}$ **Σ**, i=1,…,k. On appelle k la longueur du mot w, notée $|w|$.

La concaténation de deux mots w_1, w_2 sur **Σ** est le mot $w_1.w_2$. La longueur de ce mot est $|w_1|+|w_2|$.

On note **Σ *** l'ensemble de tous les **mots** sur **Σ**. Le mot de longueur zéro, noté **ε**, est le mot vide, **ε** ∈ **Σ** *.
Soit un mot w ∈ **Σ** *, un mot u ∈ Σ * est un **préfixe** de w s'il existe un mot v ∈ **Σ** * telle que : u **.**v = w.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Un **langage** sur l'alphabet **Σ** est un sous ensemble de **Σ ***, et on le note **L**. L = Ø est le langage vide, il ne contient aucun mot.

On définit la **préfixe-clôture** d'un langage L, comme étant le langage noté $\overline{L}$, contenant tous les préfixes des mots de L.

On dit qu'un langage L est **préfixe-clos**, s'il est égal à sa préfixe-clôture L = $\overline{L}$.

1.2. Opérations sur les langages

Soit deux langages : L_1 défini sur Σ_1, et L_2 défini sur Σ_2.

Union : l'union de L_1 et L_2, notée $L_1 \cup L_2$, est le langage contenant tous les mots contenues soit dans L_1, soit dans L_2.

Formellement : $L_1 \cup L_2 = \{v / v \in L_1 \text{ ou } v \in L_2\}$.

Intersection : l'intersection de L_1 et L_2, notée $L_1 \cap L_2$, est le langage contenant tous les mots contenues à la fois dans L_1 et dans L_2.

Formellement : $L_1 \cap L_2 = \{v / v \in L_1 \text{ et } v \in L_2\}$.

Différence : la différence entre L_1 et L_2, notée $L_1 - L_2$ ou L_1 / L_2, est le langage contenant tous les mots de L_1 qui ne sont pas dans L_2.

Formellement : $L_1 - L_2 = \{ v / v \in L_1 \text{ et } v \notin L_2\}$.

Concaténation : la concaténation de L_1 et L_2, notée $L_1 . L_2$ est le langage contenant tous les mots formés d'un mot de L_1 suivie d'un mot de L_2.

Formellement : $L_1 . L_2 = \{w / w = u.v, \ u \in L_1 \text{ et } v \in L_2\}$.

Fermeture itérative : la fermeture itérative d'un langage L, notée L^*, est l'ensemble des mots formés par une concaténation finie des mots de L.

Formellement : $L^* = \{v / \exists k \geq 0 \text{ et } v_1, v_2, \ldots . v_k \in L \text{ et } v = v_1 . v_2 \ldots . v_k\}$.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Projection naturelle : soient deux alphabets Σ_1 et Σ_2 avec $\Sigma_1 \cap \Sigma_2 \neq \emptyset$. Soit $\Sigma = \Sigma_1 \cup \Sigma_2$. On définit la projection naturelle $Pi : \Sigma^* \rightarrow \Sigma^*_i$ (i=1,2), telle que :

$Pi(\varepsilon) = \varepsilon$.

$$Pi(\sigma) = \begin{cases} \varepsilon & si\ \sigma \notin \sum_i \\ \sigma & si\ \sigma \in \sum_i \end{cases}$$

$Pi(s\sigma) = P_1(s).P_2(\sigma)$ avec $s \in \Sigma^*$, $\sigma \in \Sigma$.

Projection naturelle inverse : soient deux alphabets Σ_1 et Σ_2, avec $\Sigma_1 \cap \Sigma_2 \neq \emptyset$. Soit $\Sigma = \Sigma_1 \cup \Sigma_2$.

On définit la projection naturelle inverse $P^{-1}{}_i : \Sigma^*{}_i \rightarrow \Sigma^*$ telle que :

$P^{-1}{}_1(\varepsilon) = (\Sigma_2 - \Sigma_1)^*$

$P^{-1}{}_1(\sigma) = (\Sigma_2 - \Sigma_1)^* \sigma (\Sigma_2 - \Sigma_1)^*$

$P^{-1}{}_1(s\sigma) = P^{-1}(s)\ P^{-1}(\sigma)$ avec $s \in \Sigma^*{}_1$, $\sigma \in \Sigma^*{}_1$

1.3. Langage régulier

Une expression régulière sur un alphabet Σ est composée d'opérandes (symboles de Σ), et d'opérateurs pris dans l'ensemble {+, **. , ***}.

Un langage régulier est celui défini par une expression régulière. Les opérateurs sont classés du plus prioritaire au moins prioritaire, dans l'ordre : fermeture itérative, concaténation et union.

1.4. Automates

Un automate est une machine à états qui permet de décrire le fonctionnement d'un système à événements discrets.

1.4.1. Définitions

-Un automate peut être défini par un 5-uplet $A = (Q, \Sigma, \delta, q_0, Q_m)$ tel que :

Q : est l'ensemble des états (espace d'états discret).

Σ : est un alphabet fini.

PDF created with pdfFactory Pro trial version www.pdffactory.com

δ : est la fonction de transition.

q_0 : est l'ensemble des états initiaux.

Q_m : est l'ensemble d'états finaux (marqués) $Q_m \subseteq Q$.

-Si l'ensemble des états Q est fini, on parle d'automate fini.

-Un automate est dit déterministe, si l'état initial est unique, et si la relation de transition, appliquée à un couple (q_i , σ) définit toujours un unique état δ: Q x Σ → Q.

-Un automate non déterministe est défini de la même façon qu'un automate déterministe, excepté qu'il peut avoir plusieurs états initiaux, et sa relation de transition est définie par δ : Q x Σ →2^Q. Ce qui veut dire qu'à partir d'un état donné, un même symbole σ permet d'atteindre deux états différents, au moins.

1.4.2. Représentation

Un automate peut être représenté par un graphe de transition d'états. Les états sont représentés par des cercles.

L'état initial est repéré par une flèche entrante.

Les états finaux sont représentés par des doubles cercles.

La fonction de transition δ est représentée par des arcs associés à des symboles de l'alphabet Σ.

Exemple 1.1:

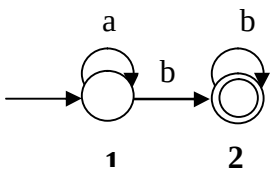


Figure 1.1 : Automate A.

Soit A = (Q, Σ, δ, q_0, Q_m) l'automate de la figure 1.1.

Q ={1 , 2}, Σ ={a , b}, q_0 ={1}, Q_m ={2}, δ ={(1,a,1), (1,b,2), (2,b,2) }.

PDF created with pdfFactory Pro trial version www.pdffactory.com

1.4.3. Langage reconnu par un automate

Un mot w est reconnu ou accepté par un automate fini, s'il existe un chemin menant de l'état initial q_0 à un état final de Q_m, étiqueté par la suite de lettres du mot w dans le même ordre.

L'ensemble des mots acceptés par un automate fini A forme le langage reconnu par cet automate, on le note L(A).

1.4.4. Langage généré et langage marqué

Le langage généré par un automate A est donné par : $L(A) = \{s \in \Sigma^* / \delta(q_0,s) \in Q\}$.

Ce langage représente l'ensemble de tous les mots qui permettent d'atteindre un état quelconque de l'automate à partir de l'état initial. Toute séquence d'événements qui se produit dans l'automate est précédée par des préfixes, donc $L(A) = \overline{L}(A)$, d'où tout langage généré par un automate fini est préfixe-clos.

On définit le langage marqué noté $L_m(A)$ par l'ensemble de tous les mots w de L(A) telle que $\delta(q_0,w) \in Q_m$. Ce langage représente l'ensemble de tous les mots qui permettent d'atteindre un état marqué à partir de l'état initial.

$L_m(A) \subseteq L(A)$, puisque $Q_m \subseteq Q$.

1.4.5. Accessibilité et co-accessibilité

Un état q est dit accessible s'il existe un mot w telle que $q = \delta(q_0,w)$, c, à, d que l'automate peut y accéder depuis l'état initial. L'automate est dit accessible si $\forall q \in Q$, q est accessible.

Un état q est dit co-accessible, s'il existe un mot w telle que $\delta(q,w) \in Q_m$, c, à, d qu'à partir de cet état l'automate peut atteindre un état marqué. L'automate est dit co-accessible si $\forall q \in Q$, q est co-accessible.

1.4.6. Automate non bloquant et émondé

PDF created with pdfFactory Pro trial version www.pdffactory.com

Un automate A est qualifié de non bloquant lorsque tout état accessible est co-accessible, c'est-à-dire, que tout mot générée par A est le préfixe d'un mot marqué : $L(A) = \overline{L}_m(A)$.

Un automate A est émondé s'il est à la fois accessible et co-accessible. Si A est émondé, A est aussi non bloquant, la réciproque est fausse : un automate non bloquant peut avoir des états non accessibles.

1.4.7. **Théorème de Kleene**

Le théorème de Kleene établi les équivalences suivantes :

1- Un langage sur un alphabet Σ est régulier (rationnel) si et seulement si il est reconnu par un automate fini. En d'autres termes, pour toute expression régulière, on peut construire un automate fini qui reconnaisse cette expression. De même, pour tout automate fini, on peut exprimer sous forme d'une expression régulière le langage qu'il reconnaît.

A partir d'une expression régulière, on peut construire un automate fini : en effet les expressions régulières de base peuvent être décrites par des automates de la Figure 1.2 :

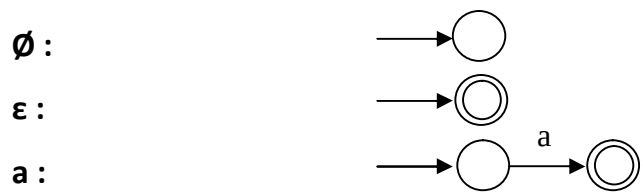


Figure 1.2 **:** Expressions régulière de base décrites par des automates

De même, pour les opérations sur les expressions régulières.

2- A partir d'un automate fini on peut construire une expression régulière qui décrit le langage reconnu par cet automate : les systèmes d'équations linéaires unilatères permettent le calcul d'une expression régulière à partir d'un automate fini.

PDF created with pdfFactory Pro trial version www.pdffactory.com

-A chaque état i on associe une expression régulière X_i du langage associé à cet état.

-On obtient un système d'équations dont les inconnus sont des expressions régulières, X_i.

-Si i =1, c'est l'état initial, alors X_1 décrit le langage reconnu par l'automate.

Exemple 1.2:

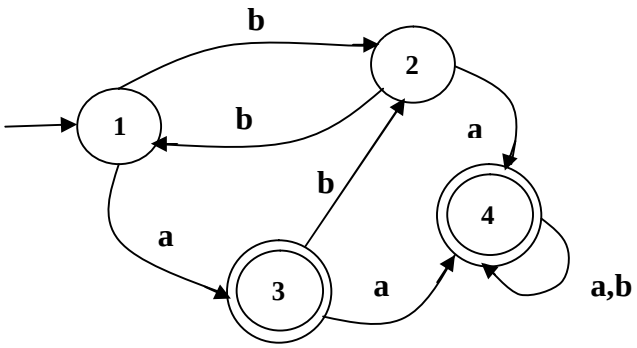


Figure 1.3 : Automate B

Soit l'automate de la figure 1.3 :

$X_1 = bX_2 + aX_3$.

$X_2 = bX_1 + aX_4$.

$X_3 = \varepsilon + aX_4 + bX_2$.

$X_4 = \varepsilon + (a+b)X_4$.

Résoudre un tel système revient à calculer X_1, car il est associé à l'état initial, et pour cela on procède par substitutions. Deux résultats sont nécessaires pour la résolution de ces équations :

-La plus petite solution de l'équation $X = AX + B$ est $X = A^* B$.

-La plus petite solution de l'équation $X = XA + B$ est $X = BA^*$.

1.2.8 Composition d'automates

PDF created with pdfFactory Pro trial version www.pdffactory.com

Un procédé réel est souvent composé de procédés élémentaires en interaction, de là vient l'intérêt de la composition des automates qui les modélisent.

La composition intervient aussi dans l'élaboration de la commande par supervision du procédé (composition du procédé avec les spécifications).

1. Si $\Sigma_1 \cap \Sigma_2 \neq \emptyset$ synchronisation des événements étiquetés de σ_i en commun.
2. Si $\Sigma_1 \cap \Sigma_2 = \emptyset$ pas de synchronisation d'événements.
3. Si $\Sigma_1 = \Sigma_2$ synchronisation totale.

Composition synchrone :

Soient $A_1 = (Q, \Sigma_1, \delta, q_0, Q_m)$, $A_2 = (X, \Sigma_2, \xi, x_0, X_m)$, $\Sigma_1 \cap \Sigma_2 \neq \emptyset$, $\Sigma = \Sigma_1 \cup \Sigma_2$.

Le composé synchrone est défini $(Q \times X, \Sigma_1 \text{ E } \Sigma_2, \delta \times \xi, q_0 \times x_0, Q_m \times X_m)$.

$Q \times X$, définit l'ensemble des états.

$\Sigma_1 \cup \Sigma_2$, définit l'alphabet du composé.

$(q_0 \times x_0)$: est l'état initial du composé.

$(Q_m \times X_m)$: est l'ensemble des états finals.

La fonction de transition d'états $\delta \times \xi$ est définie par :

$(\delta \times \xi)\ ((q,x),\sigma) = (q',x')$ si $\delta(q,\sigma) = q'!$ et $\xi(x,\sigma) = x'!$

$(\delta \times \xi)\ ((q,x),\sigma) = (q',x)$ si $\delta(q,\sigma)!$ Avec $\sigma \in \Sigma_1 / \Sigma_2$.

$(\delta \times \xi)\ ((q,x),\sigma) = (q ,x')$ si $\xi(x,\sigma)!$ Avec $\sigma \in \Sigma_2 / \Sigma_1$.

Exemple 1.3:

Soient les automates A_1 et A_2, dont $\Sigma_1 = \{\alpha, \beta\}$ et $\Sigma_2 = \{\beta, \gamma\}$ respectivement. $\Sigma_1 \cap \Sigma_2 = \{\beta\}$.(figure 1.4)

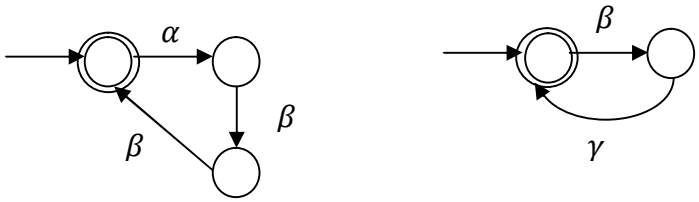


Figure 1.4 automates A_1, A_2.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Le résultat final est donné en figure 1.5

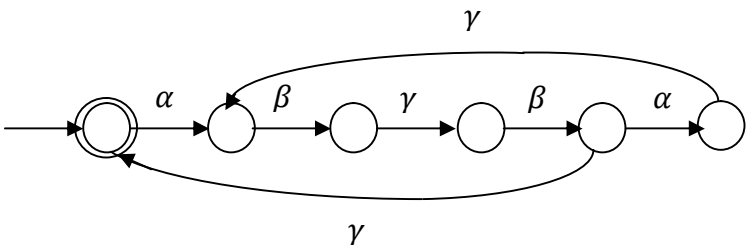


Figure 1.5 composé synchrone

Composition asynchrone :

Soient $A_1 = (Q, \Sigma_1, \delta, q_0, Q_m)$, $A_2 = (X, \Sigma_2, \xi, x_0, X_m)$, $\Sigma_1 \cap \Sigma_2 = \emptyset$, $\Sigma = \Sigma_1 \cup \Sigma_2$.

Le composé asynchrone est défini $(Q \times X, \Sigma_1 \, E \, \Sigma_2, \delta \times \xi, q_0 \times x_0, Q_m \times X_m)$.

La fonction de transition d'états $\delta \times \xi$ définie par :

$(\delta \times \xi)((q,x),\sigma) = (q',x)$ si $\delta(q,\sigma) = q'$ pour $\sigma \in \Sigma_1$.

$(\delta \times \xi)((q,x),\sigma) = (q,x')$ si $\xi(x,\sigma) = x'$ pour $\sigma \in \Sigma_2$.

La composition asynchrone n'est qu'un cas particulier de la composition synchrone.

Exemple 1.4:

Considérons un système manufacturier composé de deux machines identiques : M_1 et M_2, et un stock entre ces deux machines. Conformément à la figure 1.6, les deux machines travaillent de façon indépendante, puisent des pièces brutes en amont et rejettent des pièces usinées en aval.

Fig. 1.6 Un système manufacturier

Soient les automates M_1 et M_2 dont $\Sigma_1 = \{d_1, f_1, p_1, r_1\}$ et $\Sigma_2 = \{d_2, f_2, p_2, r_2\}$ respectivement. (figure 1.7)

PDF created with pdfFactory Pro trial version www.pdffactory.com

$$\Sigma_1 \cap \Sigma_2 = \phi$$

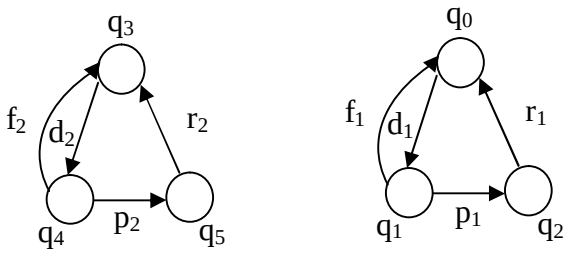


Figure1.7 automates M_1 ,M_2.

Le résultat final est donné en figure 1.8

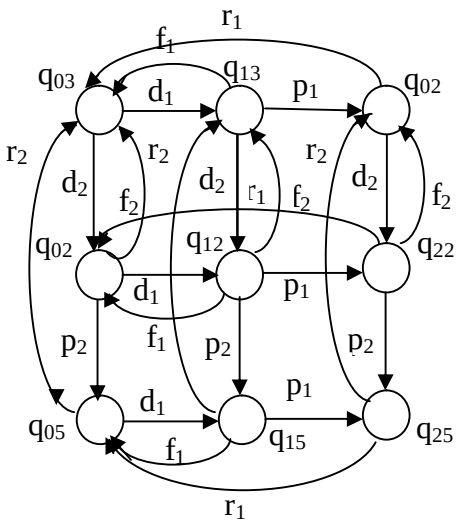


Figure 1.8 composé asynchrone.

2 Réseaux de Petri [1], [3]

2.1 Définition 1.1

Définition informelle : un réseau de Petri (RdP) est un graphe biparti orienté, composé de places et de transitions, reliées par des arcs. Une place peut contenir un nombre entier de jetons ou marques. A chaque arc peut être associé un poids (nombre entier) qui détermine la dynamique du réseau.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Définition formelle : un réseau de Petri est quadruplet **Q = (P, T, Pré, Post)** où :

$P = \{P_1, P_2, \ldots\ldots P_n\}$ est un ensemble fini de places.

$T = \{T_1, T_2, \ldots\ldots T_m\}$ est un ensemble fini de transitions.

Pré : P x T→ N est l'application d'incidence avant.

Post : P x T→ N est l'application d'incidence arrière.

$Pré(P_i, T_j)$ représente le poids de l'arc reliant P_i à T_j, alors que $Post(P_i, T_j)$ représente le poids de l'arc reliant T_j à P_i.

On définit la matrice d'incidence du réseau par l'application W telle que :

$\mathbf{W = W^+ - W^-}$.

Avec: $W^+_{ij} = Pré(P_i, T_j)$, $W^-_{ij} = Post(P_i, T_j)$.

On définit le vecteur de marquage $M = (m(P_1), m(P_2), \ldots.m(P_i), ..)$.

2.2 Franchissement d'une transition

Une transition T_j est franchissable si les places P_i en amont de T_j ont un marquage supérieur ou égal au poids de l'arc reliant P_i à T_j : $\forall P_i \in P, m(P_i) \geq Pré(P_i, T_j)$.

Lors du franchissement de T_j, le marquage des places P_i en amont de T_j est décrémenté de $Pré(P_i, T_j)$ marques, et le marquage des places P_k en aval de T_j est incrémenté de $Post(P_i, T_j)$ marques.

2.3 Equation fondamentale

On appelle séquence S une suite ordonnée de franchissements. Le vecteur $\underline{S}$ dont les composantes sont le nombre d'occurrence de franchissement de chaque transition (indépendamment de l'ordre de franchissement), est appelé vecteur caractéristique de la séquence S.

Le passage d'un marquage M_0 à un marquage M_f s'écrit sous la forme matricielle suivante :

$M_f = M_0 + W.\underline{S}$

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.4 Propriétés d'un PdP :

2.4.1 Accessibilité : un marquage M_f est accessible à partir du marquage initial M_0 s'il existe une séquence de franchissement S telle que : du franchissement de cette séquence à partir de M_0, résulte un nouveau marquage M_f et on note: M_0 [S › M_f

Cette propriété permet de savoir si un état non désiré risque de se produire.

2.4.2 Vivacité : Un réseau de Petri est dit **vivant** pour un marquage initial M_0 si pour tout marquage M accessible à partir de M_0 et pour toute transition T_i, il existe une séquence de franchissement S qui inclut la transition T_i.

Cette propriété permet de déduire si le système ne comporte pas de blocage.

2.4.3 Bornitude : Une place P_i est dite **bornée** pour un marquage initial M_0 si pour tout marquage accessible à partir de M_0, le nombre de marques dans P_i est fini.

Un RdP est dit borné si toutes ses places sont bornées.

2.4.4 Réversibilité : Un RdP est dit réversible ou **réinitialisable** pour un marquage initial M_0, si pour tout marquage M accessible à partir de M_0, il existe une séquence de franchissement S qui ramène à M_0.

2.5 Eléments de modélisation

Les RdPs permettent de modéliser un certain nombre de comportements importants dans les systèmes à événements discrets: le parallélisme, le partage des ressources, la mémorisation et lecture d'informations, la limitation d'une capacité de stockage.

2.5.1 Parallélisme :(Figure 1.9) Le parallélisme représente la possibilité que plusieurs processus évoluent simultanément au sein du même système.

PDF created with pdfFactory Pro trial version www.pdffactory.com

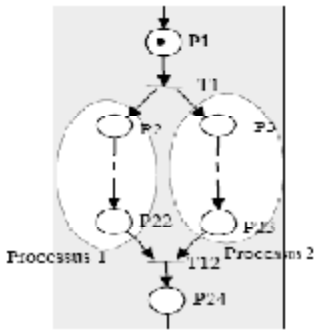


Figure 1.9 Parallélisme.

2.5.2 Synchronisation Mutuelle : (Figure 1.10) La synchronisation mutuelle ou rendez-vous permet de synchroniser les opérations de deux Processus.

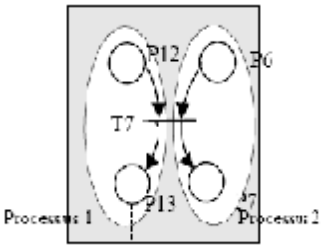


Figure 1.10 Synchronisation mutuelle

2.5.3 Sémaphore : (Figure 1.11) Les opérations du processus 2 ne peuvent se produire que si le processus1 a atteint un certain niveau dans la suite de ses opérations.

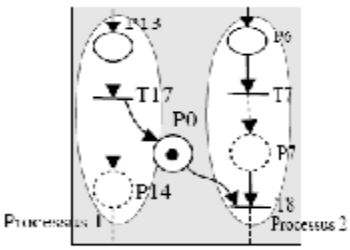


Figure 1.11 Sémaphore.

2.5.4 Partage des ressources : (Figure 1.12) Cette structure modélise le fait qu'au sein du même système plusieurs processus partagent une même ressource

PDF created with pdfFactory Pro trial version www.pdffactory.com

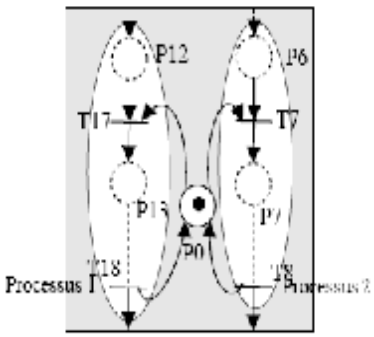


Figure 1.12 Partage des ressources

2.5.5 Mémorisation : (Figure 1.13) Mémorisation d'une transition (c, à, d, l'occurrence d'un événement), le franchissement de la transition T_{12} n'est possible que s'il y a une marque dans la place P_2. Seul le franchissement de T_1 peut mettre une marque dans la place P_2.

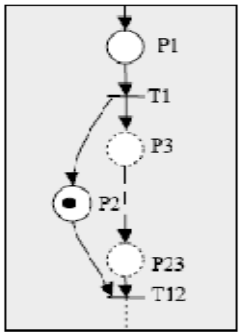


Figure 1.13 Mémorisation

2.5.6 Lecture : (Figure 1.14) Le franchissement de la transition T_1 est lié au marquage de P_{23}. lors du franchissement, son marquage n'est pas modifié. On fait alors une lecture de ce marquage.

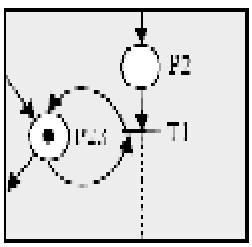


Figure 1.14 Lecture

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.5.7 Capacité limitée : (Figure 1.15) Pour que la transition T_3 soit franchissable il est nécessaire que la place P_5 contienne des marques. Le marquage de P_5 ne permet que deux franchissements successifs de T_3, la transition T_3 sera à nouveau franchissable si le franchissement de la transition T_4 permet de mettre des marques dans la place P_5. Au total, la place P_4 ne pourra pas contenir plus de 3 marques, cette partie du réseau peut modéliser un stock de capacité 3.

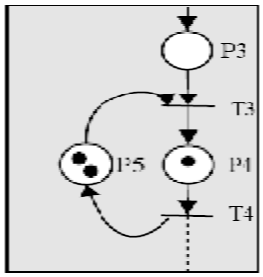


Figure 1.15 Capacité limitée

2.6 Les classes de Réseaux de Petri

2.6.1 RdP sans conflit : C'est un RdP dans lequel toute place a au plus une transition de sortie.

Conflit : ‹ P_1, T_1, T_2 › (figure 1.16).

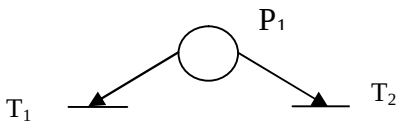


Figure 1.16 Conflit : ‹ P_1, T_1, T_2 ›.

2.2.1 RdP à choix libre: C'est un RdP dans lequel pour tout conflit ‹ P_i, T_1, T_2… ›, aucune des transitions T_1,T_2… ne possède une autre place que P_i.

2.6.2 RdP simple : C'est un RdP dans lequel chaque transition ne peut être concernée que par un conflit au plus.

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.6.3 RdP pur : C'est un RdP dans lequel il n'existe pas de transition ayant une place d'entrée qui soit également place de sortie de cette même transition.

2.6.4 Graphe d'événements : C'est un RdP dans lequel chaque place a exactement une transition d'entrée et une transition de sortie, ainsi un graphe d'événements ne présente jamais de conflits, mais peut toute fois comporter des synchronisations.

2.6.5 Graphe d'états : C'est un RdP dans lequel chaque transition a exactement une place d'entrée et une place de sortie, c'est le dual du graphe d'événements, ainsi un graphe d'états peut présenter des conflits, mais jamais comporter de synchronisations.

Définition : [20]

Un réseau de Petri est appelé :

Une arrière-machine à états si et seulement si chaque transition posséde une seule place de sortie.

Une avant-machine à états si et seulement si chaque transition possède une seule place d'entrée.

2.7 Réseaux de Petri particuliers :

2.7.1 RdPs à arcs inhibiteurs :

Un ars inhibiteur est un arc orienté d'une place P_i vers une transition T_j, pour le distinguer d'un arc classique, on représente son extrémité par un cercle à la place d'une flèche. Lorsque la place P_i est reliée à la transition T_j par l'arc inhibiteur A_k alors T_j est sensibilisé si le marquage de P_i est inférieurs au poids de A_k, et aucune marque n'est ôtée de P_i.

2.7.2 RdPs colorés :

La modélisation d'un système réel peut mener à des réseaux de Petri de taille trop importante rendant leur manipulation et leur analyse difficile. La question est alors de modifier (étendre) la modélisation par RdP de façon à obtenir des modèles

PDF created with pdfFactory Pro trial version www.pdffactory.com

plus petits. Cette question a motivé l'introduction des RdP colorés. Une taille trop importante peut découler du fait que l'on ne peut pas distinguer entre les différentes marques d'une place. Plusieurs marques dans une place peuvent modéliser un certain nombre de pièces identiques dans un stock. Si le stock contient plusieurs types de pièces, des places supplémentaires doivent être introduite pour sa modélisation. Intuitivement, si on est capable de distinguer les différentes marques les unes des autres, on pourrait associer à chaque type de marques un type de pièce. Pour distinguer les marques les une des autres, on introduit la notion de couleurs.

2.7.3 RdPs non autonomes :

2.7.3.1 RdPs temporisés :

Le comportement de ces modèles dépend du temps. Ils permettent d'évaluer le temps d'exécution d'une série d'opérations par exemple dans un procédé de fabrication industriel en fonction de la capacité des machines et des stocks. Il existe des RdPs P-temporisés et des RdPs T-temporisés. Dans un RdP P-temporisé par exemple, quand une marque est déposée dans une place P_i, elle reste indisponible pendant un temps Tempo(P_i), une fois ce temps écoulé elle devient disponible.

2.7.3.2 RdPs synchronisés :

Dans la modélisation RdP, le fait qu'une transition soit franchissable indique que toutes les conditions sont réunies pour qu'elle soit effectivement franchie. Le moment où se produira le franchissement n'est pas connu. Un RdP synchronisé est un RdP ou chaque transition est associée à un événement, la transition sera alors franchie si elle est validée mais quand l'événement associé se produit. Généralement un événement est un front montant ou descendant d'une variable logique. Un événement n'a pas de durée il est caractérisé par l'instant ou il se produit.

Pour plus de détails se référer à [1], [3], [15].

PDF created with pdfFactory Pro trial version www.pdffactory.com

3. Algèbre des dioïdes

La commande supervisée des SED par les réseaux de Petri a élaborée des lois de commande optimales et efficaces, qui garantissent le respect des spécifications sur le marquage. Cependant, elle ne prend pas en compte explicitement l'influence du temps, elle considère que les événements peuvent avoir lieu à des moments arbitraires, ce qui est contraire à la réalité, le fonctionnement de la plupart des processus industriels exige un certain nombre de contraintes temporelles, telles que les durées opératoires et les dates de début et de fin des taches. D'où l'intérêt de la modélisation de ces processus par des graphes d'événements temporisés, dont le comportement est représenté par des systèmes d'équations linéaires définies dans l'algèbre des dioïdes : Max-plus ou Min-plus, ainsi que la possibilité d'évaluation des performances et d'optimisation.

3.1 Définitions [16]

Définition1.2 : On appelle un monoïde, un ensemble M muni d'une loi interne $\oplus$ associative et qui possède un élément neutre ε tel que :

$$\forall m \in M, m \oplus \varepsilon = \varepsilon \oplus m = m.$$

Le monoïde est commutatif si la loi $\oplus$ est commutative.

Définition 1.3 : on appelle demi-anneau un ensemble D muni de deux lois internes $\oplus$ et $\otimes$ tel que :

- (D, $\oplus$) est un monoïde commutatif dont l'élément neutre est appelé élément nul.
- (D, $\otimes$) est un monoïde. Son élément neutre est appelé unité et est noté e.
- La multiplication $\otimes$ est distributive à droite et à gauche par rapport à la loi $\oplus$ $\forall a, b, c \in M$:

$c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$ et $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$.

- L'élément nul ε est absorbant pour la loi $\otimes$: $\forall a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$.

Si la loi additive $\oplus$ est idempotente : $\forall a \in D, a \oplus a = a$ alors (D, $\oplus$, $\otimes$) est appelé un demi-anneau idempotent ou dioïde.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Exemple 1.5:

$(\Re \cup \{-\infty, +\infty\}, \max, +)$ est un dioïde commutatif pour lequel $\varepsilon = -\infty$ et e=0, ce dioïde noté $\Re_{\max}$ est appelé algèbre (max,+). Dans ce dioïde, la loi $\oplus$ correspond à l'application max et la loi $\otimes$ est la somme usuelle.

$(\Re \cup \{+\infty, -\infty\}, \min, +)$ est un dioïde commutatif pour lequel $\varepsilon = +\infty$ et e=0, il est appelé algèbre (min,+). Dans ce dioïde, la loi $\oplus$ correspond à l'application min et la loi $\otimes$ est la somme usuelle.

3.2 Résolution d'équations dans les dioïdes

Par analogie avec la théorie des systèmes linéaires classiques. Le comportement dynamique du graphe d'événements temporisés est décrit par une équation d'état définie dans l'algèbre des dioïdes **:**

$X = A \otimes X \oplus B$

Remarque : la loi $\otimes$ est remplacée par un point.

La quantité $(A^*.B)$ est la plus petite solution de l'équation $X = A.X \oplus B$.

Avec $A^* = \bigoplus_{i \in N} A^i$ et $A^0 = I_n$ (Etoile de Kleene).

3.3 Fonctions compteurs et fonctions dateurs

L'évolution du graphe d'événement peut être représentée dans le domaine temporel par les fonctions compteurs, et dans le domaine événementiel par les fonctions dateurs.

3.3.1 Fonctions compteurs : on associe à chaque transition du graphe considéré une fonction du temps correspondant au nombre cumulé de franchissements de la transition à l'instant t. cette fonction est appelée compteur.

PDF created with pdfFactory Pro trial version www.pdffactory.com

3.3.2 Fonctions dateurs : on associe à chaque transition du graphe considéré une fonction de la variable $k \in N^*$, correspondant à la date du $k^{ième}$ franchissement de la transition, cette fonction est appelée dateur.

3.4 Evaluation des performances et optimisation

L'évaluation des performances s'effectue à partir du calcul d'indices de performance du système. Ces indices sont essentiellement

le taux de production λ, et le temps de cycle $T_C = \frac{1}{\lambda}$.

Quant à l'optimisation, les deux critères à optimiser sont le coût de fonctionnement et la productivité. Le problème d'optimisation des ressources consiste à minimiser le nombre des ressources de production qu'il faut engager initialement pour atteindre la productivité souhaitée. En d'autres termes, on calcule le marquage initial d'une ou de plusieurs places pour atteindre le taux de production fixé.

Conclusion :

Dans ce chapitre, des outils de modélisation, d'analyse et de commande de systèmes à événements discrets sont présentés. Nous avons introduit, dans un premier temps, les automates à états finis et les langages formels puisque la théorie de la supervision telle qu'initialement présentée, se base sur ces outils. L'outil réseau de Petri est par la suite étudié, pour pouvoir présenter les différentes extensions à la théorie de base introduite par Ramadge et Wonham.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 2

Méthodes de synthèse de superviseurs pour les systèmes à événements discrets

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 2

Méthodes de synthèse de superviseurs pour les systèmes à événements discrets.

2.1 Introduction

Les SED sont de plus en plus complexes, ce qui rend plus difficile la réalisation d'un système de commande efficace et réaliste. Plusieurs travaux ont été consacrés au problème de la synthèse de commandes pour les SED. L'objectif de la synthèse est de déterminer un superviseur garantissant le respect des spécifications.

L'approche proposée par Ramadge et Wonham s'intéresse à l'existence et à la synthèse d'un superviseur le plus permissif pour les SED. Cette approche est basée sur la modélisation des systèmes par des automates à états finis, et des langages formels. Cependant le manque de structure dans ces modèles limite le développement des algorithmes de calcul efficaces pour l'analyse et la synthèse de superviseur [11].

C'est pourquoi plusieurs méthodes de synthèse de commande basées sur les réseaux de Petri ont été proposées, pour exploiter la puissance de modélisation offerte par cet outil [2], [5], ainsi que les nombreux résultats mathématiques disponibles pour la vérification du comportement du système, tels que l'atteignabilité, la bornitude et la vivacité.

2.2 Synthèse de superviseur basée sur les automates et les langages

2.2.1 Concept de supervision :

Le modèle d'un procédé peut être vu comme un générateur spontané de mots d'événements sans control externe. En vue de commander un tel procédé dans le cadre de la théorie de la supervision, certains événements sont interdits ou autorisés. Ainsi, lorsque le procédé se trouve dans un état q_i, il reçoit de la part du superviseur une liste γ^i d'événements autorisés. La prochaine évolution du procédé se fera alors sur occurrence d'événement σ^{i+1} de γ^i. A partir du nouvel état q_j atteint par le procédé, une nouvelle liste

PDF created with pdfFactory Pro trial version www.pdffactory.com

d'événements autorisés est fournie. Ce principe d'inhibition / autorisation est représenté en figure 2.1 et est appelé **supervision**, et le fonctionnement qui en découle représente le comportement en boucle fermée du procédé.

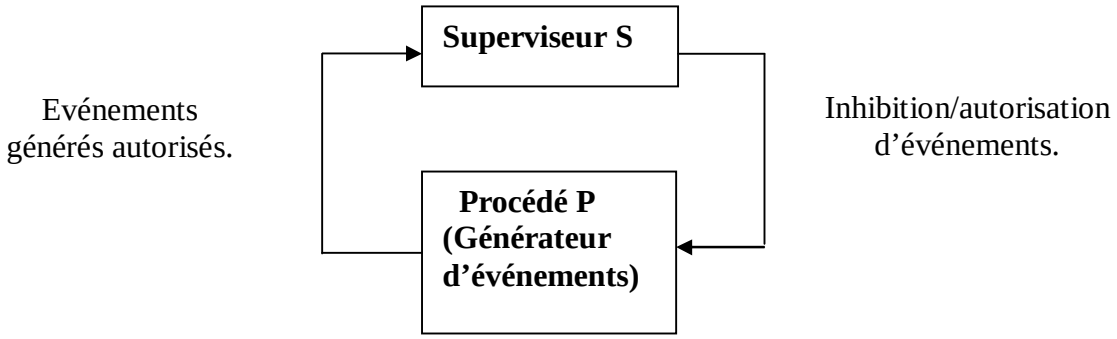


Figure 2.1 principe de supervision.

Remarque :

Ramadge et Wonham ont initié une théorie de synthèse automatique de superviseur pour les SED. Cette théorie considère l'aspect qualitatif des actions (via l'ordre d'occurrence d'événements), et permet au superviseur d'interdire l'apparition de certains événements (contrôlables) au procédé. L'aspect quantitatif des actions fait intervenir le concept d'événement forcé, la supervision devient alors commande supervisée puisque le superviseur peut forcer l'occurrence de certains événements du procédé.

2.2.2 Notion de contrôlabilité

2.2.2.1 Evénements contrôlables et événements non contrôlables :

L'ensemble Σ des événements d'un procédé est divisé en deux sous ensembles disjoints Σ_c et Σ_u, représentants respectivement l'ensemble des événements contrôlables, et l'ensemble des événements non contrôlables. Les événements contrôlables (Σ_c) sont les événements sur lesquels le superviseur peut avoir une action directe, c'est-à-dire, qu'il peut les interdire ou les autoriser. Par contre, le superviseur ne peut exercer aucune influence directe sur les événements dits non contrôlables (Σ_u). Tous les événements de Σ_u sont toujours autorisés quelque soit l'événement généré par le procédé, la condition $\Sigma_u \subseteq \gamma^i$ est toujours vraie pour tout i.

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.2.2.2 Condition de contrôlabilité [19]

Une spécification permet de traduire les contraintes imposées par le cahier des charges. Etant donné un procédé P défini par le langage L(P), et une spécification définie par un langage K.

Le but est de synthétiser un superviseur V qui permet de faire respecter cette spécification. Il convient d'abord de déterminer si le langage de la spécification K est admissible par le système, autrement dit si le système peut le respecter : c'est la notion de contrôlabilité.

Un langage $K \subseteq \Sigma^*$ est dit contrôlable si :

$$\overline{K}.\Sigma_u \cap L(P) \subseteq \overline{K}. \qquad 2.1$$

C'est-à-dire que K est contrôlable s'il n'existe pas de mot wσ, avec $w \in \overline{K}$, et $\sigma \in \Sigma_u$, telle que $w\sigma \notin \Sigma_u$: l'occurrence d'un événement non contrôlable ne fait pas sortir le système de la spécification.

2.2.3 Définition formelle d'un superviseur [21]

Soit P = (Q, Σ, δ, q_0, Q_m) un procédé avec $\Sigma = \Sigma_c \cup \Sigma_u$. l'ensemble des événements autorisés qui inclus nécessairement Σ_u, est appelé l'ensemble de contrôle noté Γ. Il est composé de l'ensemble des entrées de contrôle notées γ :

$$\Gamma = \{ \gamma \notin 2^{\Sigma} / \Sigma_u \subseteq \gamma \}. \qquad 2.2$$

On définit le superviseur comme une fonction V : L(P) → Γ qui attribut à chaque événement généré w, une entrée de contrôle γ = V(w) devant être appliquée. On note V/P le procédé supervisé par V.

On appelle fonctionnement en boucle ouverte, le comportement de P seul et le fonctionnement en boucle fermée, celui de P supervisé par V, V/P. Le langage $L(V/P) \subseteq L(P)$ est défini :

$$\begin{cases} \varepsilon \in L(V/P) \\ w\sigma \in L(V/P) \text{ si et seulement si } w \in L(V/P),\ \sigma \in V(w) \text{ et } w\sigma \in L(P). \end{cases} \qquad 2.3$$

Le langage $L_m(V/P)$ (comportement marqué) est définit :

PDF created with pdfFactory Pro trial version www.pdffactory.com

$$L_m(V/P) = L(V/P) \cap L_m(P). \qquad 2.4$$

2.2.4 Existence d'un superviseur [19]

Théorème 2.1 : Soit $K \subset L(P)$ le langage généré par P.

Il existe un superviseur V tel que L(V/P) = K si et seulement si :

K est préfixe clos : $K = \overline{K}$.

K est contrôlable : $\overline{K} . \Sigma_u \cap L(P) \subseteq \overline{K}$.

Théorème 2.2 : Soit $K \subset L_m(P)$ le langage marqué de P.

Il existe un superviseur V non bloquant ($L(V/P) = \overline{K}$) tel que $L_m(V/P) = K$ si et seulement si :

K est L_m-clos : $K = \overline{K} \cap L_m(P)$.

K est contrôlable : $\overline{K} . \Sigma_u \cap L(P) \subseteq \overline{K}$

Un langage $K \subset L$ est dit L-clos si $K = \overline{K} \cap L$: tout préfixe de K, qui est aussi un mot de L, est également mot de K.

2.2.5 Langage suprême contrôlable

Si K n'est pas contrôlable, alors on doit chercher un sous ensemble de K appelé Sup C(K) qui a la propriété de contrôlabilité en respectant toutes les contraintes. Il représente le plus grand langage contrôlable dans K.

Le superviseur est dit maximum permissif ou optimal lorsque, couplé avec le procédé, il génère le langage Sup C(K) tel qu'il est donné à la figure 2.2

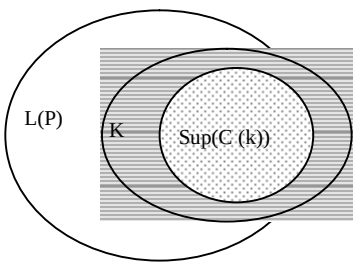


Fig. 2.2. Langage suprême contrôlable d'un fonctionnement désiré

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.2.5.1 Algorithme de Kumar [22]

Cet algorithme permet de déterminer le Sup C(K). Il est constitué des 4 étapes suivantes :

Pas 1 : construire le composé synchrone de L(P) et K.

Pas 2 : déterminer les états défendus du composé synchrone.

On appelle état défendu tout état (q,x) du composé synchrone tel qu'il existe un événement non contrôlable σ ou δ(q,σ) est définie pour P et ξ(x,σ) n'est pas définie pour K.

Pas 3 : déterminer les états faiblement défendus du composé synchrone. On appelle état faiblement défendu tout état (q,x) du composé synchrone qui n'est pas un état défendu, et tel qu'il existe une séquence d'événements non contrôlables S_u qui conduit à un état défendu.

Pas 4 : supprimer du composé synchrone les états défendus, les états faiblement défendus, ainsi que toutes les transitions liées à ces états.

2.2.6 Synthèse du superviseur

Le superviseur sera construit à partir de l'automate associé au langage Sup C(K) résultant des étapes vues ci-dessus, en déterminant les entrées de contrôle à appliquer au procédé.

Exemple 2.1:

Reprenons l'exemple 1.4, Le fonctionnement de notre système manufacturier doit respecter la présence d'un stock de capacité limitée à 1, situé entre les 2 machines (figure 2.3).

Fig. 2.3. Le système manufacturier sous la contrainte de

Le superviseur *S* de la figure 2.4 permet de garantir le respect de cette spécification. Dans cet automate les états v_0 et v_1 correspondent respectivement aux états : "stock vide" et "stock plein".

PDF created with pdfFactory Pro trial version www.pdffactory.com

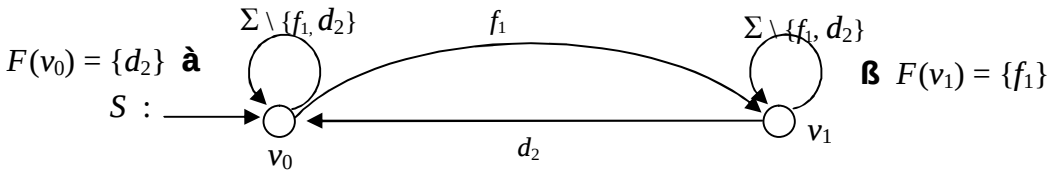


Fig. 2.4. Modèle automate de la spécification

Lorsque le stock est vide, l'occurrence de l'événement contrôlable d_2 est interdit (début du cycle de M_2). Sur l'occurrence de l'événement f_1 (fin du cycle de M_1 et dépôt d'une pièce dans le stock), l'automate S change d'état et passe dans l'état v_1. Dans cet état, l'occurrence de l'événement f_1 est interdite (fin du cycle de M_1).

Le modèle automate reconnaissant le fonctionnement en boucle fermée est obtenu en effectuant le composé synchrone de P et de S.

Le modèle de fonctionnement désiré en boucle fermé du système supervisé est donné dans la figure 2.5.

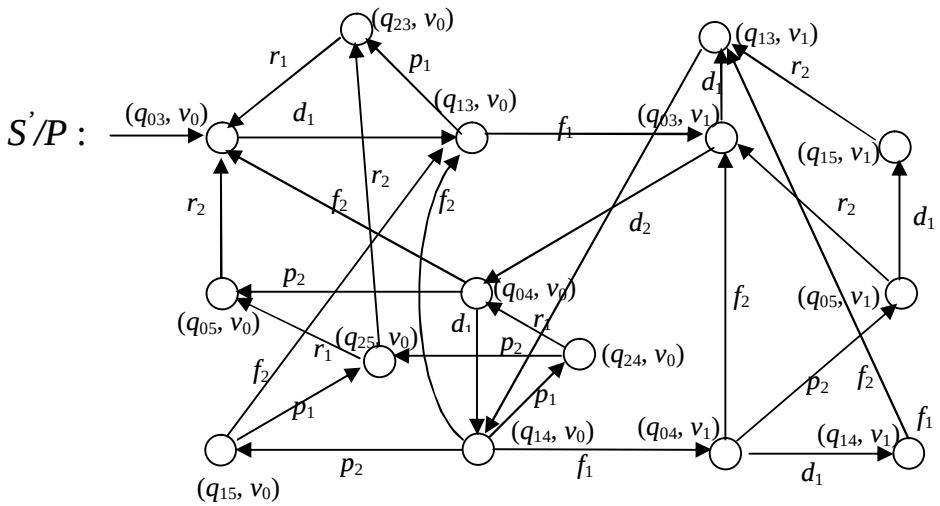


Fig. 2.5. Modèle de fonctionnement désiré en boucle fermée

Par application de l'algorithme de Kumar pour notre exemple, nous trouvons les états interdits suivants : { (q_{13},v_1), (q_{14},v_1), (q_{15},v_1) }. Dans cet exemple il n'y a pas d'états faiblement interdits (figure 2.6).

PDF created with pdfFactory Pro trial version www.pdffactory.com

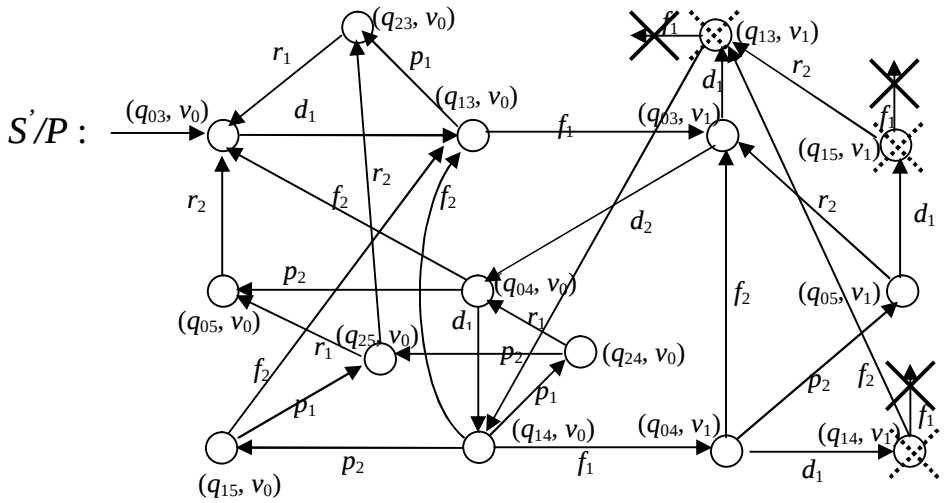


Fig. 2.6. Modèle automate du système supervisé avec des états interdits

Le modèle final de cet automate est présenté dans la figure 2.7.

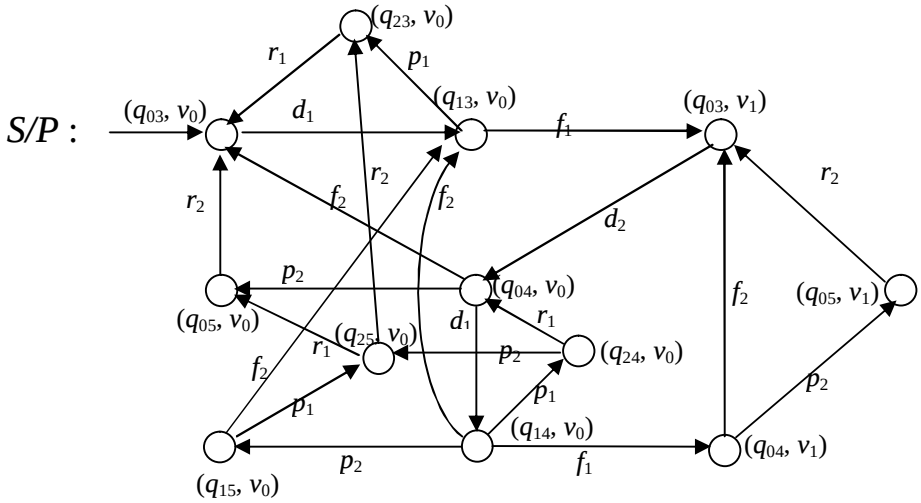


Fig. 2.7. Modèle final de système supervisé sans états interdits

2.3 Synthèse de superviseur basé sur les RdPs

Comme les systèmes réels sont devenus de plus en plus complexes, et en raison des limitations que présentent les modèles basés sur les automates et les langages, telles que la difficulté de modélisation, l'explosion combinatoire et la non garantie d'absence de blocage, le besoin d'un outil tel que les RdPs est devenu plus important. Les RdPs sont apparus comme l'outil le plus prometteur pour faciliter de tels modèles, de plus c'est un outil qui facilite l'implémentation sur un automate programmable industriel API.

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.3.1 Synthèse de superviseur par la méthode des invariants de marquage [4]

Cette méthode est appliquée pour les systèmes modélisés par les RdP, et dont les spécifications (contraintes) peuvent être exprimées en égalités, inégalités ou expressions logiques, et peuvent comporter des éléments du vecteur de marquage et / ou du vecteur de franchissement.

Les contraintes exprimées par le vecteur de franchissement doivent être transformées pour ne comporter que des éléments du vecteur de marquage pour pouvoir utiliser le principe des P- invariants pour la synthèse du superviseur.

Le superviseur est composé de places de contrôle qui sont connectées aux transitions de l'RdP du procédé de sorte à garantir que le système ne tombe pas dans les états interdits. La taille du contrôleur est proportionnelle au nombre de contraintes à satisfaire.

2.3.1.1 Principe des invariants de places (P-invariants) :

Les invariants est l'une des propriétés structurelles des RdPs (càd qui dépendent seulement de la structure topologique du RdP et non de son marquage initial). Il existe deux types d'invariants : les P-invariants et les T-invariants.

Les P-invariants sont des ensembles de places dont le nombre de jetons reste toujours constant. Le P-invariant est défini par le vecteur X(1xn), dont les éléments non nuls correspondent aux places appartenant au P-invariant, (avec n nombre de places du réseau) soit à partir du marquage initial m_0 :

$$m^T . X = m_0^T . X \qquad 2.5$$

Ou bien à partir de la matrice d'incidence W :

$$X^T . W = 0 \qquad 2.6$$

2.3.1.2 Description de la méthode :

Cette méthode exige que le procédé à contrôler soit modélisé par un RdP, et permet de construire un contrôleur par réseaux de Petri lié au procédé.

Les contraintes qui doivent être satisfaites peuvent être soit des expressions logiques, des égalités ou des inégalités.

Supposons que le système à commander est modélisé par un RdP ayant n places et m transitions, et doit satisfaire la spécification suivante :

PDF created with pdfFactory Pro trial version www.pdffactory.com

$$m_i + m_j \leq 1 \qquad 2.7$$

où m_i et m_j sont les marquages des places P_i et P_j respectivement. Cette contrainte veut dire qu'au plus, une de ces deux places soit marquée (les deux places ne peuvent être marquées en même temps). Cette inégalité peut être transformée en une égalité en introduisant une nouvelle variable m_s, et la contrainte devient alors :

$$m_i + m_j + m_s = 1 \qquad 2.8$$

La variable m_s représente dans ce cas une nouvelle place P_S qui reçoit l'excès de marques, ainsi on assure que la somme des marques m_i et m_j est toujours inférieure ou égale à 1. Cette place appartient au réseau contrôleur.

La structure de ce réseau sera calculée en notant que l'introduction de la variable m_S va introduire un P-invariant. Puisque une nouvelle place a été ajoutée au réseau, alors la matrice d'incidence W du système contrôlé global est la matrice originale D_P (n x m) augmentée d'une ligne correspondant à la place introduite P_S. Cette nouvelle ligne appartient à la matrice d'incidence du contrôleur notée W_C. Le problème peut être situé en général comme suit :

Toutes les contraintes peuvent être groupées et écrites sous forme matricielle :

$$L \cdot m_p \leq b \qquad 2.9$$

Où m_p est le vecteur de marquage modélisant le procédé, L est une matrice (n_C x n), b est un vecteur (n_C x 1) et n_C est le nombre de contraintes.

De la même manière toutes les équations de P-invariants générés après l'introduction de la variable m_S peuvent être groupées sous forme matricielle :

$$L \cdot m_p + m_c = b \qquad 2.10$$

où m_C est le vecteur (n_C x 1) qui représente le marquage des places du contrôleur.

L'équation matricielle suivante est l'équation P-invariant de tous les invariants définis par l'équation précédente :

$$X^T.W = [\, L \;\; I \,] \begin{bmatrix} Wp \\ Wc \end{bmatrix} = 0.$$

$L.W_P + W_C = 0$

PDF created with pdfFactory Pro trial version www.pdffactory.com

$W_C = -L.W_P$ 2.11

Où I est la matrice identité (n_C x n_C) puisque les coefficients de la variable m_S dans les contraintes sont égaux à 1.

La matrice W_C contient les arcs qui connectent les places du contrôleur aux transitions du réseau du procédé.

Le marquage initial du réseau contrôleur doit être calculé d'après l'équation des P-invariants

$L \cdot m_p + m_c = b$:

On a : $L \cdot m_{p0} + m_{c0} = b$ d'où

$$m_{c0} = b - L \cdot m_{p0} \quad 2.12$$

Exemple 2.2:

Considérons à titre d'exemple le réseau de la figure 2.8 suivante:

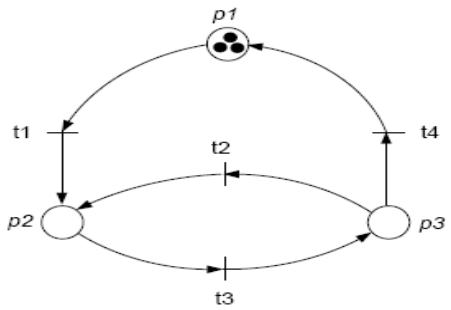


Figure 2.8 réseau du système à superviser

La matrice d'incidence de ce réseau $W_P = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 \end{bmatrix}$

Son marquage initial $m_{p0} = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}$

L'objectif est de contrôler le réseau afin que les places P_2, P_3 ne contiennent jamais plus d'une marque, c'est à dire, on désire appliquer la contrainte : $m_2 + m_3 \leq 1$. L = [0 1 1] , b = 1.

On introduit la variable m_S et l'inégalité devient une égalité : $m_2 + m_3 + m_s = 1$

PDF created with pdfFactory Pro trial version www.pdffactory.com

Cette équation représente l'invariant X = $[0 \quad 1 \quad 1 \quad 1]^T$

La matrice d'incidence contrôleur : $W_C = - L.W_P = [-1 \quad 0 \quad 0 \quad 1]$

Le marquage initial du contrôleur : $m_{C0} = 1 – L. m_{p0} = 1.$

La structure du réseau contrôlé est décrite par la matrice d'incidence :

$$W = \begin{bmatrix} W_P \\ W_C \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

Tandis que son marquage initial : $m_0 = \begin{bmatrix} m_{P0} \\ m_{C0} \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

Le réseau de Petri du système contrôlé est donné par la figure 2.9 suivante :

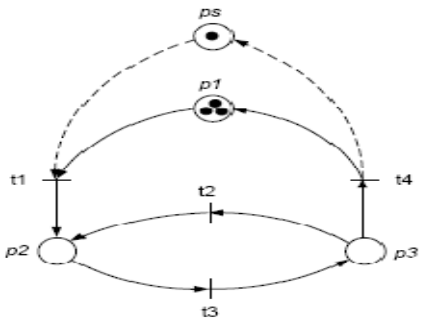


Figure 2.9 Réseau du système supervisé.

2.3.1.3 Transformation de RdP :

Cette méthode transforme les contraintes qui contiennent des éléments du vecteur de franchissement. Elle est basée sur la transformation du modèle RdP lui-même. Supposons que la contrainte à satisfaire est de la forme :

$m_i + q_j \leq 1$ (q_j élément du vecteur de franchissement).

Cela veut dire que la transition T_j ne peut pas être franchie si la place P_i est marquée et vice versa. La transformation est la suivante : la transition t_j est remplacée par deux transitions et une place entre elles comme le montre la figure 2.10 suivante :

PDF created with pdfFactory Pro trial version www.pdffactory.com

Figure 2.10 Transformation de RdP

Par conséquent la matrice W_P est augmentée d'une ligne et d'une colonne. Cette transformation est artificielle, et n'ajoute ou ne soustrait rien du modèle, son seul but est d'introduire la place P'_j qui enregistre le franchissement de la transition t_j. Le marquage m'_j de P'_j remplace q_j dans la contrainte $m_i + q_j \leq 1$ qui devient :

$$m_i + m'_j \leq 1. \qquad 2.13$$

La contrainte peut maintenant être transformée en égalité en introduisant m_S, et le contrôleur peut être calculé comme déjà vu précédemment.

Après le calcul de la structure du contrôleur, la transition et la place ajoutées sont supprimées, et la $j^{ème}$ et $(j+1)^{ème}$ colonne de W sont sommées pour produire la colonne correspondant à t_j, et la ligne correspondant à la place P'_j est supprimée.

Exemple 2.3 :

Considérons encore le réseau de l'exemple précédant. Supposons que la contrainte cette fois est : $q_2+q_3 \leq 1$ càd que les transitions t_2 et t_3 ne peuvent être franchies en même temps.

Chacune des deux transitions est remplacée par deux transitions et une place entre elles comme le montre la figure 2.11 suivante :

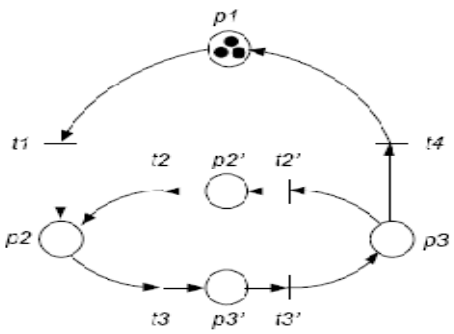


Figure 2.11 Réseau transformé

PDF created with pdfFactory Pro trial version www.pdffactory.com

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

$$m_{p0} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

La contrainte peut maintenant être exprimée en fonction du marquage de P'_2 et P'_3 : $m'_2 + m'_3 \leq 1$.

$L = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \end{bmatrix}$, $b = 1$.

On introduit la variable m_S qui traduit le marquage de la place du contrôleur P_S :

$m'_2 + m'_3 + m_s = 1$.

$W_C = -L \cdot W_P = \begin{bmatrix} 0 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$, $m_{C0} = 1 - L \cdot m_{p0} = 1$.

$$W = \begin{bmatrix} W_P \\ W_C \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

$$m_0 = \begin{bmatrix} m_{P0} \\ m_{C0} \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Le réseau contrôlé est donné par la figure 2.12 suivante:

PDF created with pdfFactory Pro trial version www.pdffactory.com

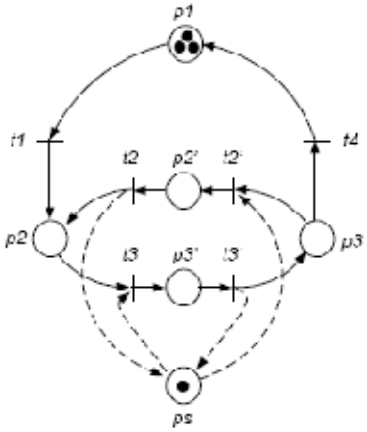


Figure 2.12 Réseau transformé supervisé

Une fois le contrôleur calculé : la 2ème et la 3ème colonne de W sont sommées pour produire la colonne correspondant à t_2, alors que la 3ème et la 4ème colonne de W sont sommées pour produire la colonne correspondant à t_3. Et les lignes 4 et 5 correspondant aux deux places de transformation sont supprimées, alors la structure du RdP original contrôlé est :

$$W = \begin{bmatrix} W_P \\ W_C \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$m_0 = \begin{bmatrix} m_{P0} \\ m_{C0} \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

et son réseau de Petri est donné par la figure 2.13.

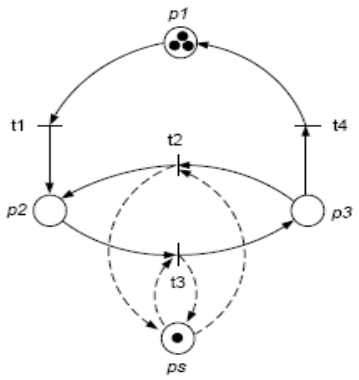


Figure 2.13 Réseau original supervisé.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Remarque :

La méthode telle qu'initialement proposée, est applicable que dans le cas où toutes les transitions sont contrôlables.

Dans le cas de présence de transitions non contrôlables, une méthode a eté proposée (Basile et al, 2006), ce cas sera étudié au chapitre suivant.

2.3.2 Synthèse de superviseur par la méthode des régions : [6]

L'approche de supervision par la théorie des régions a été proposée par A. Ghaffari et N. Rezk. Le système est modélisé par un Rdp borné et le superviseur à synthétiser doit être maximum permissif en respectant les spécifications (états interdits), la vivacité du réseau, et les transitions non contrôlables.

Le superviseur résultant est un ensemble de places de contrôle à ajouter au réseau initial, qui interdisent l'accès aux états interdits.

2.3.2.1 Description de la méthode

Cette méthode est divisé en deux étapes, déterminer l'ensemble des marquages admissibles, puis construire des places de contrôle en utilisant la théorie des régions.

Etape 1 : Pour déterminer l'ensemble des marquages admissibles on suit les étapes suivantes :

- déterminer l'ensemble des marquages interdits.
- Générer le graphe des marquages partiel.
- Déterminer l'ensemble des marquages dangereux.
- Générer le graphe des marquages admissibles R_c.

Etape 2 : Cette étape construit les places de contrôle à ajouter au modèle initial à partir du graphe des marquages admissibles en utilisant la théorie des régions.

2.3.2.2 Synthèse du contrôleur

Etant donné le modèle du système à contrôler (N, M_0)et le graphe d'atteignabilité R_c (le comportement admissible du système contrôlé). On utilise la théorie des régions pour construire les places de control $\{P_c\}$à ajouter.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Considérons une place de control p_c, chaque marquage M dans le graphe des marquages admissibles R_c doit rester atteignable après l'ajout de p_c, ce qui implique que p_c doit satisfaire la condition d'atteignabilité :

$$M(p_C) = M_0(p_C) + W(p_C,\cdot)\vec{\Gamma}_M \geq 0 \qquad 2.14$$

où Γ_M est un chemin non orienté quelconque dans R_c de M_0 à M .

Chaque place p_c, doit satisfaire l'équation cycle pour les cycles dans R_c :

$$\sum_{t\in T}(p_C,t).\vec{g}(t) = 0 \quad \forall g \in S_C \qquad 2.15$$

où S_c est l'ensemble des cycles du graphe d'atteignabilité R_c, ainsi que la condition de séparation :

$$M_0(p_C) + W(p_C,.)\vec{\Gamma}_M + W(p_C,t) \leq 0 \qquad 2.16$$

Les relations 2.14 2.15 2.16 déterminent la place de control p_c

Exemple 2.4 :

Considérons encore l'exemple 1.4.

Figure 2.14 le système sous la contrainte de stock.

Les modèles Rdp des machines 1 et 2 et de la spécification sont donnés respectivement par a, b, et c de la figure 2.15.

PDF created with pdfFactory Pro trial version www.pdffactory.com

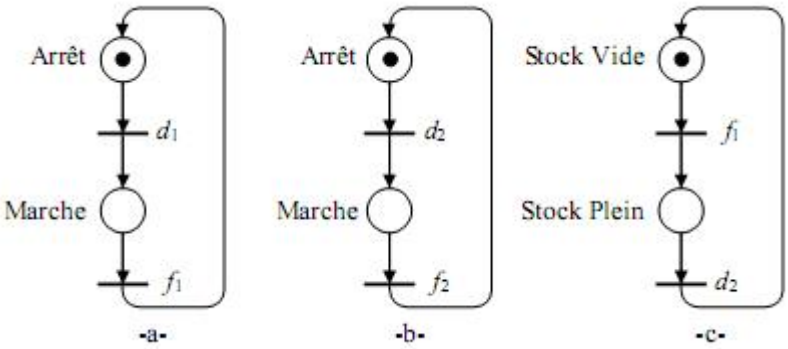


Figure 2.15 Modèle Rdp. a) Machine 1. b) Machine 2. c) spécification.

Le fonctionnement en boucle fermée du procédé global (M_1, M_2, et stock), nous donne le graphe de marquage de la figure 2.16. En boucle fermée, une transition est franchissable si elle est validée par rapport au procédé et validée par rapport à la spécification.

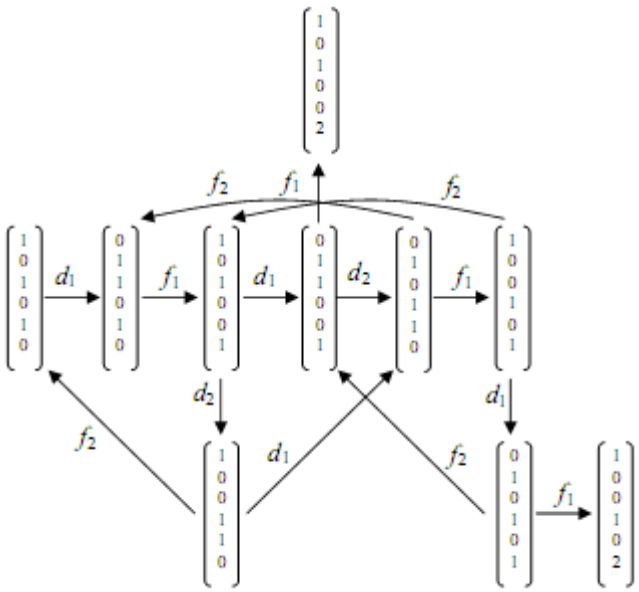


Figure 2.16 : graphe des marquages.

Comme la capacité du stock est de 1, alors l'ensemble des marquages interdits est {M_8, M_9}, où M_8= (1 0 0 1 0 2) et M_9= (1 0 1 0 0 2).

L'ensemble des marquages dangereux qui mènent aux marquages interdits est {M_3, M_6}.

L'ensemble des marquages admissibles est {M_1, M_2, M_4, M_5, M_7}.

Pour ne pas atteindre le marquage M_3, il faut interdire la transition d_1 à partir du marquage M_2, ainsi que pour ne pas atteindre le marquage M_6, il faut interdire la transition d_1 à partir du marquage M_5. Cela nous donne le graphe de marquages admissibles de la figure 2.18.

PDF created with pdfFactory Pro trial version www.pdffactory.com

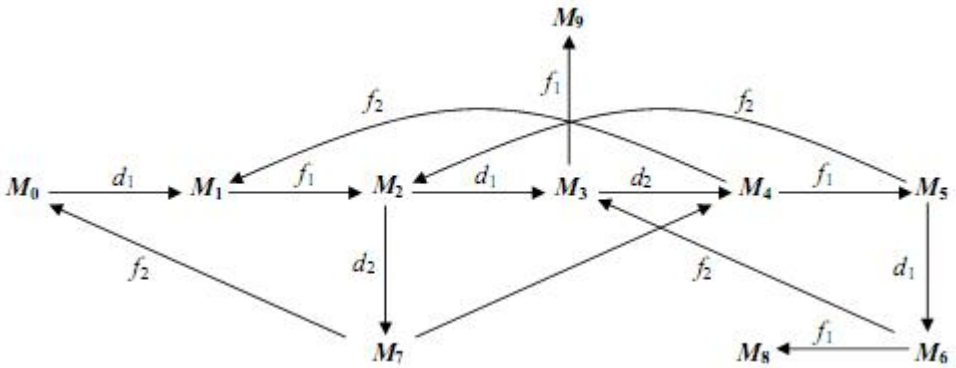


Figure2.17 : graphe des marquages partiel

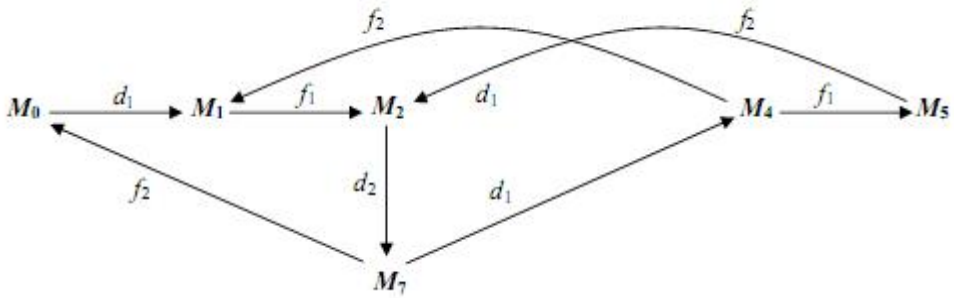


Figure 2.18 : graphe des marquages admissibles R_C.

Equations d'atteignabilités :

$M_0(p_c) \geq 0$

$M_1(p_c) = M_0(p_c) + W(p_c, d_1) \geq 0$

$M_2(p_c) = M_0(p_c) + W(p_c, d_1) + W(p_c, f_1) \geq 0$

$M_3(p_c) = M_0(p_c) + W(p_c, d_1) + W(p_c, f_1) + W(p_c, d_2) \geq 0$

$M_4(p_c) = M_0(p_c) + 2W(p_c, d_1) + W(p_c, f_1) + W(p_c, d_2) \geq 0$

$M_5(p_c) = M_0(p_c) + 2W(p_c, d_1) + 2W(p_c, f_1) + W(p_c, d_2) \geq 0$

Equation cycle :

$W(p_c, d_1) + W(p_c, f_1) + W(p_c, d_2) + W(p_c, f_2) = 0$

Equations de séparation d'événements :

$M_0(p_{c1}) + 2W(p_{c1}, d_1) + W(p_{c1}, f_1) \leq 0$

$M_0(p_{c2}) + 3W(p_{c2}, d_1) + 2W(p_{c2}, f_1) + W(p_{c2}, d_2) \leq 0$

PDF created with pdfFactory Pro trial version www.pdffactory.com

En résolvant le système précédant, on trouve la places de contrôle p_{C1} telle que $W(p_{C1},.) = (-1\ 0\ 1\ 0)$ et $M_0(p_{C1}) = 1$ interdisant la première transition, et p_{C2} telle que $W(p_{C2},.) = (-1\ 0\ 1\ 0)$ et $M_0(p_{C2}) = 1$ interdisant la deuxième transition.

On supprime la place p_{C2} car elle est redondante. Le modèle du système supervisé est donné dans la figure 2.19.

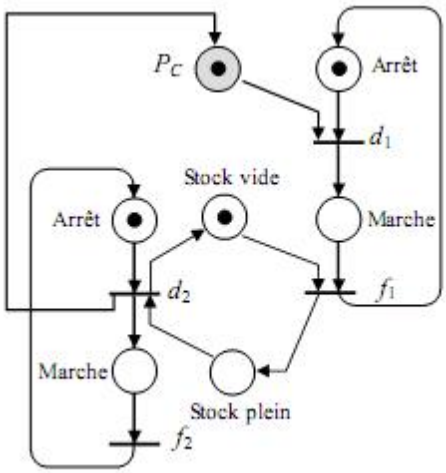


Figure 2.19 : modèle Rdp du système supervisé.

Conclusion :

Dans ce chapitre, trois méthodes de supervision sont introduites et discutées. La première approche est celle de ramadge et Wonham, utilisant les automates et les langages pour la synthèse de superviseur. Malgré la puissance de l'approche, elle souffre du problème d'explosion combinatoire souvent rencontré dans le cas de SED à grande dimension. La deuxième approche est basée sur les réseaux de Petri et utilise la notion de P-invariant pour synthétiser des places de supervision qui permettent de garantir le respect des spécifications imposées par un cahier des charges. Cette approche est intéressante parce qu'elle est structurelle. Elle ne prend malheureusement pas en charge les transitions non contrôlables. La troisième approche est due à (Rezk el al, 1999) et utilise la théorie des régions pour la synthèse des places de supervision. Elle se base sur la construction de l'espace d'atteignabilité du réseau de Petri modélisant le processus étudié.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 3

Synthèse de superviseur en présence de transitions non contrôlables

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 3

Synthèse de superviseur en présence de transitions non contrôlables.

1 Introduction :

Le principe de la supervision est d'interdire à un système donné d'atteindre un ensemble d'états interdits qui sont traduit part un ensemble de contraintes.

Avec les réseaux de Petri, interdire l'accès à ces états se traduit par l'interdiction de franchissement des transitions en amont de ces états. Lorsque ces transitions sont non contrôlables (spécifications non contrôlables), leur interdiction ne correspond pas avec la réalité physique du système, par exemple, interdire le franchissement d'une transition associée à l'événement panne.

Rencontrer des transitions non contrôlables dans un système est inévitable, d'où l'idée de restriction de l'ensemble des spécifications à un sous ensemble contrôlable et qui doit être le moins restrictif (le plus permissif) possible.

2 Supervision en présence de transitions non contrôlables :

La synthèse du superviseur restreignant le comportement en boucle fermée d'un système à $L \cap L(G)$, où L est le langage en boucle ouverte et L(G) la spécification, est possible si et seulement si deux conditions sont vérifiées : L est préfixe clos et commandable. Si L n'est pas commandable, on peut considérer une classe de sous langages préfixes clos et commandables de L, $\Omega(L) = \{\gamma \subseteq L / \gamma \text{ est préfixe clos et commandable}\}$, pour chaque langage γ de cette classe on peut construire un superviseur qui restreint le comportement en boucle fermée à $k \cap L(G) \subset L \cap L(G)$. La classe $\Omega(L)$ est fermée pour l'union des langages et non vide. Elle admet un unique élément suprême $L^{\uparrow} = \text{Sup}\,\Omega(L)$ appelé sous langage suprême commandable qui est la solution optimale du problème de commande.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Une approche similaire peut être envisagée en considérant l'évolution d'état du système plutôt que les traces des événements générés.

Cette approche est particulièrement intéressante quand le système est représenté par un réseau de Petri. Dans ce cas on suppose que certaines transitions, appelées contrôlables, peuvent être interdites par une action extérieure.

Considérons le système réseau de Petri (N,m_0) avec m places, dont l'ensemble des marquages atteignables est $R(N,m_0) \subseteq N^m$, supposons que nous avons un ensemble de marquages admissibles $\Gamma \subseteq N^m$, et considérons le problème de synthèse de superviseur qui restreint l'ensemble d'atteignabilité du système en boucle fermé à $\Gamma \cap R(N,m_0)$. Cela est possible si deux conditions sont vérifiées : Γ doit être contrôlable et son atteignabilité est admissible càd que les marquages admissibles doivent être atteint à travers une évolution d'état ne contenant pas de marquages interdits. Cette condition est équivalente à la propriété de préfixe clôture.

Si Γ n'est pas contrôlable, on peut considérer une classe de sous ensembles contrôlables de Γ, $\Omega(\Gamma) = \{\gamma \subseteq \Gamma / \gamma \text{ controlable}\}$. Pour chaque ensemble γ de $\Omega(\Gamma)$, on peut synthétiser un superviseur qui restreint l'ensemble d'atteignabilité du système en boucle fermé à $\gamma \cap R(N,m_0) \subset \Gamma \cap R(N,m_0)$. La classe $\Omega(\Gamma)$ est fermée et non vide, et admet un élément suprême unique $\Gamma^{\uparrow} = \text{Sup}\Omega(\Gamma)$ appelé sous ensemble suprême contrôlable qui est la solution optimale du problème de control dans le sens où elle autorise le plus large ensemble d'atteignabilité en boucle fermé.

Un intérêt particulier est donné pour le problème de control où l'ensemble des marquages admissibles Γ est exprimé par un ensemble de n_c inégalités linéaires appelées contraintes d'exclusion mutuelle généralisées (CEMG).

Dans ce cas on écrit $\Gamma = M(L,k) = \{m \in N^m / Lm \leq k\}, L \in Z^{nc\,x\,m}, k \in Z^{nc}$, les problèmes de ce type ont été étudiés par plusieurs auteurs (Giua,DiCesare et Silva,1992 ; Krogh et Holloway,1991 ; Li et Wonham,1994 ; Moody, Lemmon et Antsaklis,1996). Cette structure de l'ensemble des marquages admissibles a l'avantage que si Γ est contrôlable, le superviseur de cette classe de problèmes prend la forme d'autant de places qu'il y a de contraintes,

PDF created with pdfFactory Pro trial version www.pdffactory.com

appelées moniteurs. Chacune d'elles possède des arcs entrant et sortant de certaines transitions du réseau.

Supposons maintenant que Γ est non contrôlable, suivant l'approche précédente, on doit trouver l'ensemble $\Gamma^{\uparrow}$. La synthèse du superviseur devient complexe puisque on a affaire à des sous ensembles contrôlables.

Dans la plupart des cas cette structure spéciale de l'ensemble admissible est perdu (Giua et al 1992), $\Gamma^{\uparrow}$ ne peut pas être exprimé par un ensemble d'inégalités linéaires, le superviseur correspondant n'a pas la structure des places de supervision.

Ce problème a motivé Moody et al (1996) et Moody et Antsaklis (2000) à considérer une restriction de l'ensemble d'atteignabilité. L'idée est de trouver un sous ensemble $\Gamma^{\uparrow}$ tel que:

a) Il soit contrôlable par les conditions structurelles, la condition de contrôlabilité structurelle indique qu'il n y a pas d'arcs allant des places de supervision vers les transitions non contrôlables pour prévenir le franchissement de ces dernières.

b) Il peut être exprimé par un ensemble de n_c contraintes.

On peut définir l'ensemble :

$$\Omega_{nc}(\Gamma) = \left\{\gamma \subseteq \Gamma / \text{k est structurellement controlable}, \exists L' \in Z^{nc\ x\ m}, k' \in Z^{nc} : \gamma = M(L', k')\right\}.$$

Dans Moody et al (1996) et Moody et Antsaklis (2000), une procédure a été donnée pour calculer l'élément $\gamma \in \Omega_{nc}(\Gamma)$ et sa structure de moniteurs correspondant.

2.1 Définitions [14]

Définition 3.1

Considérons le réseau N, et les CEMG, on définit le réseau non contrôlable de N, noté N_u= (P, Tu, W^-_u, W^+_u) le réseau résultant de l'élimination de chaque transition contrôlable de N. Il est évident que $R(N_u, m) \subseteq R(N, m)$.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Définition 3.2

L'ensemble des marquages admissibles $\Gamma \subseteq N^m$ est comportementalement contrôlable pour un système $\langle N, m_0 \rangle$, avec le sous réseau non contrôlable N_u si $\mathbf{U}_{m \in \Gamma \cap R(N,m_0)} R(N_u, m) \subseteq \Gamma$

Γ est contrôlable si à partir de chaque marquage $m \in \Gamma$, il n y a pas de marquage interdit atteint par le franchissement de séquence contenant que des transitions non contrôlables.

Définition 3.3

Soit l'ensemble des marquages admissibles représenté par $\Gamma = M(L, k)$, et le réseau N ayant l'ensemble des transitions $T = T_c \cup T_u$, avec $T_c \cap T_u = \emptyset$.

Γ est structurellement contrôlable si $LW_u \leq 0$ où $W_u \in Z^{m \times n_u}$ est la matrice d'incidence du réseau non contrôlable N_u, avec n_u est le nombre de transitions non contrôlables.

2.2 Proposition 3.1 (Moody et al, 1996)

Si on peut trouver $R_1 \in N^{n_C \times m}, R_2 \in N^{n_C \times n_C}$ satisfaisant $[R_1 \quad R_2] \begin{bmatrix} mp_0 \\ Lmp_0 - (k+1) \end{bmatrix} \leq -1$ alors le contrôleur $W_c = -L'W_p$, $m_{C0} = k' - L'mp_0$ où $L' = R_1 + R_2 L$, $k' = R_2(k+1) - 1$ peut assurer que le système en boucle fermée vérifie $Lmp \leq k$.

2.3 Théorème 3.1 : (Basile et al, 2006)

Considérons un système représenté par un réseau de Petri $\langle N, m_0 \rangle$. Soit $\Gamma = M(L, k) = \{m \in \aleph^m \mid Lm \leq k\}$ un ensemble non contrôlable avec $L \in Z^{nc \times m}$ et $k \in Z^{nc}$, la classe $\Omega_{nc}(\Gamma)$ est :

a) non vide ;
b) non fermé par l'union.

Dans la partie a) du théorème, on peut montrer que $\Omega_{nc}(\Gamma)$ n'est pas vide car il contient l'ensemble vide. Cependant, si l'élément suprême de $\Omega_{nc}(\Gamma)$ est l'ensemble $\gamma = \emptyset$, alors le problème de contrôle n'a pas de solution, car la condition $m_0 \in \gamma$ n'est pas vérifiée.

PDF created with pdfFactory Pro trial version www.pdffactory.com

L'interprétation de la partie b) est que l'union des ensembles $\gamma_1, \gamma_2, ... \in \Omega_{nc}(\Gamma)$, $\gamma = \gamma_1 \cup \gamma_2 \cup ....$ n'est pas un ensemble convexe et ne peut pas être exprimé par un ensemble d'inégalités linéaires

3 Algorithme de paramétrisation linéaire des moniteurs : (Basile et al. (2006)).

L'algorithme consiste à construire la table $\begin{bmatrix} W_u & I_{mxm} & O_{mx1} \\ n & q & r_2 \end{bmatrix}$, où pour la première étape $n = l.W_u, q = 0_{1xm}, r_2 = 1.$

A chaque étape le but de l'algorithme est de rendre nuls les éléments positifs de n. Un élément positif de n correspond à une transition t_s, et l'élément négatif de la même colonne dans W_u correspond à une place p_r.

L'algorithme annule les éléments positifs de n notés $n(t_s)$ en sommant la dernière ligne de n multipliée par l'élément pivot noté $-W_u(p_r, t_s)$ avec la $r^{ième}$ ligne de W_u multiplier par $n(t_s)$. La dernière ligne de la table est remplacée par le résultat de l'addition.

Si on dénote (l', k') les contraintes transformées à l'étape courante de l'algorithme nous avons $l' = r_2 l + q,\ k' = r_2(k+1) - 1,\ \text{et}\ n = l' W_u$.

Par cette opération les composantes du vecteur l' relatifs à la place $p_r \in {}^{o}t_s$ sont augmentées d'une quantité positive, le résultat de la transformation de contraintes est que la place de contrôle dérivée de l' n'a aucun arc sortant vers t_s, mais un arc sortant vers la transition $t \in {}^{o}P_r$.

S'il existe plus d'un élément négatif dans la $s^{ième}$ colonne de W_u, différentes solutions sont alors obtenues.

Pour considérer toutes les solutions possibles, une variable symbolique α_i de valeur naturelle est introduite pour chaque élément négatif de la colonne $W_u(\cdot, t_s)$.

PDF created with pdfFactory Pro trial version www.pdffactory.com

La table est augmentée d'autant de lignes qu'il y a de combinaisons linéaires de la ligne contenant l'élément positif dans n et les lignes annulant cet élément dans W_u.

Ces lignes ajoutées sont définies par $[N(\alpha_i,\cdot) \quad P(\alpha_i,\cdot)]$ comme suit :

$N(\alpha_i,t)=\beta\cdot W_u(p_r,t_s)\,si\,t\neq t_s$, $N(\alpha_i,t)=0\,si\,t=t_s$ avec β le plus petit multiplieur commun entre les éléments négatifs de $W_u(\cdot,t_s)$.

$P(\alpha_i,p)=0\,si\,p\neq p_r$, $P(\alpha_i,p)=1\,si\,p=p_r$.

La table est alors augmentée comme suit : $\begin{bmatrix} W_u & I_{mxm} & O_{mx1} \\ n & q & r_2 \\ N & P & O_{Zx1} \end{bmatrix}$

Chaque ligne de $[N \;\; P \;\; O_{Zx1}]$ est relative à une des z variables symboliques.

A chaque étape $l'=l+q+\alpha^T P$, $l'W_u=n+\alpha^T N$, alors que les variables symboliques n'interviennent pas dans le calcul de k'.

Puisque $n=l'W_u$, alors annuler l'élément positif $n(t_s)$ revient à annuler $l'W_u(\cdot,t_s)$. La résolution de l'équation $l'W_u(\cdot,t_s)=0$ à chaque étape donnera le système d'équations $A\alpha=b$. Une nouvelle équation doit être ajoutée au système à chaque étape.

Après l'introduction des variables symboliques, le but de l'algorithme est maintenant de rendre nuls les éléments positifs de la portion $\begin{bmatrix} n \\ N \end{bmatrix}$.

L'algorithme s'arrête lorsque il n'existe plus d'éléments positifs dans n ou N, ou lorsque il n'existe plus d'éléments négatifs dans les colonnes de W_u pour rendre nuls les éléments positifs de n ou N.

Exemple 3.1 :

Considérons le réseau de Petri de la figure 3.1. Soit la contrainte sur le marquage suivante : $M(l,k)=\{m\in \aleph^m \mid m(p_3)\leq 1\}$. La place de control p_c ne satisfait pas la condition de contrôlabilité structurelle car il existe un arc (en discontinu) de p_c vers la transition non contrôlable t_4.

Dans ce qui suit, Les transitions en doubles traits représentent les transitions contrôlables, et celle en un trait représentent les transitions non contrôlables.

PDF created with pdfFactory Pro trial version www.pdffactory.com

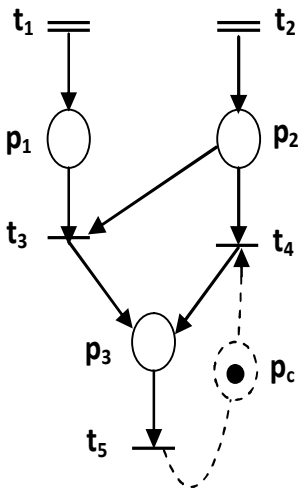


Fig 3.1 Réseau de Petri avec transitions non contrôlables

En appliquant l'algorithme de Basile et al, la procédure de transformation de la contrainte est donnée en figure 3.2

	$t_3 \quad t_4 \quad t_5$		
p_1 p_2 p_3	$-1 \quad 0 \quad 0$ $-1 \quad -1 \quad 0$ $1 \quad 1 \quad -1$	W_u	
	$1 \quad 1 \quad -1$	$n = lW_u$	$l = (0 \quad 0 \quad 1), \quad k = 1$
Step 1	$0 \quad 1 \quad -1$	n	$l' = (\alpha_1 \quad \alpha_2 \quad 1), \quad k' = 1$
α_1 α_2	$0 \quad 0 \quad 0$ $0 \quad 1 \quad 0$	N	$-\alpha_1 - \alpha_2 + 1 = 0$ $\alpha_1 + \alpha_2 = 1 \qquad \alpha_i \in \aleph^2$
Step 2	$0 \quad 0 \quad -1$	n	$l' = (\alpha_1 \quad \alpha_2 + \alpha_3 \quad 1), \quad k' = 1$
α_1 α_2 α_3	$0 \quad 0 \quad 0$ $0 \quad -1 \quad 0$ $-1 \quad 0 \quad 0$	N	$-\alpha_1 - \alpha_2 + 1 = 0$ $\alpha_1 + \alpha_2 = 1$ $\alpha_2 + \alpha_3 = 1 \qquad \alpha_i \in \aleph^3$

Fig 3.2 Etapes de l'algorithme de transformation de la contrainte $m(p_3) \leq 1$

PDF created with pdfFactory Pro trial version www.pdffactory.com

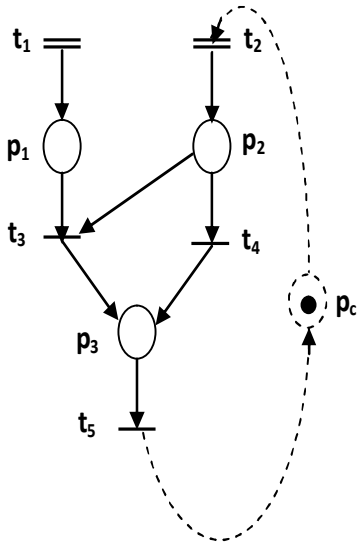


Fig 3.3 Réseau en boucle fermée.

La transition t_2 contrôle le franchissement de t_3 et t_4, alors que t_1 contrôle seulement le franchissement de t_3, d'où si p_1 est inclue dans la contrainte transformée, cela implique nécessairement que p_2 est inclue aussi.

Les deux solutions possibles sont :

$$\begin{cases} M(l^1,k^1)=\{m\in \aleph^m \mid m(p_2)+m(p_3)\le 1\}. \\ ou \\ M(l^2,k^2)=\{m\in \aleph^m \mid m(p_1)+m(p_2)+m(p_3)\le 1\}. \end{cases}$$

Remarquons que $M(l^2,k^2)\subset M(l^1,k^1)$ d'où $M(l^1,k^1)$ est la moins restrictive donc la solution optimale.

Si $M(l^1,k^1)$ et $M(l^2,k^2)$ sont incomparables, on peut choisir l'une des deux solutions. Cependant, si d'autres spécifications de contrôle sont présentes telles que la condition de vivacité par exemple, l'un des deux moniteurs est préférable que l'autre pour avoir un contrôle optimal.

PDF created with pdfFactory Pro trial version www.pdffactory.com

4 Application sur un système manufacturier

Dans cette section, nous appliquons la méthode de transformation de contrainte à un système industriel d'assemblage.

Les produits sont transportés par des palettes libres, chaque palette ne peut transporter qu'un seul produit à la fois.

Le système peut être décomposé en trois classes d'éléments (Figure 3.4) :

1. Le magasin : il permet le stockage des palettes qui ne sont pas en production.
2. La boucle centrale de transfert : elle permet le transport des palettes entre les postes de travail.
3. Les quatre postes de travail en dérivation : ils permettent la transformation des produits transportés par les palettes.

PDF created with pdfFactory Pro trial version www.pdffactory.com

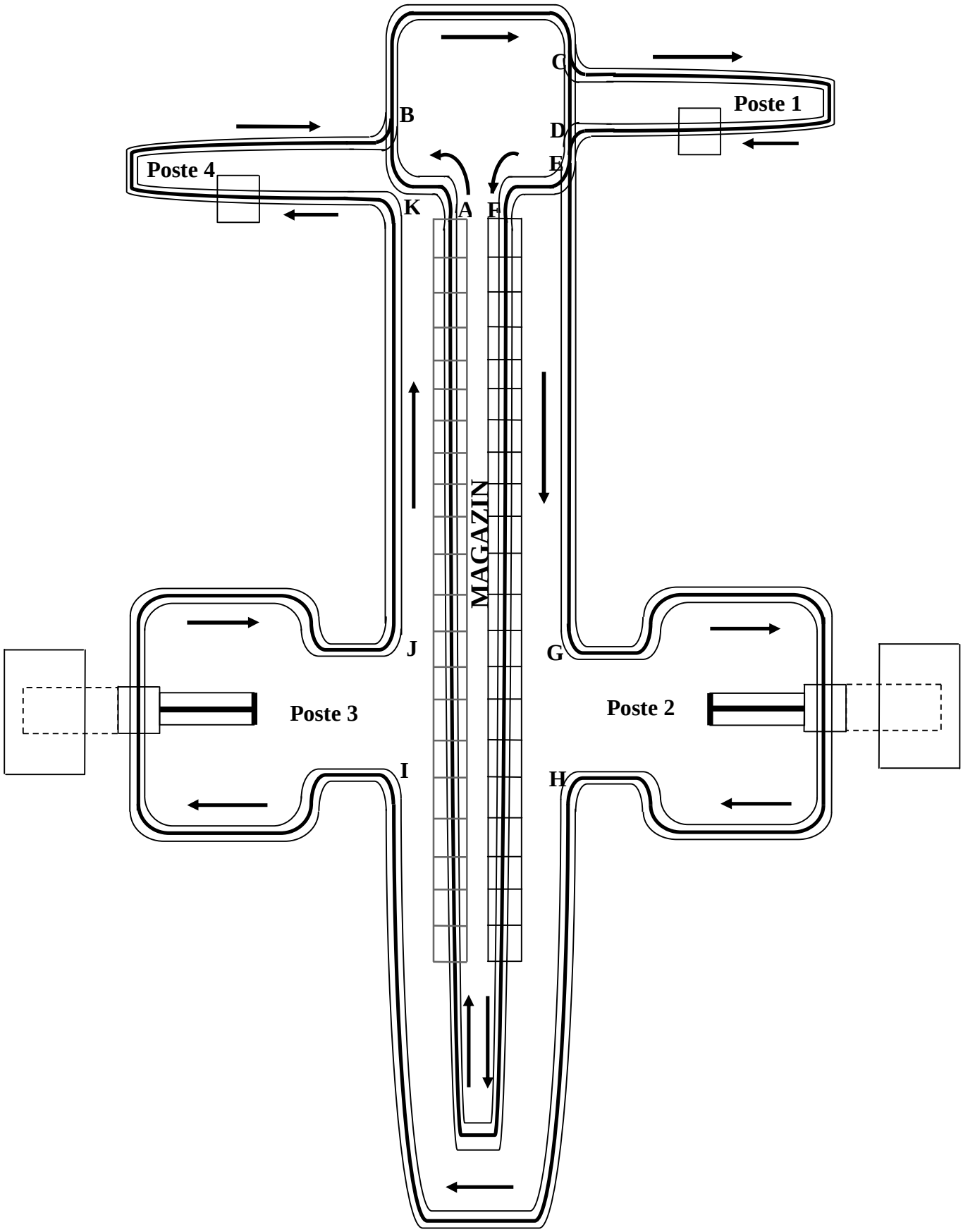


Figure 3.4 système manufacturier.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Les postes 1 et4 sont respectivement les deux postes de palettisation (mettre le produit sur la palette), et de dépalettisation (enlever le produit de la palette). Les deux postes successifs 2 et 3 dont le rôle est l'assemblage, ont une structure identique (Figure 3.5).

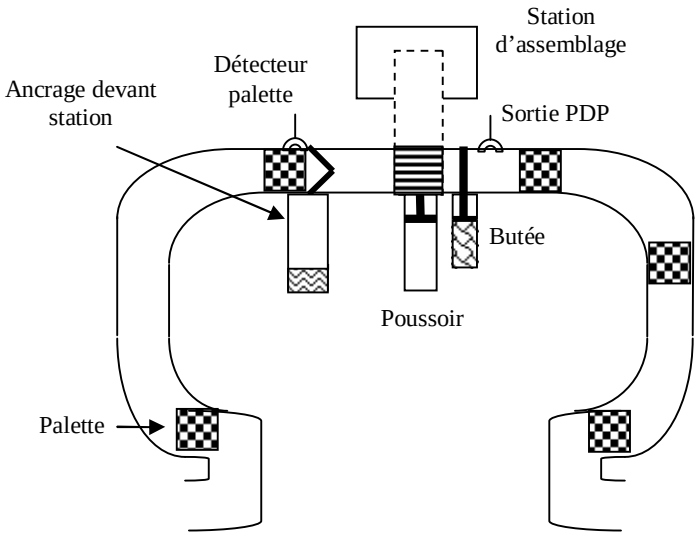


Figure 3.5 postes 2, 3.

Afin de réduire la complexité du problème de supervision, nous optons à utiliser une supervision modulaire qui permet de découper le problème en n sous-problèmes indépendants. [10]

Le superviseur résultant est la conjonction des n superviseurs S_1^ S_2 ^….. S_n (Figure 3.6)

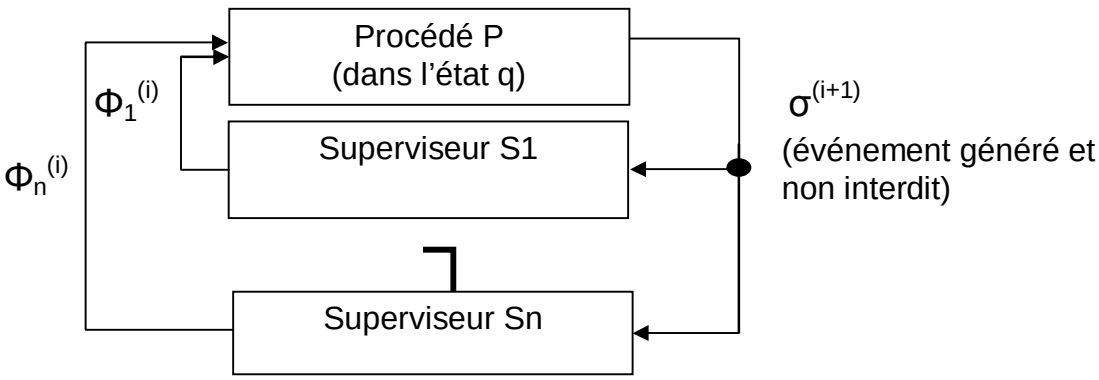


Figure 3.6 supervision modulaire

PDF created with pdfFactory Pro trial version www.pdffactory.com

Le modèle RdP du poste 2, 3 est donné par la figure 3.7.

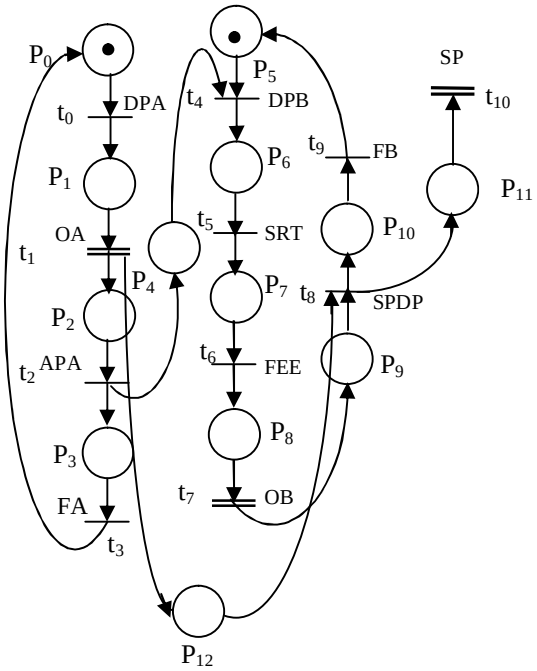


Fig 3.7 Le modèle RdP du poste 2, 3.

Les événements associés à chaque transition et les places du réseau sont explicités aux tableaux 3.1 et 3.2 respectivement :

Evénement	Action	Evénement	Action	Evénement	Action
DPA	Détection palette devant Ancrage	OA	Ouverture ancrage	APA	Absence palette devant Ancrage
FA	Fermeture ancrage	DPB	Détection palette devant Butée	SRT	début lecture étiquette
FEE	Fin écriture étiquette	OB	Ouverture Butée	SPDP	Sortie palette du PDP
FB	Fermeture Butée	SP	Sortie du poste		

Tableau 3.1. Evénements associés aux transitions

PDF created with pdfFactory Pro trial version www.pdffactory.com

Nom de place	Description	Nom de Place	Description
P_0	L'ancrage est fermé L'absence de palettes	P_7	Opération de lecture, assemblage et écriture
P_1	Présence de palettes devant l'ancrage	P_8	Fin d'assemblage
P_2	L'ancrage est ouvert	P_9	Butée est ouvert
P_3	Absence de palet lorsque l'ancrage est ouvert	P_{10}	Absence de palette après butée
P_4	Présence de palette entre l'ancrage et butée	P_{11}	Une palette après butée
P_5	Butée est fermé	P_{12}	Une palette entre ancrage et butée
P_6	Présence de palette devant butée		

Tableau 3.2. Description de chaque place

Concernant les postes 2 et 3 nous avons choisi les spécifications suivantes :

1) le nombre de palettes entre l'ancrage devant station et la butée est limité à 1.

2) le nombre de palettes à la sortie du poste de présentation (SPDP) est limité à 3.

Les spécifications sont traduites par les contraintes suivantes :

$$\begin{cases} M_1(l,k) = m(p_{12}) \le 1 \\ M_2(l,k) = m(p_{11}) \le 3 \end{cases}$$

La première spécification ne sera pas transformée car elle vérifie la condition de contrôlabilité structurelle (sa place de contrôle n'a pas d'arc sortant vers une transition non contrôlable). Cependant, la deuxième spécification doit être transformée à cause de l'arc sortant de sa place de contrôle vers la transition non contrôlable t_8 (SPDP).

L'algorithme de transformation est donné par la table de la figure 3.8.

PDF created with pdfFactory Pro trial version www.pdffactory.com

	t_0	t_2	t_3	t_4	t_5	t_6	t_8	t_9		
p_0	-1	0	1	0	0	0	0	0		
p_1	1	0	0	0	0	0	0	0		
p_2	0	-1	0	0	0	0	0	0		
p_3	0	1	-1	0	0	0	0	0		
p_4	0	1	0	-1	0	0	0	0		
p_5	0	0	0	-1	0	0	0	1		
p_6	0	0	0	1	-1	0	0	0	W_u	
p_7	0	0	0	0	1	-1	0	0		
p_8	0	0	0	0	0	1	0	0		
p_9	0	0	0	0	0	0	-1	0		
p_{10}	0	0	0	0	0	0	1	-1		
p_{11}	0	0	0	0	0	0	1	0		
p_{12}	0	0	0	0	0	0	-1	0		
	0	0	0	0	0	0	1	0	$n = lW_u$	$l = (0000000000010), k = 3$
Step 1	0	0	0	0	0	0	0	0	n	$l' = (000000000\alpha_1 01\alpha_2), k' = 3$
α_1	0	0	0	0	0	0	0	0	N	$\alpha_1 + \alpha_2 = 1$
α_2	0	0	0	0	0	0	0	0		$\alpha_i \in \aleph^2$

Fig 3.8 Etapes de l'algorithme de transformation de la contrainte $m(p_{11}) \leq 3$.

De cette transformation résultent deux possibilités de contrôle, et les contraintes résultantes sont alors :

$$\begin{cases} M_2(l^1, k^1) = \{m \in \aleph^m \mid m(p_{11}) + m(p_{12}) \leq 3\} \\ ou \\ M_2(l^2, k^2) = \{m \in \aleph^m \mid m(p_9) + m(p_{11}) \leq 3\} \end{cases}$$

Le choix de l'une de ces deux solutions est fait selon leurs optimalité : la solution dont l'élément maximal coïncide avec l'élément maximal de la contrainte contrôlable $M_1(l, k)$, est la solution optimale.

L'élément maximal de $M_1(l, k)$ est $m(p_{12}) = 1$;

L'élément maximal de $M_2(l^1, k^1)$ est $(m(p_{11}) = 3, m(p_{12}) = 0 \neq 1)$;

PDF created with pdfFactory Pro trial version www.pdffactory.com

L'élément maximal de $M_2(l^2,k^2)$ est $(m(p_9)=1, m(p_{11})=2)$, dans ce cas on peut aussi avoir $m(p_{12})=1$.

D'où $M_2(l^2,k^2)$ est la solution optimale du problème de commande.

Le réseau du système en boucle fermée est donné par la figure 3.9.

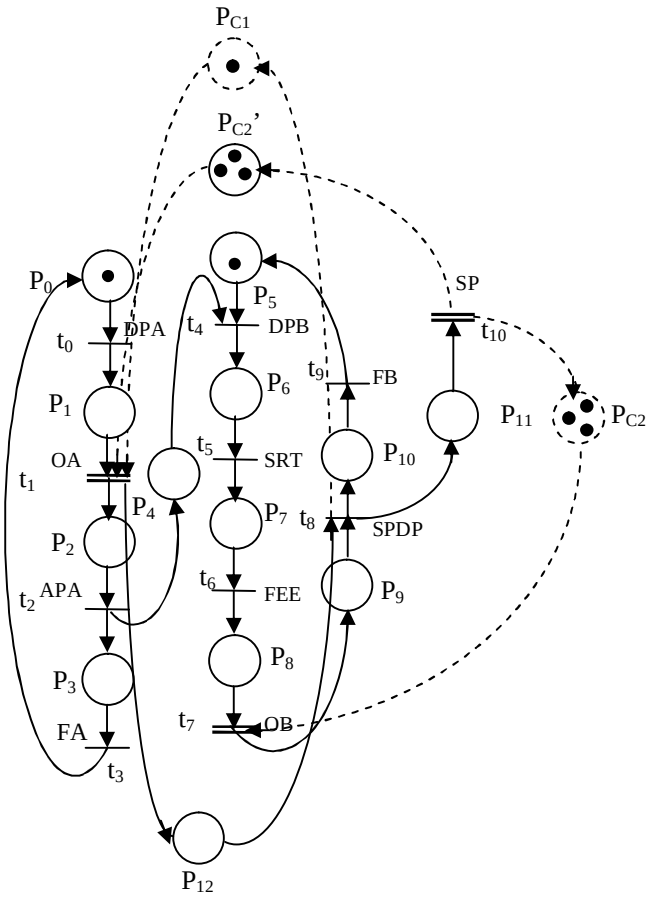


Fig 3.9 Réseau du poste 2, 3 en boucle fermée.

Le moniteur p_{C1} garantie la première spécification, les moniteurs p_{C2} et p'_{C2} sont les deux choix garantissant la deuxième spécification, où p_{C2} est le moniteur optimal.

Les tâches exécutées dans chacun des postes 2 et 3 sont modélisées par des macro places afin de réduire la taille du réseau. Chaque tâche est modélisée par un simple réseau avec une seule place (une fois marquée représente la tâche en exécution) et deux transitions (début de tâche, fin de tâche). Les transitions début de tâche sont toujours contrôlables, et les transitions fin de tâche sont non contrôlables.

Le modèle Rdp de la ligne est donné par la figure 3.10.

PDF created with pdfFactory Pro trial version www.pdffactory.com

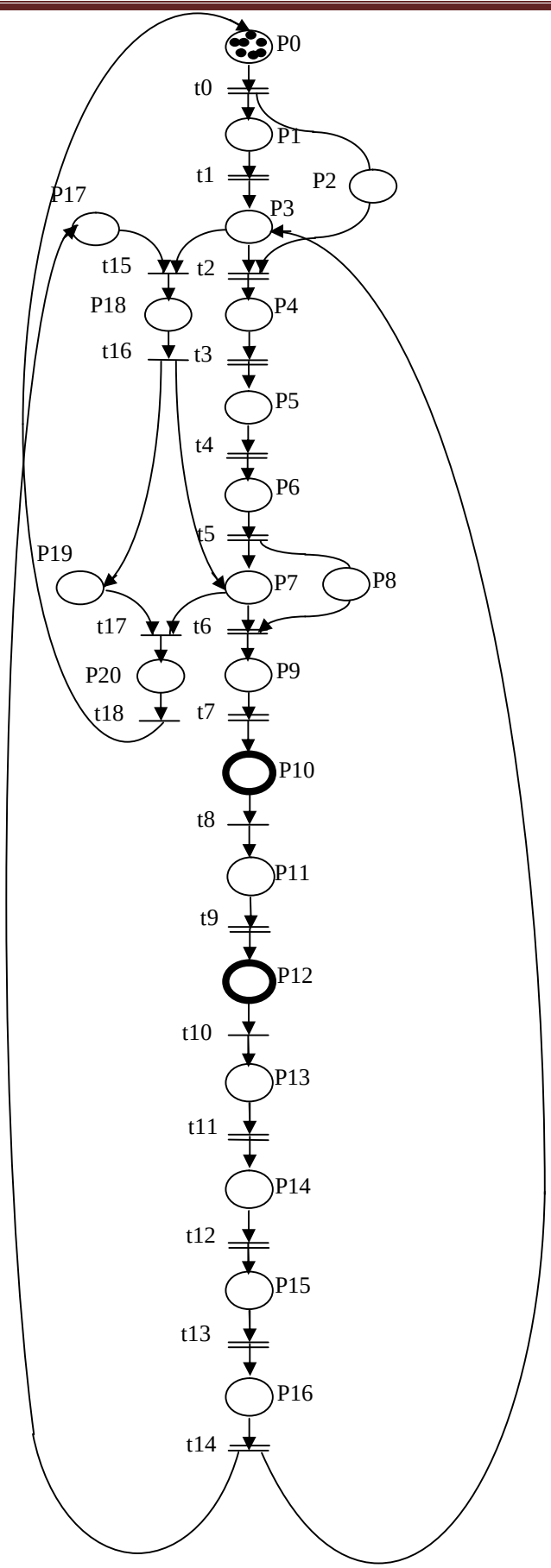


Figure 3.10 modèle Rdp de la ligne.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Les événements associés à chaque transition et les places du réseau sont explicités aux tableaux 3.3 et 3.4 respectivement :

Evénement	Action	Evénement	Action	Evénement	Action
t_0	Sortie du magasin et entrée dans AB	t_7	Sortie de EG et entrée dans poste 2.	T_{14}	Sortie du poste 4 et entrée dans BC.
T_1	Sortie de AB et entrée bans BC.	T_8	Sortie du poste 2 et entrée dans HI.	T_{15}	Sortie de BC et entrée dans CD.
T_2	Sortie de BC et entrée dans poste 1.	T_9	Sortie de HI et entée dans poste 3.	T_{16}	Sortie de CD et entrée dans DE.
T_3	Début de paléttisation.	T_{10}	Sortie du poste 3 et entrée dans JK.	T_{17}	Sortie de DE et entrée dans EF.
T_4	Fin de paléttisation.	T_{11}	Sortie de JK et entrée dans poste 4.	T_{18}	Sortie de EF et retour au magasin.
T_5	Sortie du poste 1 et entrée dans DE.	T_{12}	Début de dépalettisation.		
T_6	Sortie de DE et entrée dans EG.	T_{13}	Fin de dépalettisation.		

Tableau 3.3 Evénements associés aux transitions

Nom de place	Description	Nom de Place	Description
P_0	Palette dans le magasin.	P_{11}	Palette dans HI.
P_1	Palette dans AB.	P_{12}	Palette dans poste 3.
P_2		P_{13}	Palette dans JK.
P_3	Palette dans BC.	P_{14}	Palette dans poste 4.
P_4	Palette dans poste 1.	P_{15}	Dépalétisation.
P_5	Palettisation	P_{16}	dépalétisation finie.
P_6	Palettisation finie.	P_{17}	
P_7	Palette dans DE.	P_{18}	Palette dans CD.
P_8		P_{19}	
P_9	Palette dans EG.	P_{20}	Palette dans EF.
P_{10}	Palette dans poste 2.		

Tableau 3.4 Description de chaque place.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Les spécifications imposées au reste du système sont :

-chacun des tronçons AB, BC, CD, DE, EF contiennent une seule palette au plus.

- la capacité des postes 1 et 4 est de 1.

-la capacité des postes 2 et 3 est de 6.

Ces spécifications sont alors traduites par les contraintes suivantes :

$$M_1(l,k) = m(p_1) \leq 1.$$

$$M_2(l,k) = m(p_3) \leq 1.$$

$$M_3(l,k) = m(p_4) + m(p_5) + m(p_6) \leq 1.$$

$$M_4(l,k) = m(p_7) \leq 1.$$

$$M_5(l,k) = m(p_{10}) \leq 6.$$

$$M_6(l,k) = m(p_{12}) \leq 6.$$

$$M_7(l,k) = m(p_{14}) + m(p_{15}) + m(p_{16}) \leq 1.$$

$$M_8(l,k) = m(p_{18}) \leq 1.$$

$$M_9(l,k) = m(p_{20}) \leq 1.$$

Seules Les deux contraintes $M_8(l,k)$ et $M_9(l,k)$ nécessitent une transformation, car leurs places de control possèdent un arc sortant vers les transitions non contrôlables t_{15} et t_{17}.

L'algorithme de transformation de la contrainte $M_8(l,k)$ est donné par la figure 3.11

PDF created with pdfFactory Pro trial version www.pdffactory.com

	t_4	t_8	t_{10}	t_{13}	t_{15}	t_{16}	t_{17}		
p_0	0	0	0	0	0	0	0		
p_1	0	0	0	0	0	0	0		
p_2	0	0	0	0	0	0	0		
p_3	0	0	0	0	−1	0	0		
p_4	0	0	0	0	0	0	0		
p_5	−1	0	0	0	0	0	0		
p_6	1	0	0	0	0	0	0		
p_7	0	0	0	0	0	1	−1		
p_8	0	0	0	0	0	0	0		
p_9	0	0	0	0	0	0	0		
p_{10}	0	−1	0	0	0	0	0	C_u	
p_{11}	0	1	0	0	0	0	0		
p_{12}	0	0	−1	0	0	0	0		
p_{13}	0	0	1	0	0	0	0		
p_{14}	0	0	0	0	0	0	0		
p_{15}	0	0	0	−1	0	0	0		
p_{16}	0	0	0	1	0	0	0		
p_{17}	0	0	0	0	−1	0	0		
p_{18}	0	0	0	0	1	−1	0		
p_{19}	0	0	0	0	0	1	−1		
p_{20}	0	0	0	0	0	0	1		
	0	0	0	0	1	−1	0	$n = lC_u$	$l = (00000000000000000100), k = 1$
Step 1	0	0	0	0	0	−1	0	n	$l' = (000\alpha_1 0000000000000\alpha_2 100), k' = 1$
α_1	0	0	0	0	0	−1	0	N	$\alpha_1 + \alpha_2 = 1$
α_2	0	0	0	0	0	−1	0		$\alpha_i \in \aleph^2$

Figure 3.11 Etapes de l'algorithme de transformation de la contrainte $M_8(l,k)$.

De cette transformation résultent deux possibilités de contrôle, et les contraintes résultantes sont alors :

PDF created with pdfFactory Pro trial version www.pdffactory.com

$$\begin{cases} M_8(l^1,k^1)=\{m\in \aleph^m \mid m(p_3)+m(p_{18})\le 3\} \\ ou \\ M_8(l^2,k^2)=\{m\in \aleph^m \mid m(p_{17})+m(p_{18})\le 3\} \end{cases}$$

Le choix entre ces deux possibilités est indifférent.

L'algorithme de transformation de la contrainte $M_9(l,k)$ est donné par la figure 3.12.

	t_4	t_8	t_{10}	t_{13}	t_{15}	t_{16}	t_{17}		
p_0	0	0	0	0	0	0	0		
p_1	0	0	0	0	0	0	0		
p_2	0	0	0	0	0	0	0		
p_3	0	0	0	0	−1	0	0		
p_4	0	0	0	0	0	0	0		
p_5	−1	0	0	0	0	0	0		
p_6	1	0	0	0	0	0	0		
p_7	0	0	0	0	0	1	−1		
p_8	0	0	0	0	0	0	0		
p_9	0	0	0	0	0	0	0		
p_{10}	0	−1	0	0	0	0	0	C_u	
p_{11}	0	1	0	0	0	0	0		
p_{12}	0	0	−1	0	0	0	0		
p_{13}	0	0	1	0	0	0	0		
p_{14}	0	0	0	0	0	0	0		
p_{15}	0	0	0	−1	0	0	0		
p_{16}	0	0	0	1	0	0	0		
p_{17}	0	0	0	0	−1	0	0		
p_{18}	0	0	0	0	1	−1	0		
p_{19}	0	0	0	0	0	1	−1		
p_{20}	0	0	0	0	0	0	1		
	0	0	0	0	0	0	1	$n=lC_u$	$l=(00000000000000000001), k=1$
Step 1	0	0	0	0	0	0	0	n	$l'=(0000000a_1000000000a_21), k'=1$
α_1	0	0	0	0	0	1	0	N	$\alpha_1+\alpha_2=1$
α_2	0	0	0	0	0	1	0		$\alpha_i\in \aleph^2$
Step 2	0	0	0	0	0	0	0	n	$l'=(0000000a_10000000000\ a_3a_21), k'=1$

PDF created with pdfFactory Pro trial version www.pdffactory.com

α_1 0 0 0 0 0 0 0 α_2 0 0 0 0 0 1 0 α_3 0 0 0 0 1 0 0	N	$\alpha_1+\alpha_2=1$ $\alpha_1+\alpha_2=\alpha_3$ $\alpha_i \in \aleph^3$
Step 3 0 0 0 0 0 0 0	n	$l'=(0000000a_1000000000(a_3+a_4)a_21), k'=1$
α_1 0 0 0 0 0 0 0 α_2 0 0 0 0 0 0 0 α_3 0 0 0 0 1 0 0 α_4 0 0 0 0 1 0 0	N	$\alpha_1+\alpha_2=1$ $\alpha_1+\alpha_2=\alpha_3$ $\alpha_1+\alpha_2=\alpha_3+\alpha_4$ $\alpha_i \in \aleph^4$
Step 4 0 0 0 0 0 0 0	n	$l'=(000a_5000a_1000000000a_6(a_3+a_4)a_21), k'=1$
α_1 0 0 0 0 0 0 0 α_2 0 0 0 0 0 0 0 α_3 0 0 0 0 0 0 0 α_4 0 0 0 0 1 0 0 α_5 0 0 0 0 0 0 0 α_6 0 0 0 0 0 0 0	N	$\alpha_1+\alpha_2=1$ $\alpha_1+\alpha_2=\alpha_3$ $\alpha_1+\alpha_2=\alpha_3+\alpha_4$ $\alpha_3+\alpha_4=\alpha_5+\alpha_6$ $\alpha_i \in \aleph^6$
Step 5 0 0 0 0 0 0 0	n	$l'=(000(a_5+a_7)000a_1000000000(a_6+a_8)(a_3+a_4)a_21), k'$
α_1 0 0 0 0 0 0 0 α_2 0 0 0 0 0 0 0 α_3 0 0 0 0 0 0 0 α_4 0 0 0 0 0 0 0 α_5 0 0 0 0 0 0 0 α_6 0 0 0 0 0 0 0 α_7 0 0 0 0 0 0 0 α_8 0 0 0 0 0 0 0	N	$a_1+a_2=1$ $a_1+a_2=a_3$ $a_1+a_2=a_3+a_4$ $a_3+a_4=a_5+a_6$ $a_3+a_4=a_5+a_7+a_6+a_8$ $a_i \in \aleph^8$

Figure 3.11 Etapes de l'algorithme de transformation de la contrainte $M_9(l,k)$.

Nous avons 8 possibilités de contrôles dont 3 sont redondantes, puisque $\alpha_4 = \alpha_7 = \alpha_8 = 0$

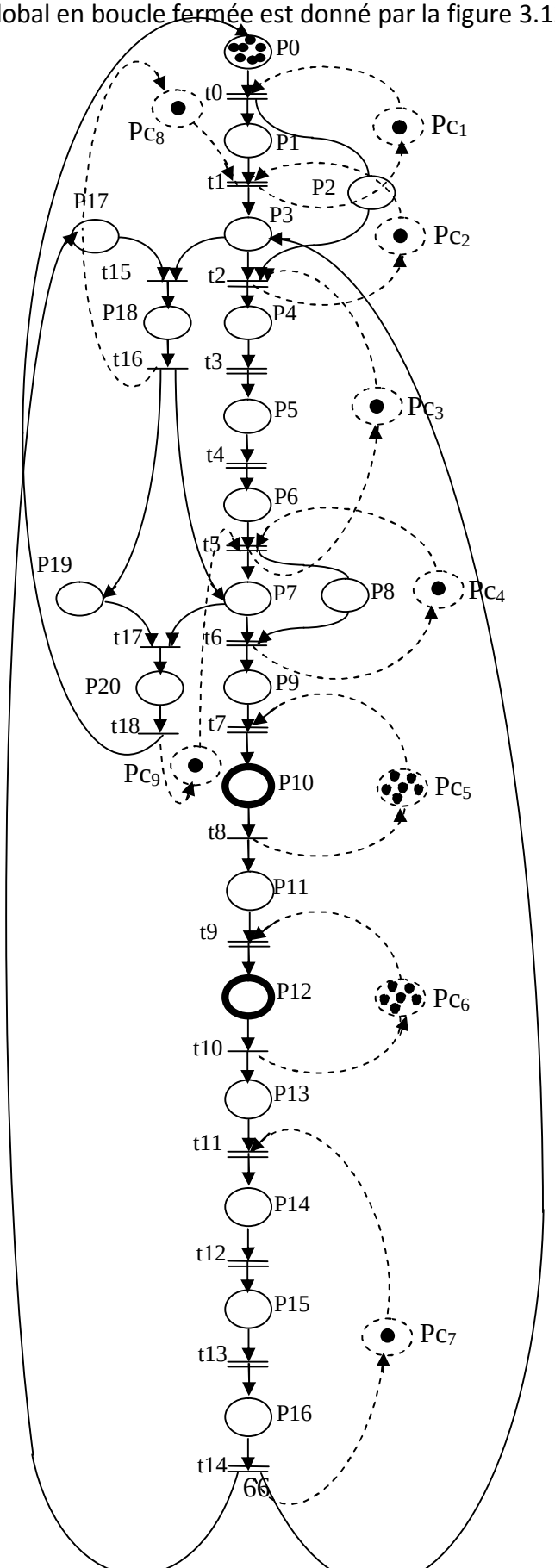
Les contraintes résultantes sont :

$$\begin{cases} m(p_{20}) + m(p_{19}) + m(p_{18}) + m(p_{17}) \le 1 \\ m(p_{20}) + m(p_{19}) + m(p_{18}) + m(p_3) \le 1 \\ m(p_{20}) + m(p_7) + m(p_{18}) + m(p_3) \le 1 \\ m(p_{20}) + m(p_7) + m(p_{18}) + m(p_{17}) \le 1 \\ m(p_{20}) + m(p_7) \le 1 \end{cases}$$

PDF created with pdfFactory Pro trial version www.pdffactory.com

Le choix du superviseur optimal se porte sur $m(p_{20}) + m(p_7) \leq 1$, car c'est le plus permissif possible entre les 5 choix.

Le réseau du système global en boucle fermée est donné par la figure 3.12



PDF created with pdfFactory Pro trial version www.pdffactory.com

Figure 3.12 réseau global supervisé

5 Conclusion :

L'existence d'événements non contrôlables, entraîne des états interdits. Nous avons vu que pour éviter ces états, il faut restreindre l'ensemble d'atteignabilité du système en boucle fermée en un sous ensemble qui soit le maximum permissif, et la solution optimale est l'élément suprême de cet ensemble.

Nous avons vu aussi une paramètrisation linéaire du choix existant entre plusieurs moniteurs résultants d'une transformation d'une contrainte non contrôlable, ainsi que l'algorithme de Basile et al, donnant ces différents choix.

Le choix du moniteur optimal se fait alors selon sa permissivité par rapport aux autres moniteurs et selon d'autres conditions telles que la vivacité par exemple.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 4

Supervision par les polyèdres

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chapitre 4

Supervision par invariance.

4.1 Introduction :

Dans cette section, une nouvelle méthode est présentée pour la supervision des systèmes à événements discrets par Rdps. Contrairement à l'approche structurelle, basée sur les invariants de marquage, cette méthode est basée sur la propriété d'invariance des ensembles. Nous donnons des conditions nécessaires et suffisantes sur le franchissement des transitions afin de garantir le respect des spécifications imposées au système.

4.2 Absence de transitions non contrôlables

4.2.1 Position du problème :

Etant donné un système à événements discrets modélisé par réseaux de Petri, à commander avec des contraintes sur l'évolution du marquage de type Polyèdrale $P(Q,\mu)=\{m\in N^n \ / \ Q.m\le\mu\}$ où : n est le nombre de place du réseau de Petri. Supposons que toutes les transitions du modèle RdP sont contrôlables. La commande de ce système consiste à interdire au marquage de sortir du Polyèdre des contraintes. Autrement dit, à partir d'un marquage $m_K\in P(Q,\mu)$, le marquage m_{K+1} résultant du franchissement d'une transition validée est toujours dans $P(Q,\mu)$.

4.2.2 Synthèse de superviseur basée sur la propriété d'invariance positive

Hypothèse :

On suppose dans ce qui suit que seulement une transition est franchie à la fois (non concurence).

PDF created with pdfFactory Pro trial version www.pdffactory.com

On considère la classe graphe d'états arrière des réseaux de Petri.

Etant donné un ensemble Γ, on dénote $\mathrm{Int}(\Gamma)$ son intérieur, et $\partial(\Gamma)$ son contour

4.2.1 Définitions [18]

Définition 4.1

Un ensemble C est dit convexe lorsque, pour tout *x* et *y* de C, le segment [*x*, *y*] est tout entier contenu dans C, i.e. $\forall\, x, y \in C, \forall\, t \in [0,1], t\,x + (1-t)\,y \in C$

Définition 4.2

L'ensemble convexe $P \subseteq R^n$, formé de l'intersection d'un nombre fini q de demis espaces fermés, défini par : $P = \{x \in R^n \;/\; G.x \le g\}$ est un polyèdre, où $G \in R^{q\,x\,n}, g \in R^q, q, n \in N$.

Définition 4.3

L'ensemble $P(G,g)$ est dit positivement invariant pour un système donné si et seulement si sa trajectoire d'état reste dans $P(G,g)$.

Soit un système à événement discrets modélisé par un Rdp dont l'évolution est décrite par l'équation $m_{K+1} = m_K + W\sigma_K$, et soumis à des contraintes de marquage polyhèdrales données par : $P(Q,\mu) = \{m \in N^n \;/\; Q.m \le \mu\}$ où $Q \in N^{n\,x\,nc}$ et $\mu \in N^{nc}$.

Notre objectif est de contrôler le système de sorte que le polyèdre des contraintes de marquage soit respecté. La loi de commande qui assure cet objectif est le superviseur défini par :

Définition 4.4 [15]

$$\begin{aligned} f :\ R(m_0) &\to \quad T \\ m_K \quad &\to f(m_K) = \Sigma_K \end{aligned} \qquad 4.1$$

Où $R(m_0)$ est l'ensemble de tous les marquages atteignables à partir de m_0, T est l'ensemble des transitions du réseau, m_K est un marquage donné et Σ_K est l'ensemble des transitions autorisées à partir de m_K.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Les propositions suivantes nous donnent des conditions nécessaires et suffisantes pour l'invariance positive de l'ensemble des contraintes polyèdrales :

4.2.2 Résultats [15]

Proposition 4.1

L'ensemble Σ_K est égal à l'ensemble des transitions T pour tout marquage m_K appartenant à $\mathrm{Int}(P(Q,\mu))$.

En d'autres termes, une transition T_j d'un réseau de Petri donné, est toujours autorisée à partir d'un marquage m_K appartenant à $\mathrm{Int}(P(Q,\mu))$.

Preuve :

C'est évident que si m_K appartient à $\mathrm{Int}(P(Q,\mu))$ alors m_{K+1} appartient à $\mathrm{Int}(P(Q,\mu))$ ou à l'une des faces du polyèdre, en tenant compte qu'une seule transition est franchie à la fois (non concurrence).

On a vu dans la Proposition 1, la condition de franchissement lorsque $m_K \in \mathrm{Int}(P(Q,\mu))$, la proposition suivante nous donnera la condition de franchissement lorsque $m_K \in \partial(P(Q,\mu))$:

Proposition 4.2

Le franchissement d'une transition T_j d'un réseau de Petri donné est autorisé à partir d'un marquage m_K appartenant à la $i^{ième}$ facette du polyèdre $P(Q,\mu)$ si et seulement si :

$$Q^i.W.\sigma_k \le 0 \qquad 4.2$$

Et interdit sinon, où σ_K est le vecteur caractéristique du franchissement de la transition T_j.

Preuve :

La $i^{ième}$ facette $P(Q^i,\mu^i)$ du polyèdre $P(Q,\mu)$ est définie par :

$$\begin{cases} Q^i.m_K = \mu^i \\ \qquad\qquad i \ne l. \\ Q^l.m_K \le \mu^l \end{cases} \qquad 4.3$$

PDF created with pdfFactory Pro trial version www.pdffactory.com

L'ensemble $P(Q,\mu)$ est positivement invariant si et seulement si :

$$\forall m_K \in P(Q^i,\mu^i) \Rightarrow m_{K+1} \in P(Q,\mu)$$
$$\Rightarrow Q.m_{K+1} \leq \mu$$

Condition nécessaire :

Montrons que si $Q^i.m_{K+1} \leq \mu^i$ implique que $Q^i.W.\sigma_k \leq 0.$

En remplaçant m_{k+1} par son expression d'évolution, nous aurons :

$$Q.(m_K + W.\sigma_K) \leq \mu.$$

$$\begin{cases} Q^i.m_K + Q^i.W.\sigma_K \leq \mu^i \\ \qquad\qquad\qquad\qquad\quad i \neq l. \\ Q^l.m_K + Q^l.W.\sigma_K \leq \mu^l \end{cases}$$

Puisque $Q^i.m_k = \mu^i$, alors la condition de franchissement est :

$$Q^i.W.\sigma_k \leq 0.$$

Condition suffisante :

Montrons que si $Q^i.W.\sigma_k \leq 0$ implique que $Q^i.m_{K+1} \leq \mu^i$.

$$\begin{aligned} Q^i.W.\sigma_K \leq 0 \quad &\Rightarrow Q^i.W.\sigma_K + \mu_i \leq \mu_i \text{ (puisque } \mu_i \geq 0) \\ &\Rightarrow Q^i.W.\sigma_K + Q^i m_K \leq \mu_i \text{ (puisque } Q^i m_K = \mu_i) \\ &\Rightarrow Q^i.(W.\sigma_K + m_K) \leq \mu_i \\ &\Rightarrow Q^i.m_{K+1} \leq \mu_i \end{aligned}$$

$Q^i.W.\sigma_k \leq 0$ est une condition nécessaire et suffisante pour le franchissement de la transition T_j.

Exemple 4.1:

Soit le système de transport de la figure 4.1 composé d'un entrepôt E et deux postes de travail W_1 et W_2. Des véhicules autoguidés circulent entre ces trois postes. L'objectif assigné au superviseur est d'éviter les collisions des véhicules.

PDF created with pdfFactory Pro trial version www.pdffactory.com

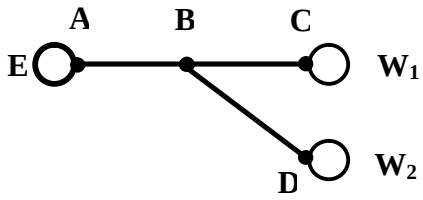


Figure 4.1 système de transport.

Le modèle réseau de Petri du système est donné par la figure 4.2.

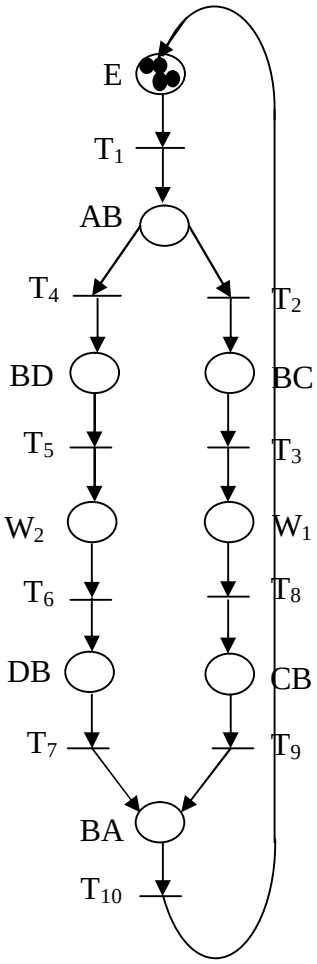


Figure 4.2 Modèle réseau de Petri du système de transport ;

PDF created with pdfFactory Pro trial version www.pdffactory.com

La matrice d'incidence du réseau est :

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

Les spécifications garantissant cet objectif sont les suivantes :

$$\begin{cases} m_{AB} + m_{BA} \leq 1 \\ m_{BC} + m_{CB} \leq 1 \\ m_{BD} + m_{DB} \leq 1 \\ m_{AB} + m_{CB} + m_{DB} \leq 2 \end{cases}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Intuitivement, et à partir du réseau, pour satisfaire les contraintes, il est nécessaire d'interdire les transitions d'entrée des places contraintes :

Facette 1 : interdire T_1, T_7, T_9.

Facette 2 : interdire T_2, T_8.

Facette 3 : interdire T_4, T_6.

Facette 4 : interdire T_1, T_6, T_8.

Testons ces résultats intuitifs en utilisant la proposition 2 :

Soit σ_k^j le vecteur de franchissement élémentaire de la transition T_j, càd, seulement la $j^{ième}$ composante de σ_k est égale à 1.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Facette 1: $m_{AB} + m_{BA} = 1$.

Pour j=1: $Q^1.W.\sigma_K^1 = 1 > 0 \Rightarrow T_1$ est interdite.

Pour j=2: $Q^1.W.\sigma_K^2 = -1 \leq 0 \Rightarrow T_2$ autorisée.

M

Comme σ_k est un vecteur de franchissement élémentaire, on peut conclure sur les transitions interdites à partir du vecteur $Q^i.W$:

$$Q^1W = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} \\ 1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 & 1 & -1 \end{pmatrix}.$$

Les composantes positives indiquent les transitions interdites qui sont T_1, T_7,T_9.

Facette 2: $m_{BC} + m_{CB} = 1$.

$$Q^2W = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{pmatrix}.$$

Les transitions interdites sont T_2, T_8.

Facette 3: $m_{BD} + m_{DB} = 1$.

$$Q^3W = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix}$$

Les transitions interdites sont T_4 , T_6.

Facette 4: $m_{AB} + m_{CB} + m_{DB} = 2$.

$$Q^4W = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} \\ 1 & -1 & 0 & -1 & 0 & 1 & -1 & 1 & -1 & 0 \end{pmatrix}$$

Les transitions interdites sont T_1, T_6, T_8.

PDF created with pdfFactory Pro trial version www.pdffactory.com

2.3 La synthèse du superviseur :

On peut synthétiser un superviseur en associant à chaque facette une place de contrôle :

- § pour la facette i, calculons le vecteur $Q^i.W$.
- § Les composantes positives de $Q^i.W$ correspondent aux transitions interdites, qui sont les transitions de sortie de la place de contrôle.
- § Les composantes nulles de $Q^i.W$ correspondent aux transitions qui n'ont aucun lien avec la place de contrôle.
- § Les composantes négatives de $Q^i.W$ correspondent aux transitions d'entrée à la place de contrôle [4].
- § Le marquage initial de la place de contrôle est $m_{C0}^i = \mu^i - Q^i m_{K0}$ [4].

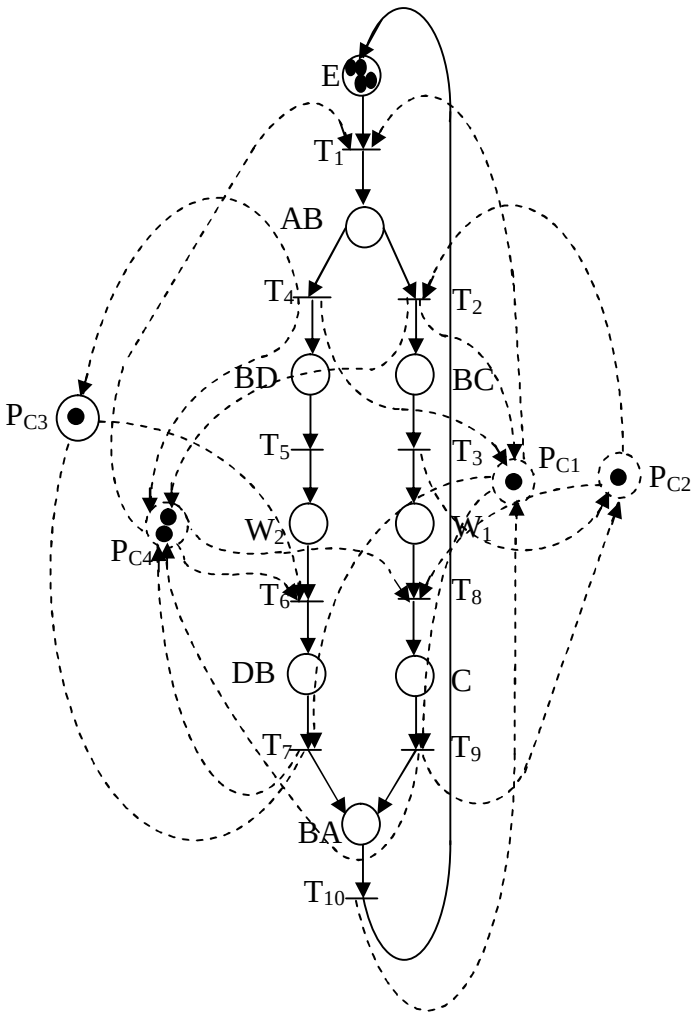


Figure 4.3 Réseau du système supervisé

PDF created with pdfFactory Pro trial version www.pdffactory.com

Chaque place de control p_{Ci}, i=1,4 correspond à une facette i du polyèdre des contraintes.

3 Présence de transitions non contrôlables :

On sait qu'on peut toujours interdire le franchissement d'une transition contrôlable, quand celle-ci mène à l'extérieure du polyèdre des contraintes.

Examinons le cas du franchissement d'une transition non contrôlable :

Proposition 4.3

D'après la définition 3.2 et en tenant toujours compte de l'hypothèse de non concurrence, nous aurons :

L'ensemble $P(Q,\mu)$ est contrôlable si à partir d'un marquage $m_K \in P(Q,\mu)$, et après le franchissement d'une transition non contrôlable T_{nc} le marquage résultant $m_{K+1} \in P(Q,\mu)$.

- $m_K \in Int(P(Q,\mu))$: même résultat et même preuve que dans proposition 4.1.
- $m_K \in \partial(P(Q,\mu))$: appliquons la propriété d'invariance positive :

$$\forall m_K \in P(Q^i,\mu^i) \Rightarrow m_{K+1} \in P(Q,\mu)$$
$$\Rightarrow Q.m_{K+1} \le \mu$$
$$\Rightarrow Q.\left(m_K + (W_{nc}\ W_c).(\sigma_{knc}\ \sigma_{kc})^T\right) \le \mu$$

Or, pas de transition contrôlable franchie, d'où $\sigma_{kc} = 0$.

$$\Rightarrow Q.(m_k + W_{nc}.\sigma_{knc}) \le \mu$$
$$\Rightarrow \begin{cases} Q^i.m_K + Q^i.W_{nc}.\sigma_{knc} \le \mu^i \\ \qquad\qquad\qquad\qquad i \ne l. \\ Q^l.m_K + Q^l.W_{nc}.\sigma_{knc} \le \mu^l \end{cases}$$

Comme $Q^i.m_k = \mu^i$, alors la condition suffisante de franchissement est :

$$Q^i.W_{nc}.\sigma_{knc} \le 0 \qquad 4.4$$

PDF created with pdfFactory Pro trial version www.pdffactory.com

4 Conclusion

Dans ce chapitre, nous avons d'abord défini quelques notions sur les ensembles que nous avons estimés nécessaires pour notre travail, telles que la convexité d'un ensemble, et l'invariance positive d'un ensemble.

Puis nous avons élaboré deux conditions garantissant que l'évolution du marquage du modèle Rdp, ne sorte pas du contour d'un ensemble polyèdral représentant l'ensemble des contraintes imposées au système à superviser.

Cette méthode a l'avantage d'identifier les transitions qui mènent à des états non désirables pour les interdire, sans passer par l'étape du graphe des marquages accessibles qui peut être de grande dimension.

Nous l'avons appliqué sur un exemple et nous avons conclu que ses résultats conviennent bien aux résultats intuitifs.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Conclusion Générale

PDF created with pdfFactory Pro trial version www.pdffactory.com

Conclusion générale :

La synthèse de contrôleurs pour les systèmes à événements discrets a été abondamment étudiée dans la littérature à partir des outils automates et Réseaux de Petri. La théorie de la synthèse de contrôleurs a été introduite par Ramadge et Wonham en utilisant les automates. L'inconvénient principal de cette approche est l'explosion du nombre d'états pour les systèmes complexes. Pour pallier à ce problème, il existe différentes approches qui utilisent les modèles RdP.

La commande des SED rencontre un problème majeur quand celui-ci est en présence de spécifications non contrôlables. Lorsqu'un événement non contrôlable non désiré se produit, le contrôleur reste incapable de l'interdire. De là, vient l'idée de chercher un sous ensemble contrôlable mais ceci restreint le comportement du système en boucle fermée, et on perd beaucoup du point de vu permissivité. On cherche alors, la solution optimale qui soit la moins restrictive (ou la plus permissive).

Nous avons étudié, dans ce mémoire, les travaux de Basile et al. qui ont élaboré un algorithme simple et efficace pour transformer des contraintes linéaires non contrôlables. Cet algorithme permet de paramétrer toutes les solutions possibles par une équation linéaire, mais symbolique. Chaque solution de cette équation est un superviseur « contrôlable ». Une application de cet algorithme à un système manufacturier dont la spécification présente une composante non contrôlable a permis d'extraire deux solutions (superviseurs) possibles. Notre choix s'est porté sur le superviseur maximalement permissif.

Par la suite, nous avons proposé deux conditions nécessaires et suffisantes qui permettent de nous donner l'ensemble des transitions à interdire à la suite d'un marquage donné. En particulier, il a été établi d'une part, que cet ensemble est vide lorsque le marquage est à l'intérieur de l'ensemble des contraintes. D'autre part, si le

PDF created with pdfFactory Pro trial version www.pdffactory.com

marquage est sur les bords de cet ensemble, on interdit le franchissement des transitions qui nous font sortir de cet ensemble. Cette méthode a l'avantage de prédire les transitions qui mènent à des états non désirables pour les interdire, sans passer par l'étape du graphe de marquages accessibles qui peut être de grande dimension.

Comme perspective à ce travail, on pense généraliser les résultats établis dans ce mémoire à des contraintes non contrôlables. Il s'agit au fait, de chercher le plus grand ensemble commandable « viable » inclus dans cette ensemble de contrainte.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Bibliographie

PDF created with pdfFactory Pro trial version www.pdffactory.com

Bibliographie :

[1] A.Giua, K.Seatzu : A systems theory view of Petri nets, Did.Ingegneria Elettrica ed Elettronica, Università di Cagliari, Italy.

[2] Roberto BACOS JUNIOR : Supervision de systèmes de production par les réseaux de Petri Application au pilotage d'AGV. Ecole doctorale STIM, Laboratoire IRCCyN,Equipe ACSED.

[3] L.E Holloway, B.H. Krogh, A. Giua : A survey of Petri Net methods for controlled discrete event systems, Kluwer Academic publishers, Boston. Manufactured in the Netherlands (1997).

[4] K. Yamalidou, J. moody, M. Lemmon, P. Antsaklis : feedback control of Petri Nets based on place invariants. Technical Repport ISIS-94-002, University of notre Dame (1994).

[5] B. Kattan : Synthèse d'un contrôleur basée sur le grafcet, Thèse de Doctorat de l'université de Joseph Fourier- Grenoble1 (2004).

[6] A. Gaffari, N. Rezg, X. Xie: Design of live and maximally permissive Petri Net controller using the théory of Regions. IEEE Transactions on Robotics and Autamation, vol 19, NO.1 (2003).

[7] Y. Arnaud : Pilotage sûr et optimal de véhicules Autoguidés. Ecole doctorale STIM, lab IRCCyN, Equipe ACSED.

[8] M.Iordache, J. Moody, Member, IEEE, and P. Antsaklis, Fellom, IEEE : Synthesis of deadlock prevention supervisors using Petri Nets. IEEE Transactions on Robotics and Autamation, vol 18, NO.1 (2002).

[9] J.Moody, Member, IEEE, and P. Antsaklis, Fellow, IEEE : Petri Net supervisors for DES with uncontrollables and unobservable transitions. IEEE Transactions on Robotics and Autamation, vol 45, NO.3 (2000).

PDF created with pdfFactory Pro trial version www.pdffactory.com

[10] M. Nourelfath, E. Niel : Modular supervisory control of an experimental automated manufacturing system.Control Engineering Practice (2003).

[11] J. L. Ferrier : Commande par supervision: outils et synthèse. Lab LISA, Université d'Angers.

[12] Jean Marie Moanda et Ir. Junior Bakola Mongo : Synthèse des controleurs optimaux pour les systèmes à événements discrets. Université de Kinshasa.

[13] A. Dideban : Synthèse de controleurs discrets par simplification de contraintes et de conditions. Thèse de doctorat l'université de Joseph Fourier- Grenoble1 (2007).

[14] F. Basile, P. Chiacchio, A. Giua : Suboptimal supervisory control of Petri Nets in presence of uncontrollable transitions via monitor places. Automatica 42 (2006) 995-1004.

[15] H. Harrouche, R.Kara, J.J.Loiseau : supervisory control based on positive invariance property of class of Petri net, université UMMTO Tizi-ouzou, Algerie,

[16] Guy Cohen : Théorie algebrique des systèmes à événements discrets, Centre Automatique et systèmes. Ecole des mines de Paris, Fontainebleu et INRIA, Rocquencourt 1995.

[17] Murata : Petri Nets : properties, analysis and aplications. Proc. of the IEEE, vol 77, N.4, pp 541-580, 1989.

[18] Blanchini : Set invariance in control, Automatica, 35(11), 1747-1767. 1999.

[19] P.J.Ramadge and W.M.Wonham : supervisory control of class of discret event processes. SIAM Tournal of control and optimisation, vol 25(1), pp 206-230, 1987.

[20] C.Amer-Yahia : Sur les méthodes d'analyse structurelle des réseaux de Petri, thèse de Doctorat, l'université de Mouloud Maameri de Tizi-ouzou, (1999).

PDF created with pdfFactory Pro trial version www.pdffactory.com

[21] A.Godon : Contribution à la commande des systèmes à événements discrets par réseaux de Petri, thèse de Doctorat, l'université d'Angers, (1996)

[22] R. Kumar: Supervisory Synthésis Techniques for Discrete Event Dynamical systems, thesis for the degree of doctor of philosophy, université du Texas.

PDF created with pdfFactory Pro trial version www.pdffactory.com