



HAL
open science

Approches canoniques pour la synthèse des contrôleurs réseaux de Petri

Sadok Rezig

► **To cite this version:**

Sadok Rezig. Approches canoniques pour la synthèse des contrôleurs réseaux de Petri. Automatique / Robotique. Université de Lorraine, 2016. Français. NNT : 2016LORR0117 . tel-01446779

HAL Id: tel-01446779

<https://theses.hal.science/tel-01446779v1>

Submitted on 26 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

*Approches Canoniques pour la Synthèse des Contrôleurs
Réseaux de Petri*

THÈSE

présentée et soutenue publiquement le 22 Septembre 2016

pour l'obtention du

DOCTORAT de l'Université de Lorraine

(Mention Automatique, Traitement du signal et des images, Génie Informatique)

Par

Sadok REZIG

Composition du jury :

Rapporteurs :	Eric NIEL	Professeur à l'Institut National des Sciences Appliquées de Lyon
	Mustapha NOURELFATH	Professeur à l'Université Laval, Canada
Examineurs :	Alessandro GIUA	Professeur à l'Université Aix-Marseille
	Pascal BERRUET	Professeur à l'Université de Bretagne Sud
Directeur de thèse :	Nidhal REZG	Professeur à l'Université de Lorraine
Co-directeurs de thèse :	Zied ACHOUR	Maitre de conférences à l'Université de Lorraine

A la mémoire de mon père Taieb

A ma mère

A mes sœurs

Remerciements

Mes travaux de thèse ont été réalisés au sein du laboratoire LGIPM à l'Ecole Nationale d'Ingénieurs de Metz (ENIM) dont je tiens à remercier le directeur M. Yves GRANJON, Professeur à l'Université de Lorraine.

Je tiens à exprimer mes remerciements les plus sincères à M. Nidhal REZG, Professeur à l'Université de Lorraine, qui, en plus de sa confiance et de son aide scientifique qu'il m'a accordée, m'a suivi et guidé tout au long de ce travail. Je tiens à lui exprimer ma gratitude et mes remerciements les plus distingués pour le temps qu'il a consacré à diriger cette thèse.

J'exprime aussi toute ma reconnaissance à mon co-directeur de thèse M. Zied ACHOUR, Maître de conférences à l'Université de Lorraine, pour la confiance qu'il m'a témoignée, son soutien et ses précieux conseils tout le long de cette thèse.

De plus, j'exprime ma reconnaissance amicale à Mohamed Ali KAMMOUN, Ingénieur de recherche au LGIPM, qui par son aide et sa disponibilité, a contribué à la réalisation de ce travail.

Ma gratitude et mes remerciements vont également aux membres du jury :

A M. Alessandro GIUA, Professeur à l'Université Aix-Marseille, d'avoir bien voulu examiner ce travail et pour l'intérêt qu'il a porté au sujet.

A M. Eric NIEL, Professeur à l'Institut National des Sciences Appliquées de Lyon et M. Mustapha NOURELFATH, Professeur à l'Université Laval au Canada, pour avoir accepté d'être rapporteurs de ce travail et pour leur commentaires constructifs et leur remarques pertinentes.

A M. Pascal BERRUET, Professeur à l'Université de Bretagne Sud, pour avoir accepté d'examiner ce travail, pour l'intérêt qu'il a porté au sujet.

Je remercie vivement tous les membres du laboratoire LGIPM pour les encouragements et le soutien reçu durant ma thèse.

Mes remerciements et ma gratitude vont à mes trois sœurs, Saoussen, Leila et Ines, qui ont toujours eu confiance en moi, pour m'avoir encouragé tout le long de cette thèse.

Un remerciement très particulier à Zaineb, qui m'a donné le courage pour surmonter les moments difficiles durant cette thèse, pour son amour, son soutien et sa confiance.

Enfin, un remerciement très particulier à ma chère mère Hayet qui m'a toujours soutenu et aidé à surmonter les moments difficiles durant cette thèse, pour son amour, son soutien et sa confiance.

Tables des matières

Introduction	1
1 Systèmes à Evènements discrets : Principe et état de l'art	4
1.1 Introduction	5
1.2 Théorie des langages et des automates	6
1.2.1 Langages formels	6
1.2.2 Automates finis	7
1.3 Problème de synthèse de contrôleur à base des langages	9
1.3.1 Problème de supervision	10
1.3.2 Contrôlabilité et existence des superviseurs	11
1.3.3 Synthèse des superviseurs	12
1.4 Réseaux de Petri	12
1.4.1 Notions de base	13
1.4.2 Principe de fonctionnement et propriétés	14
1.5 Les réseaux de Petri pour la supervision et le contrôle	17
1.5.1 Problème d'états interdits	17
1.5.2 Problème de transitions d'états interdites	18
1.5.3 Contrôlabilité et existence des contrôleurs RdP	19
1.6 Approches de synthèse des contrôleurs RdP	20
1.6.1 Théorie des régions	21
1.7 Introduction à l'analyse combinatoire	28
1.7.1 Définitions	28
1.7.2 Combinaisons	29
1.8 Conclusion	29
2 Synthèse de contrôleurs RdP basée sur les coupes minimales	31
2.1 Introduction	32
2.2 Théorie des régions pour la synthèse de contrôle	32
2.2.1 Synthèse de places de contrôle pures	32
2.3 Synthèse de contrôle basée sur les coupes minimales	36
2.3.1 Approche 1 : Coupe minimale à instances de séparation d'évènements corrélées	37
2.3.2 Approche 2 : Coupe minimale à instances de séparation d'évènements quelconques	42
2.4 Coordination et Switch entre les contrôleurs RdP	44
2.4.1 Les vecteurs d'activation et de désactivation	44
2.4.2 Fonction booléenne de contrôle	45
2.5 Application : Commande d'un système de production flexible	46
2.5.1 Application de l'approche 1	47
2.5.2 Application de l'approche 2	49
2.6 Comparaison avec les travaux précédents	51
2.6.1 Exemple 1 : FMS, G_1	52
2.6.2 Exemple 2 : H	53
2.6.3 Exemple 3 : G_2	54
2.7 Conclusion	55

3	Commande optimale des RdP via les marquages canoniques et les contraintes de chemins admissibles	57
3.1	Introduction	58
3.2	Approche 1 : Synthèse de contrôleurs RdP basée sur les marquages canoniques	59
3.2.1	Conception des marquages canoniques	60
3.2.2	Exemple théorique	61
3.2.2.1	Détermination de la forme canonique de MI	63
3.2.2.2	Détermination des marquages canoniques	64
3.2.2.3	Application de la théorie des régions	64
3.2.3	Identification des instances de séparation d'évènements dans $G(N, M_0)$	65
3.2.3.1	Les vecteurs d'activation et de désactivation	65
3.2.3.2	Calcul du comportement maximum permissif	66
3.3	Approche 2 : Synthèse de contrôleurs RdP basée sur les chemins admissibles dans un graphe de marquage	68
3.3.1	Contraintes de chemins admissibles CCA	68
3.4	Application sur un système de production flexible	72
3.5	Conclusion	77
4	Synthèse de contrôleurs RdP sans génération du graphe de marquage	79
4.1	Introduction	80
4.2	Approche 1 : Théorie des régions sans graphe de marquage	80
4.2.1	Procédé de synthèse de commande	81
4.2.1.1	Détermination des équations de cycles de base	81
4.2.1.2	Détermination des conditions d'atteignabilité et de séparation	82
4.2.2	Exemple illustratif	85
4.2.3	Application sur un système de production flexible	88
4.3	Approche 2 : Synthèse de contrôle basée sur les propriétés des RdP	92
4.3.1	Détermination du système linéaire de la théorie des régions	93
4.3.2	Exemples	95
4.3.2.1	Exemple 1	95
4.3.2.2	Exemple 2	98
4.4	Implémentation des contrôleurs RdP sur un automate programmable (PLC)	102
4.4.1	Modélisation des stations d'assemblages par RdP	103
4.4.2	Programmation de l'automate programmable	104
4.4.3	Programmation des stations d'assemblage 3 et 4	105
4.5	Conclusion	107
	Conclusion	109
	Bibliographie	111

Table des figures

1.1	Changement d'état par occurrence des évènements d'un SED	5
1.2	Graphe de transition d'états d'un automate fini	7
1.3	Modèles accepteurs d'un distributeur de jeton	9
1.4	Schéma de la supervision	10
1.5	RdP marqué	13
1.6	Graphe de marquage	15
1.7	Principe du contrôle par retour d'état	18
1.8	Graphe d'atteignabilité d'un RdP réduit [HP11]	27
2.1	Un exemple de PTEI	33
2.2	Comportement désiré du RdP de la figure 2.1	33
2.3	Modèle RdP initial et son graphe d'atteignabilité	34
2.4	Modèle RdP contrôlé	35
2.5	Problème insoluble par la théorie des régions	35
2.6	Six types d'états	37
2.7	Un exemple de coupe minimale à instances de séparation d'évènements corrélées avec t_1	38
2.8	Un exemple de coupe minimale	41
2.9	Identification d'une <i>CM</i> en utilisant l'approche 2	42
2.10	Résolution du problème insoluble de la théorie des régions	44
2.11	Relation entre M_{0i} et Min_i	45
2.12	Partage de ressources	46
2.13	Environnement expérimental	47
2.14	Graphe de marquage 1 avec des coupes minimales	48
2.15	Modèle RdP contrôlé 1	49
2.16	Graphe de marquage 2 avec des coupes minimales	50
2.17	Modèle RdP contrôlé 2	51
3.1	Modélisation du trafic ferroviaire par un RdP	62
3.2	Synthèse des marquages canoniques	63
3.3	Modèle RdP contrôlé	65
3.4	Identification des instances de séparation d'évènements	67
3.5	Calcul du comportement maximum permissif	68
3.6	Un exemple RdP	69
3.7	Graphe de marquage généré	70
3.8	Chemins admissibles	71
3.9	Modèle RdP contrôlé	72
3.10	Système physique	73
3.11	Modèle RdP	73
3.12	La forme généralisée de <i>CMC</i>	74
3.13	Marquages canoniques	75
3.14	Modèle RdP contrôlé	76
3.15	Identification des instances de séparation d'évènements	76
3.16	Calcul du comportement maximum permissif	77
4.1	Exemple Illustratif	86
4.2	Méthode canonique	87
4.3	Modèle RdP contrôlé	88

4.4	Station de contrôle qualité	89
4.5	Transition incontrôlable	91
4.6	Modèle RdP contrôlé	92
4.7	Exemple RdP 1	95
4.8	Exemple RdP 2	99
4.9	Stations d'assemblage de l'atelier flexible	102
4.10	Modèle RdP des stations 3 et 4	103
4.11	Branchement entre le PC et l'automate	105
4.12	Organigramme du programme	106

Liste des tableaux

2.1	Informations sur les marquages des coupes minimales	48
2.2	Les vecteurs d'activation et de désactivation	49
2.3	Informations sur les marquages des coupes minimales	51
2.4	Les vecteurs d'activation et de désactivation	51
2.5	Les comparaisons de l'exemple 1	52
2.6	Contrôleurs RdP additionnés	53
2.7	Les comparaisons de l'exemple 2	53
2.8	Contrôleurs RdP additionnés	54
2.9	Les comparaisons de l'exemple 3	54
2.10	Contrôleurs RdP additionnés	55
3.1	Vecteurs d'activation et de désactivation	77
4.1	Les séquences utilisables et leurs marquages	87
4.2	Description du modèle RdP	89
4.3	Combinaisons admissibles de marquages	90
4.4	Combinaison interdite de marquages	90
4.5	Séquences utilisables	91
4.6	Identification des capteurs	103
4.7	Identification des effecteurs	104
4.8	Description des transitions/places	104

Introduction générale

Le contrôle des systèmes automatisés suscite depuis longtemps un grand intérêt dans le monde scientifique. De nombreuses recherches ont été faites pour la synthèse des contrôleurs. Cependant, les outils de base utilisés pour commander ces systèmes à événements discrets (SED), tels que le Grafset, consistaient à concevoir un modèle de la commande et à le valider par une série de tests. Néanmoins, cette méthode n'est pas applicable pour les systèmes complexes qui exigent un grand niveau de sécurité et de précision comme les centrales nucléaires, les raffineries de pétrole ou les avions. Ce type de système nécessite des méthodes formelles de contrôle permettant de garantir a priori un fonctionnement conforme aux spécifications. Une littérature abondante a été développée dans cette direction depuis 1980. De nombreuses théories ont été proposées, mais le problème reste évidemment d'actualité.

Les systèmes à événements discrets sont des systèmes événementiels à états discrets. Leur évolution se fait conformément à l'occurrence des événements caractérisant le changement d'état du système. Les SED ne se soucient pas du déroulement continu des phénomènes, de leur enchaînement logique, dynamique ou temporel. Les systèmes informatiques, les réseaux de files d'attente, les systèmes manufacturiers, et les systèmes de transport sont des exemples de systèmes à événements discrets.

Il existe plusieurs outils de modélisation et d'analyse des SED. Les réseaux de Petri (RdP), qui représentent l'outil choisi dans le travail présenté, en sont un exemple. Cependant, les automates, les chaînes de Markov et les modèles algébriques dynamiques, etc. sont aussi des exemples d'outils permettant l'étude des SED. Le choix des RdP n'a pas été fait au hasard. En effet, les réseaux de Petri se distinguent par des propriétés très intéressantes et avantageuses. Ils constituent, d'une part, un outil de modélisation graphique à interpréter et simuler pour les systèmes comportant du parallélisme et de la synchronisation. Et d'autre part, ils sont utilisés pour vérifier des propriétés comme la réversibilité ou l'absence de blocage grâce aux outils mathématiques disponibles pour l'analyse des RdP.

Les réseaux de Petri sont également très utilisés dans la synthèse de contrôle des SED. Si le comportement spontané d'un système est jugé non satisfaisant par rapport à une exigence donnée, alors le contrôle du SED est nécessaire. Ce contrôle est réalisable grâce à la présence de certains événements dits *contrôlables*. L'action de contrôle peut se traduire sous la forme d'un ensemble d'états à interdire ou de séquences d'événements à générer. Un état bloquant est un état qu'on cherche à interdire, tandis que deux tâches qui s'effectuent d'une manière alternative, sont un exemple de séquence souhaitée.

Ce sont les travaux de P.J Ramadge et W.M. Wonham qui constituent les bases de la théorie générale de la commande par supervision des SED [RW89]. Dans leur théorie, un superviseur est l'entité qui contrôle et qui a la tâche de limiter le comportement d'un système en observant les séquences d'événements qu'il génère et en lui introduisant les événements contrôlables. Dans [RW89], le superviseur est capable de réaliser un comportement désiré suivant les

exigences demandées. Dès lors, la publication de ces premiers résultats a suscité l'intérêt de la recherche sur la supervision et le contrôle des SED, et les approches se sont variées et multipliées.

Bien que le formalisme développé dans [RW89] permette la modélisation d'une large classe de SED, il est peu adapté aux systèmes réels à cause de sa sensibilité aux explosions combinatoires des états. C'est pour cette raison que de nombreuses tentatives ont été faites afin d'adapter la théorie générale de la commande des SED à d'autres formalismes de modélisation (Holloway et al., Yamalidou et al., et Ghaffari et al., etc.).

La méthode de Ghaffari et al. exploite les réseaux de Petri pour définir une approche de synthèse de contrôle pour les SED basée sur la théorie des régions. Cette théorie a fait l'objet de plusieurs travaux adaptés au problème de synthèse de contrôleurs RdP et qui traitent les problèmes d'états interdits.

La théorie des régions est une méthode de synthèse qui peut, à partir d'un RdP quelconque du système à contrôler, concevoir un ensemble de places de contrôle, constituant le *contrôleur RdP*, qui, une fois ajouté au modèle RdP initial, permet de réaliser exactement le comportement souhaité. Ce comportement est un comportement admissible correspondant à un problème de *transitions d'états interdits*, qui est un problème plus général qu'un *problème d'états interdits*. Par conséquent, il est possible de construire un contrôleur RdP maximum permissif pour tout modèle RdP sujet à n'importe quelle spécification de contrôle de type transitions d'états interdits.

Malheureusement, le caractère général de cette approche et l'élégance de la solution ont un prix. En effet, la taille des problèmes réels peut constituer une limite pour l'application de la théorie des régions pour la synthèse de contrôleurs RdP. La théorie des régions nécessite l'exploration du graphe d'états. Par conséquent, une possible explosion combinatoire des états peut rendre la synthèse de contrôleurs très complexe. De ce fait, nous avons cherché dans cette thèse à développer plusieurs approches avec lesquelles nous arrivons à réduire la complexité de calcul des contrôleurs RdP en utilisant toujours la théorie des régions et les réseaux de Petri. De surcroît, dans certaines approches, nous avons cherché à tirer profit des particularités et des propriétés des réseaux de Petri pour définir une logique de contrôle en considérant *les évènements incontrôlables*. En effet, nous sommes arrivés à nous déposséder entièrement du graphe d'états du réseau de Petri pour calculer le superviseur RdP. Par contre, nous avons fait l'hypothèse que tous les évènements sont observables pour le contrôleur.

Ce travail a par conséquent la structure suivante : dans le premier chapitre, nous présentons les principaux travaux existants sur la supervision et le contrôle des systèmes à évènements discrets. Le deuxième chapitre développe une méthode de synthèse de contrôle basée sur les coupes minimales dans un graphe de marquages d'un modèle RdP donné. Ensuite, dans le troisième chapitre, nous présentons une extension de l'approche de synthèse pour répondre aux exigences de contrôle. Enfin, le dernier chapitre développe deux nouvelles approches qui se basent sur la théorie des régions sans génération du graphe de marquage pour faire face aux problèmes d'états et de transitions d'états interdits.

Chapitre 1

Systemes à Evènements discrets : Principe et état de l'art

***Résumé:** Dans ce premier chapitre, nous donnerons une présentation succincte des notions de base des systèmes à événements discrets (SED). Nous commencerons par une introduction, puis nous exposerons les fondements de la théorie du contrôle et de la supervision des SED en nous basant sur les travaux de Ramadge et Wonham (R/W) sur l'existence et la synthèse des superviseurs. Ces travaux montrent qu'un procédé évolue en boucle fermée avec son superviseur. Ce dernier assure un fonctionnement admissible en agissant sur les séquences d'évènements générées afin d'interdire des évènements contrôlables. La majorité des résultats trouvés dans le domaine du contrôle par supervision se base essentiellement sur les travaux de R/W, qui utilisent les langages formels pour modéliser un SED et son superviseur.*

Dans ce chapitre, nous mettrons l'accent sur les approches qui ont recours aux réseaux de Petri (RdP). L'utilisation des RdP s'est dévoilée très intéressante dans la modélisation des SED et la résolution des problèmes d'états interdits. Nous exposerons dans ce chapitre trois approches utilisant les réseaux de Petri pour la synthèse de contrôle. Il s'agit des approches de Giua et al., l'approche de Ghaffari et al. et celle de Huang et al. Ces deux dernières approches utilisent la théorie des régions pour la synthèse des superviseurs RdP. Finalement nous présenterons quelques méthodes de dénombrements afin d'incorporer l'analyse combinatoire dans la synthèse des contrôleurs sous forme de places de réseaux de Petri. En conclusion, nous situerons nos travaux réalisés dans cette thèse par rapport aux approches existantes.

1.1 Introduction

Un système à évènements discrets (SED) est un système dynamique dans lequel l'espace d'état est discret. L'évolution de ces systèmes est évaluée par l'occurrence instantanée des évènements à des intervalles de temps éventuellement inconnus et/ou irréguliers. Les systèmes de communication et de transport, les systèmes informatiques et les systèmes manufacturiers sont des exemples de SED. En effet, tant qu'il n'y a pas d'occurrence d'un évènement, l'état du système reste inchangé. Un évènement peut correspondre à la transmission d'un message dans un réseau de communication, à l'occurrence d'une perturbation dans un réseau de transport ou encore à l'arrivée d'une pièce dans une cellule de production.

Considérons l'exemple classique d'une machine qui peut être dans trois états : marche, arrêt et panne. On suppose qu'il peut y avoir l'occurrence de quatre évènements : début de la tâche, fin de la tâche, panne et réparation. Ces évènements sont étiquetés respectivement par les symboles suivants : D , F , P , R . L'ensemble des états de notre système est présenté par l'ensemble E suivant : $E = \{m, a, p\}$. Le changement des états par l'occurrence des évènements est présenté dans la figure 1.1.

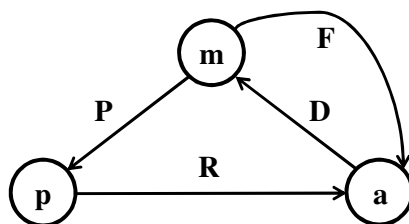


Figure 1.1-Changement d'état par occurrence des évènements d'un SED

Les systèmes à évènements discrets sont à l'heure actuelle d'une complexité telle que des méthodes intuitives ne suffisent plus afin d'assurer une supervision efficace et opérationnelle. Pour traiter de tels problèmes, la théorie de la supervision des SED a été développée par P.J. Ramadge et W.M. Wonham [RW89]. Grâce aux langages formels et aux automates, ces auteurs ont introduit des concepts semblables à ceux utilisés dans la théorie de la commande automatique des systèmes continus. Ainsi, le SED peut être considéré en boucle fermée avec son superviseur. Ce dernier permet d'assurer un comportement légal ou désiré. En effet, le superviseur gère l'évolution de l'état du système en contrôlant son comportement et en empêchant l'occurrence des évènements dits contrôlables ; les autres évènements qui ne peuvent être inhibés par le superviseur sont dits incontrôlables.

Les réseaux de Petri se sont montrés plus puissants que les automates pour modéliser les SED comportant la synchronisation et le parallélisme. De surcroît, les RdP sont plus avantageux pour traiter des problèmes de contrôle définis par des spécifications sur les états du système (voir l'exemple [LW94] et [YMLA 96]).

Dans ce chapitre, nous présentons une synthèse des approches de contrôle et de la supervision des SED. De plus, dans la section suivante, nous illustrons la théorie des langages formels et des automates qui constituent les modèles de base utilisés dans la théorie de la supervision des

SED [RW89]. La section 1.3 définit le problème de synthèse de contrôleur à base des langages et regroupe essentiellement les résultats de la synthèse de l'existence de superviseurs. Ensuite, dans la section 1.4, nous donnons une présentation succincte des notions de base des réseaux de Petri. En outre, on illustre dans la section 1.5 les approches de résolution des problèmes à états interdits et des problèmes de transitions à états interdits, particulièrement celle de Giua et al. [Giua96][GDS92] ; de plus, on présentera dans cette section des approches utilisant la théorie des régions pour la synthèse de contrôle, notamment l'approche de Ghaffari et al. [GRX03a][GRX03b][GRBX02a][GRX02b] et celle de Huang et al. [HP11] pour le problème d'évitement de blocage. Finalement, nous ferons un tour d'horizon sur l'analyse combinatoire et les méthodes de dénombrements. En conclusion de ce chapitre, nous situerons les travaux développés dans cette thèse par rapport aux approches existantes.

1.2 Théorie des langages et des automates

La théorie des langages et des automates constitue un des moyens formels afin de bien étudier le comportement logique des systèmes à événements discrets. En effet, les langages représentent un outil mathématique avantageux mais difficile à manipuler. Chaque SED est associé à un ensemble d'évènements Σ fini et l'alphabet du langage est représenté par Σ . Chaque symbole modélise un évènement. Par conséquent, toutes les séquences d'évènements sont appelées mots/chaines sur l'alphabet Σ . Les automates en sont une représentation graphique plus facile à manipuler. Pour des exemples plus détaillés, le lecteur pourra voir [CL08] et [HMU07].

1.2.1 Langages formels

L'ensemble de tous les mots/chaines qui peuvent être composés à partir des éléments de Σ , y compris la chaine vide ε , est noté Σ^* . La longueur d'un mot m est notée $|m|$ et représente le nombre de symboles de m .

Définition 1.1 :

On appelle un langage défini sur un alphabet Σ tout ensemble de Σ^* . Plus formellement, en désignant l'alphabet $\Sigma = \{u_1, v_2\}$, on obtient alors $\Sigma^* = \{\varepsilon, u_1, v_2, u_1u_1, u_1v_2, v_2u_1, u_1u_1u_1 \dots\}$. L'ensemble Σ^* définit l'ensemble de tous les mots finis sur un alphabet Σ .

Une séquence d'évènements u est un préfixe d'une séquence v de Σ^* , s'il existe une autre séquence w de Σ^* , telle que $v = u.w$ i.e., v est la concaténation de u et w . Si v est une séquence réalisable dans le système, alors tout préfixe de v l'est également. La chaine vide ε représente l'élément identité de la concaténation : $u.\varepsilon = \varepsilon.u = u$ pour toute chaine u . La préfixe-clôture du langage L est $\bar{L} = \{u \mid \exists v \in \Sigma^*, u.v \in L\}$.

Exemple 1.1 : Soit le langage $L = \{u_1u_1v_2v_2\}$, la préfixe-clôture de L sera définie par $\bar{L} = \{\varepsilon, u_1, u_1u_1, u_1u_1v_2, u_1u_1v_2v_2\}$. Par définition, on remarque que \bar{L} contient L . En effet, tout langage égal à sa préfixe-clôture est qualifié de préfixe-clos, c'est-à-dire que L est préfixe-clos, si $L = \bar{L}$.

Exemple 1.2: Soit le langage $L = \{u_1 u_1 v_2 v_2\}$, L n'est pas préfixe-clos, tandis que le langage $\bar{L} = \{\varepsilon, u_1, u_1 u_1, u_1 u_1 v_2, u_1 u_1 v_2 v_2\}$ est préfixe-clos.

Les expressions régulières sur un alphabet Σ sont des expressions dont les opérandes sont des symboles de Σ et dont les opérateurs appartiennent à l'ensemble $\{+, \cdot, ^*\}$.

Exemple 1.3 : L'expression $u_1 + u_1 \cdot v_2^*$ est une expression régulière.

Définition 1.2 :

Un langage régulier est un langage défini par une expression régulière.

Exemple 1.4 : Soit le langage régulier sur l'alphabet $\Sigma = \{u_1, v_2\}$, qui comprend tous les mots tels que u_1 et v_2 apparaissent alternativement et u_1 apparaît en premier lieu. Le langage $L = \{\varepsilon, u_1, u_1 v_2, u_1 v_2 u_1, u_1 v_2 u_1 v_2 u_1, \dots\}$; ce langage est régulier car il peut être décrit par l'expression régulière suivante : $(u_1 v_2)^* \cdot (\varepsilon + u_1)$. L'expression $\varepsilon + u_1 \cdot (v_2 u_1)^* \cdot (\varepsilon + v_2)$ est aussi une expression régulière qui peut traduire le langage L . Par conséquent, plusieurs expressions régulières peuvent correspondre à un même langage régulier.

1.2.2 Automates finis

Un système à événements discrets peut être représenté par un automate A sur un alphabet Σ avec un ensemble d'états Q . D'une manière générale, un automate est une machine à des entrées et des sorties discrètes et qui réagit à une modification de ses entrées en changeant des sorties. Formellement, un automate fini et déterministe A peut être défini de la façon suivante :

Définition 1.3: Un automate fini A est un 5-tuplet $A = (Q, \Sigma, \delta, q_0, Q_m)$ où :

- Q : est l'ensemble fini d'états.
- Σ : est un ensemble fini de symboles d'entrée i.e. l'ensemble des événements appelé l'alphabet d'entrée.
- δ : est la fonction de transition d'états de $Q \times \Sigma$ vers Q qui associe un état d'arrivée à un état de départ et un ensemble d'entrées.
- q_0 : est l'état initial appartenant à Q .
- $Q_m \subseteq Q$: est l'ensemble des états marqués.

Un automate A est représenté d'une manière générale par son *graphe de transitions d'états* donné dans la figure 1.2 suivante.

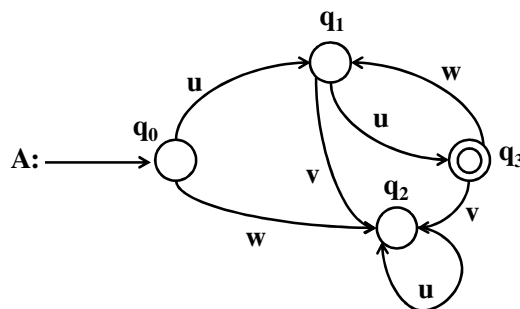


Figure 1.2- Graphe de transition d'états d'un automate fini

Les états de A sont modélisés par des cercles avec $Q = \{q_0, q_1, q_2, q_3\}$. L'état initial q_0 est indiqué par une flèche, tandis que les états finals sont représentés par des doubles cercles, donc $Q_m = \{q_3\}$. La fonction δ de transition d'états est représentée par des arcs associés à des symboles de l'alphabet Σ . Dans cet exemple, l'alphabet est donné par $\Sigma = \{u, v, w\}$. Il existe une transition d'état associée au symbole "u" entre q_0 et $q_1 \Rightarrow \delta(q_0, u) = q_1$.

Définition 1.4:

Un état $q \in Q$ est un état accessible s'il existe une chaîne $s \in \Sigma^*$, telle que $q = \delta(q_0, s)$, c'est-à-dire que l'automate peut y accéder depuis l'état initial. Par extension, l'automate A est accessible si tout état $q \in Q$ est accessible.

Le langage généré par un automate A est donné par : $L(A) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$; c'est un langage régulier représentant tous les mots qui peuvent rejoindre les états quelconque de l'automate à partir de l'état initial.

Remarque 1.1: L'automate A est dit déterministe dans le sens où depuis tout état, il n'existe pas deux transitions de sortie qui soient associées à un même symbole et qui conduisent à deux états différents.

a. Composition synchrone entre deux automates

Nous définissons dans cette section un autre type d'automate : l'automate accepteur. Cet automate ne comporte pas de sorties. Formellement, on définit un accepteur comme suit :

Définition 1.5:

Un accepteur A est un quadruplet $A = (Q, \Sigma, \delta, q_0)$ où Σ est un ensemble fini de symboles d'entrée (ensemble d'évènements), appelé alphabet d'entrées, Q est l'ensemble fini des états, δ est la fonction de transition d'états de $Q \times \Sigma$ vers Q qui associe un état d'arrivée à un état de départ et un symbole d'entrée et $q_0 \in Q$ représente l'état initial.

Dans la suite, le mot *accepteur* sera utilisé à la place d'automate. Cette hypothèse sera adoptée pour la composition synchrone entre deux automates. En effet, il existe deux sortes de compositions, la *composition synchrone* et la *composition asynchrone* des accepteurs. La composition synchrone est réalisée quand les alphabets des langages associés aux automates considérés ont au moins un évènement en commun. La composition asynchrone est effectuée quand les alphabets des langages associés aux automates considérés n'ont aucun évènement en commun. Soit les deux automates A_1 et A_2 suivants définis par : $A_1 = (Q_1, \Sigma_1, \delta_1, q_{01})$ et $A_2 = (Q_2, \Sigma_2, \delta_2, q_{02})$. En effet, une composition synchrone entre les deux automates A_1 et A_2 est l'accepteur $A_s = A_1 \parallel_s A_2 = (Q, \Sigma, \delta, q_0)$ où :

- $Q = Q_1 \times Q_2$ est l'ensemble des états.
- $\Sigma = \Sigma_1 \cup \Sigma_2$ représente l'alphabet
- $q_0 = (q_{01}, q_{02})$ est l'état initial.

Pour tout $q = (q_1, q_2) \in Q$ et pour tout $\delta \in \Sigma$ est la fonction de transition, δ est définie de la façon suivante :

- Pour $\delta \in \Sigma_1 \cap \Sigma_2$: on a $\delta[(q_1, q_2), \delta] = (q'_1, q'_2)$ si $\delta_1(q_1, \delta) = q'_1$ et $\delta_2(q_2, \delta) = q'_2$

- Pour $\delta \notin \Sigma_1 \cap \Sigma_2$: on a $\delta[(q_1, q_2), \delta] = (q'_1, q'_2)$ si $\delta_1(q_1, \delta)=q'_1$ et $\delta \in \Sigma_1$
 $\delta[(q_1, q_2), \delta] = (q'_1, q'_2)$ si $\delta_2(q_2, \delta)=q'_2$ et $\delta \in \Sigma_2$

Si un événement δ appartient à $\Sigma_1 \cap \Sigma_2$, il doit se produire de manière synchrone dans les deux automates, tandis que s'il n'appartient qu'à un ensemble, il se produit de manière asynchrone.

Exemple 1.5 : Le fonctionnement d'un distributeur de jetons est représenté par les deux modèles accepteurs de la figure 1.3 suivante.

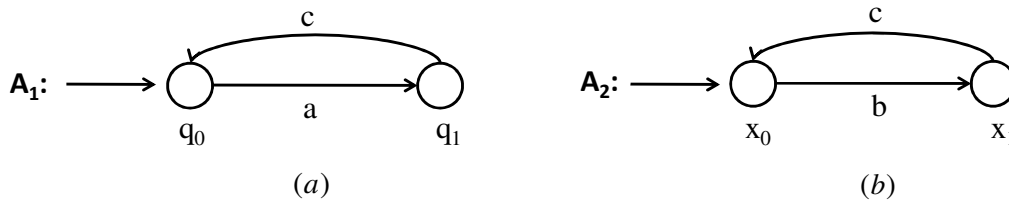


Figure 1.3- Modèles accepteurs d'un distributeur de jetons

L'accepteur A_1 de la figure 1.3.a est construit sur un alphabet de deux symboles $\{a, c\}$. Nous remarquons que l'accepteur A_1 reconnaît l'ensemble des séquences telles que a se produit en premier lieu et telles que a et c se produisent alternativement. Supposons que le symbole a modélise l'évènement, « L'introduction d'une pièce dans une machine », et que c représente l'évènement, « Libération d'un jeton ». Alors, A_1 modélise la contrainte de fonctionnement que le jeton est délivré seulement si une pièce a précédemment été introduite dans la machine (on suppose toujours que les évènements a et c apparaissent alternativement).

L'automate A_2 de la figure 1.3.b est construit sur un alphabet de 2 symboles $\{b, c\}$. Le symbole b modélise l'évènement « pression du bouton par l'utilisateur » et le symbole c a la même signification que dans A_1 (Libération d'un jeton). Ce modèle exprime la contrainte qu'un jeton est délivré seulement si l'utilisateur a auparavant appuyé sur le bouton.

A partir de A_1 et A_2 , l'opération de *composition synchrone* permet d'obtenir un modèle global de la machine. Ce modèle permet de traduire la conjonction de deux contraintes de fonctionnement définies par A_1 et A_2 . Dans l'accepteur A_3 , un état correspond à un couple (q, x) où q présente un état de A_1 et x est un état de A_2 . L'état initial (q_0, x_0) est l'état correspondant aux états initiaux de A_1 et A_2 . Cet accepteur est construit sur un alphabet, réunion des deux alphabets de A_1 et A_2 i.e. $\{a, b, c\}$ où le symbole c est le seul évènement qui est en commun dans les alphabets de A_1 et A_2 .

1.3 Problème de synthèse de contrôleur à base des langages

La théorie de la commande par supervision a été inventée par Ramadge et Wonham [RW89] pour faire face aux problèmes de supervision. En effet, le procédé est considéré comme évoluant en boucle fermée avec le superviseur (voir Figure 1.4). Ce dernier oriente l'évolution du système afin de restreindre son fonctionnement au comportement souhaité.

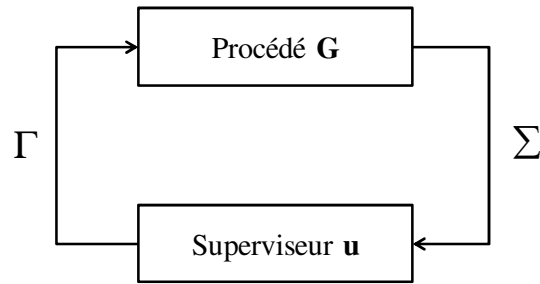


Figure 1.4-Schéma de la supervision

Des entrées du procédé sont les sorties du superviseur et vice versa. Le superviseur autorise/interdit l'occurrence des événements dits *contrôlables* Σ_c . Par contre, il ne peut pas inhiber l'occurrence des événements dits *incontrôlables* Σ_{uc} ; c'est ce qu'on appelle le contrôle par retour d'événements. Le superviseur qui réalise un comportement souhaité et qui laisse au système un maximum de liberté est dit *optimal* ou *maximum permissif*.

Dans cette section nous présentons le problème de la supervision ainsi que les conditions nécessaires pour l'existence et la synthèse du superviseur.

1.3.1 Problème de supervision

Considérons un SED $G = (\Sigma, Q, \delta, q_0, Q_m)$ avec $\Sigma = \Sigma_c \cup \Sigma_{uc}$. Le rôle du superviseur u consiste à définir pour chaque séquence d'évènement s généré par le système, un ensemble de continuations possibles. Ces continuations représentent une liste d'évènements $\gamma_s \subseteq \Sigma$. Si un évènement e appartient à γ_s , alors il est autorisé par u , suite à la séquence s , et il est interdit dans le cas contraire.

Le superviseur u doit respecter l'incontrôlabilité des événements de Σ_{uc} , son action de contrôle doit satisfaire l'expression suivante: $\forall s \in L(G), \Sigma_{uc} \subseteq \gamma_s$. Les événements Σ_{uc} sont toujours permis.

L'ensemble Γ des actions de contrôle correspond à l'ensemble de toutes les parties de Σ , i.e. $\Gamma = 2^\Sigma$. Formellement, un superviseur u est une fonction $u: L(G) \rightarrow 2^\Sigma$ qui associe à chaque séquence d'évènement s l'action de contrôle donnée par l'ensemble $u(s)$.

L'évolution du SED G est limitée sous l'influence du superviseur u . En effet, si G est dans un état $\delta(q_0, s) = q$, pour que la transition $\delta(q, e)$ ait lieu, il ne suffit plus qu'elle soit définie dans l'automate, il faut que $e \in u(s)$. Le comportement contrôlé (en boucle fermée) est défini comme suit :

Définition 1.6 :

Un comportement sous le contrôle d'un superviseur u est décrit par le langage $L(G, u)$ tel que :

- $\varepsilon \in L(G, u)$ et
- $se \in L(G, u)$ si et seulement si $s \in L(G)$, $e \in u(s)$ et $se \in L(G)$.

De la même manière, les séquences qui conduisent à des états marqués quand G est autonome ne sont pas toutes réalisables dans le système contrôlé. Celles qui sont encore possibles en présence du superviseur décrivent le langage marqué du système contrôlé :

$$L_m(G, u) = L_m(G) \cap L(G, u)$$

Quand les états marqués correspondent à des fins de tâches, ce langage devient important dans le sens où il caractérise des séquences *intéressantes* qui permettent d'accomplir des tâches tout en respectant la contrainte de supervision. Un superviseur u est *non-bloquant* pour un automate, si toute séquence du système à contrôler peut toujours être complétée pour atteindre un état marqué .i.e. $\overline{L_m}(G, u) = L(G, u)$.

De façon générale, le problème de supervision revient à restreindre le comportement d'un système en ajoutant un superviseur qui garantit un comportement en boucle fermée décrit par un langage clos spécifié $L_{spec} \subseteq L(G)$. L'objectif, c'est aussi d'avoir un certain langage marqué pour le système contrôlé tout en respectant la propriété du non-blocage.

1.3.2 Contrôlabilité et existence des superviseurs

La contrôlabilité des langages est introduite par Ramadge et Wonham pour démontrer l'existence des superviseurs pour un problème de synthèse de superviseur donné [RW89].

Définition 1.7:

Un langage $K \subseteq \Sigma^*$ est contrôlable par rapport à $L(G)$ si $\overline{K} \Sigma_{uc} \cap L(G) \subseteq \overline{K}$

C'est-à-dire que si K est contrôlable, alors toute séquence réalisable dans le système composé d'un préfixe quelconque d'une séquence K , suivi d'un événement incontrôlable, est aussi un préfixe d'une séquence de K .

En effet, l'existence d'un superviseur dépend des conditions nécessaires et suffisantes, données par le *théorème 1.1* suivant :

Théorème 1.1 ([RW89], [RW87]) :

Soit un *SED* non-bloquant G de langage $L(G)$ et de langage marqué $L_m(G)$.

1. Pour tout langage non vide $K \subseteq L(G)$, existe un superviseur u tel que $L(G, u) = K$ ssi K est un préfixe-clos et contrôlable par rapport à $L(G)$.
2. Pour tout langage non vide $K \subseteq L_m(G)$, il existe un superviseur non-bloquant u tel que $L_m(G, u) = K$ ssi $\overline{K} \cap L_m(G) = K$ (K est $L_m(G)$ -clos) et K est contrôlable par rapport à $L(G)$.

Les conditions décrites dans le *théorème 1.1* permettent de conclure quant à l'existence d'un superviseur pour réaliser un langage donné K .

Théorème 1.2 :

Soit un *SED* représenté par un automate à états finis G ($L(G)$ et $L_m(G)$ sont réguliers) et soit L_{spec} un langage de spécification régulier.

1. Si L_{spec} satisfait la condition 1 du théorème 1.1, alors le superviseur correspondant u peut être représenté par un automate à états finis.
2. De même, si L_{spec} satisfait la condition 2 du théorème 1.1, alors le superviseur correspondant u peut-être représenté par un automate à états finis. (pas de tiret à peut être)

1.3.3 Synthèse des superviseurs

Le langage des spécifications L_{spec} d'un procédé est réalisable si ce dernier est considéré comme contrôlable, tout en respectant l'incontrôlabilité des événements de Σ_{uc} . Par contre, pour faire respecter les spécifications, il faut limiter encore le comportement contrôlé à un sous-langage K de L_{spec} qui soit contrôlable. Néanmoins, afin de procéder au maximum d'états possibles, il est judicieux de choisir K comme le plus grand sous-langage contrôlable de L_{spec} .

Considérons l'ensemble $\mathcal{C}(L)$ des sous-langages contrôlables d'un langage L . Cet ensemble est automatiquement non vide car le langage vide est contrôlable. Par ailleurs, $\mathcal{C}(L)$ est clos par union, .i.e. l'union de deux éléments de $\mathcal{C}(L)$ est un élément de $\mathcal{C}(L)$. Par conséquent, $\mathcal{C}(L)$ admet un plus grand élément $Sup\mathcal{C}(L)$ et il est unique [RW87]. Ainsi, si L_{spec} n'est pas contrôlable, le langage $K = Sup\mathcal{C}(L_{spec})$ est la bonne approximation qui permet de respecter les spécifications en restreignant le procédé le minimum possible. Le superviseur qui réalise un langage pareil est *maximum permissif* ou *optimal*.

Généralement, il n'existe pas d'algorithmes performants pour le calcul du plus grand sous-langage contrôlable d'un langage de spécification. Si on se trouve dans le cas particulier où les langages de spécifications et du procédé sont réguliers et admettent des représentations sous forme d'automates à états finis, toutes les propriétés requises pour l'existence du superviseur sont décidables par une exploration des espaces d'états des automates.

De surcroît, il a été prouvé que le plus grand langage contrôlable est régulier [WR87]. Un algorithme a été développé par Ramadge et Wonham pour le calcul de ce langage. Cet algorithme est polynomial dans la taille des automates de spécification et du procédé [GW00].

Bien que les automates présentent un formalisme approprié à la représentation efficace d'une large classe des SED, la modélisation des systèmes à événement discrets réels reste généralement difficile à mettre en œuvre à cause de l'explosion combinatoire des états (même pour les petits systèmes).

1.4 Réseaux de Petri

Les réseaux de Petri (RdP) présentent un outil très approprié pour étudier les systèmes à événements discrets grâce à leur puissance de modélisation et leurs propriétés mathématiques. En les comparant aux automates, les RdP ont l'avantage d'être un modèle beaucoup plus général, bénéficiant de structures beaucoup plus riches. Les RdP s'adaptent parfaitement à la description de certains types de SED. En effet, un automate n'est autre qu'un graphe de marquage généré à partir d'un RdP.

Dans cette section, nous présenterons d'abord les notions de base et les notations des réseaux de Petri utilisées pour présenter nos travaux. Pour plus amples détails, le lecteur pourra consulter les travaux suivants: [Mur89], [PX96], [CL08] et [DA10].

1.4.1 Notions de base

Un réseau de Petri est un graphe orienté qui comporte deux types de nœuds ; des places et des transitions, et des arcs les reliant. La façon dont ces places et ces transitions sont reliées définit le comportement d'un RdP. Une transition est présentée sous forme d'une barre ou d'un rectangle, tandis qu'une place est modélisée par un cercle et peut contenir des marques ou jetons. Le RdP est dit marqué.

Définition 1.8 :

Formellement, un RdP est un quadruplet $N = (P, T, F, w)$ où :

- $P = \{p_1, \dots, p_n\}$ est un ensemble fini de places ;
- $T = \{t_1, \dots, t_m\}$ est un ensemble fini de transitions ;
- $F \subseteq (T \times P) \cup (P \times T)$ est un ensemble fini d'arcs ;
- $w: F \rightarrow \mathbb{N}$ est la fonction d'évaluation des arcs, (\mathbb{N} est l'ensemble des entiers naturels).

Si les poids des arcs sont tous égaux à 1, alors le RdP est dit *ordinaire*.

Dans cette thèse, comme nous présumons que $T = T_{uc} \cup T_c$, les transitions contrôlables sont représentées par des rectangles et les transitions incontrôlables par des barres.

Les jetons ou les marques modélisent la dynamique du système modélisé par leur existence ou leur absence dans les places.

a- Le marquage

Le marquage d'un RdP est un vecteur de dimension n (n : nombre de place du réseau) à composantes entières positives ou nulles. Un RdP marqué est noté (N, M_0) où M_0 est le marquage initial du RdP. La figure 1.5 représente un réseau de Petri marqué.

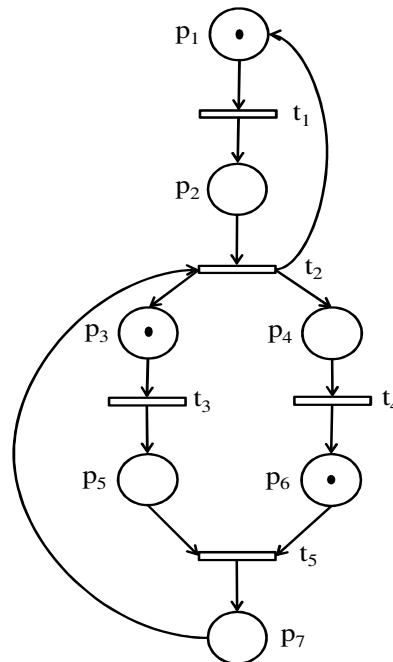


Figure 1.5- RdP marqué

Dans ce RdP marqué, chaque place contient un nombre entier (positif ou nul) de marques. Le nombre de jetons dans une place p_i sera noté $M(p_i)$. Pour cet exemple, on a $M(p_1) = M(p_3) = M(p_6) = 1$ et $M(p_2) = M(p_4) = M(p_5) = M(p_7) = 0$. Un chemin orienté dans un RdP est une séquence alternée de places et transitions, qui relie deux sommets distincts du réseau. Nous définissons le marquage d'un chemin orienté $\pi = t_1 p_1 \dots p_n t_{n+1}$ à un marquage M , noté $M(\pi)$ comme la somme des marquages des places appartenant à π , i.e. $M(\pi) = M(p_1) + \dots + M(p_n)$.

Pour un arc (p, t) , on dit qu'une place p (respectivement une transition t) est une place d'entrée de la transition t (respectivement une transition de sortie de la place p). On dit qu'une place est *impure* (par opposition à la place *pure*) si elle est à la fois une place d'entrée et de sortie pour une même transition t .

Notation 1.1 : Soit $p^{(t)}$ (respectivement ${}^{(t)}p$) l'ensemble des transitions de sortie (respectivement d'entrée de la place p) et $t^{(p)}$ (respectivement ${}^{(p)}t$) désigne l'ensemble des places de sortie (respectivement d'entrée) de la transition t .

1.4.2 Principe de fonctionnement et propriétés

Un ensemble fini d'évènements définit une transition t . Cette transition est franchissable à partir d'un marquage M si toute place d'entrée de t (${}^{(t)}p$) contient un nombre de jetons aux moins égal au poids de l'arc qui relie p à t i.e. $\forall p \in {}^{(t)}t, M(p) \geq w(p, t)$. Une transition est donc validée et on écrit $M[t >$.

Le marquage $M(p)$ est alors transformé en un marquage $M'(p)$ comme suit:

$$M'(p) = \begin{cases} M(p) - w(p, t) & \text{Si } p \in {}^{(p)}t \\ M(p) + w(p, t) & \text{Si } p \in t^{(p)} \\ M(p) & \text{Sinon} \end{cases}$$

L'ensemble des marquages atteignables à partir du marquage initial M_0 est noté M . Cet ensemble constitue alors un graphe de marquage noté $G(N, M_0)$. Dans $G(N, M_0)$, les transitions sont représentées par des arcs et les nœuds modélisent les marquages ou les états obtenus.

Dans l'exemple de la figure 1.5, le marquage initial du RdP est $M_0^T = [1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]$. A partir de M_0 , seules les transitions t_1 et t_3 sont validées. Si t_1 est franchie, on atteint l'état $M_1^T = [0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]$. Dans ce cas, seule la transition t_3 est validée. Quand la transition t_3 est franchie, on atteint l'état $M_2^T(p) = [1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0]$. Dans ce cas, les deux transitions t_1 et t_5 sont franchissables. Avec ce principe, tous les états atteignables à partir de M_0 peuvent être calculés. La figure 1.6 suivante illustre le graphe de marquage du modèle RdP de la figure 1.5.

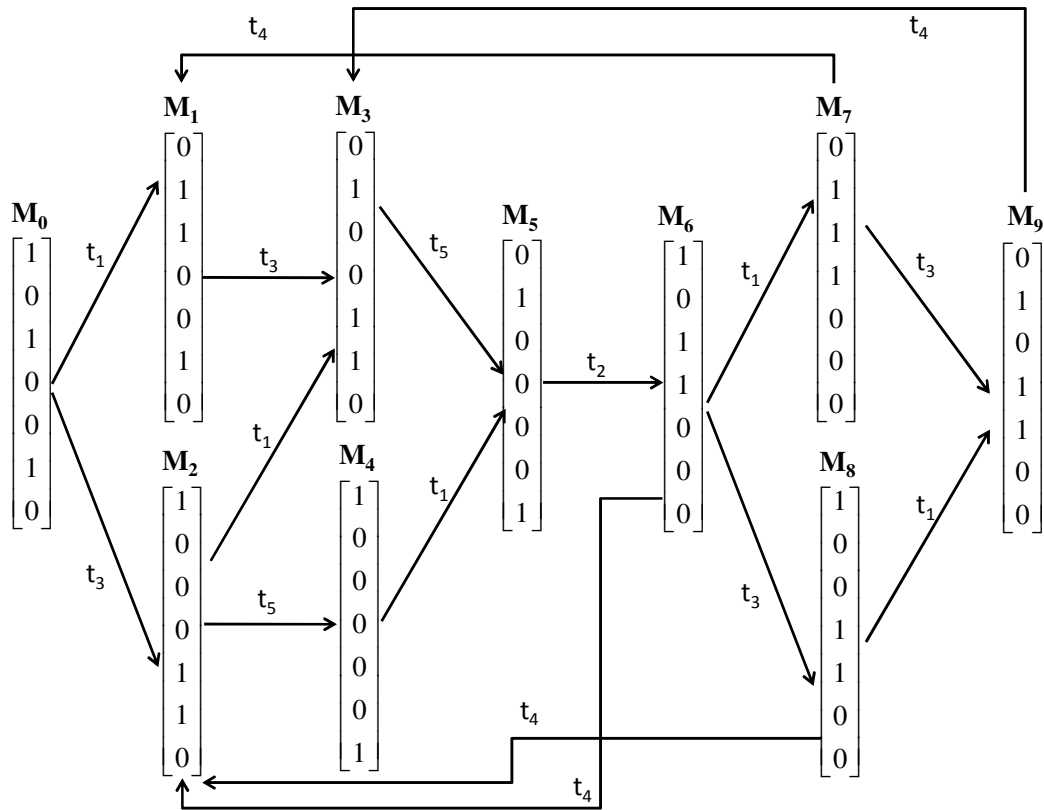


Figure 1.6- Graphe de marquage

a- Equation fondamentale

Une séquence de transitions franchissables à partir de l'état initial M_0 est notée σ . On associe un vecteur d'occurrences à cette séquence notée $\vec{\sigma}$ de dimension $\text{card}(T) = m$. La $i^{\text{ème}}$ composante de ce vecteur correspond au nombre d'occurrences de la transition t_i dans σ . L'équation fondamentale (ou équation d'état) des RdP est donnée par :

$$M = M_0 + C \cdot \vec{\sigma} \quad (1.1)$$

(Avec) M est le marquage obtenu en franchissant σ à partir de M_0 et C est la matrice d'incidence de (N, M_0) . Cette matrice C est définie comme suit :

$$C = [C(P_i, t_j)]_{i=1..n, j=1..m} = \begin{cases} w(t_j, p_i) & \text{Si } t_j \in {}^{(t)}p_i \\ -w(p_i, t_j) & \text{Si } t_j \in p_i^{(t)} \\ 0 & \text{Sinon} \end{cases}$$

Une colonne de la matrice d'incidence correspond aux modifications apportées aux marquages des places par la validation de la transition correspondante. C peut être défini par la différence entre deux matrices : C^+ la matrice d'incidence avant et C^- la matrice d'incidence arrière, telles que :

$$C(P_i, t_j)^+_{i=1..n, j=1..m} = \begin{cases} w(t_j, p_i) & \text{Si } t_j \in {}^{(t)}p_i \\ 0 & \text{Sinon} \end{cases}$$

$$C(P_i, t_j)_{i=1..n, j=1..m}^- = \begin{cases} w(p_i, t_j) & \text{Si } t_j \in p_i^{(t)} \\ 0 & \text{Sinon} \end{cases}$$

b- Quelques propriétés

Les RdP présentent un riche ensemble de propriétés intéressantes pour modéliser les systèmes manufacturiers et le partage des ressources. Nous présenterons dans la suite les propriétés qui nous seront utiles dans le cadre de ce travail :

RdP borné :

Une place $p_i \in P$ d'un RdP est dite k -bornée, $k \in \mathbb{N}$, pour un état initial M_0 donné, si $M(p_i) \leq k$ pour tout état accessible $M \in M$. Le RdP est appelé borné si toutes ses places sont bornées.

Remarque 1.2 : Un RdP est appelé *sauf* pour un marquage initial M_0 donné, si toutes ses places sont 1-bornées (contenant au plus un jeton).

RdP vivant :

Un RdP est vivant si chacune de ses transitions est vivante, c'est-à-dire que, quel que soit le marquage M , il existe toujours un marquage atteignable à partir de M qui permet de franchir t .

RdP persistant :

Un RdP est appelé persistant si, pour n'importe quelle paire de transitions validées, le franchissement de l'une ne rend pas l'autre infranchissable, c'est-à-dire qu'en cas de conflit entre deux transitions, le choix qui est fait sur l'ordre de franchissement n'a aucune conséquence sur la possibilité de franchir ces transitions. N'importe quelle

Blocage d'un RdP :

Un blocage est un marquage tel qu'aucune transition ne peut être validée.

RdP réversible :

Un RdP est réversible si, à partir de n'importe quel état atteignable M , il existe une séquence franchissable qui permet de revenir à M_0 .

RdP conservatif :

Un RdP est dit conservatif, s'il existe un vecteur de pondération des places X à composantes strictement positives tel que, pour tout marquage initial M_0 et tout marquage M atteignable à partir de l'état M_0 :

$$X^t \cdot M = X^t \cdot M_0$$

Conflit structurel :

Une place $p_i \in P$ constitue un conflit structurel si elle est une place d'entrée pour au moins deux transitions.

Conflit effectif :

On parle d'un conflit effectif entre deux transitions en conflit structurel s'il existe un marquage qui sensibilise les deux transitions et que le franchissement d'une transition

empêche le franchissement de l'autre : une seule transition sera franchie, mais rien dans le réseau ne permet de prévoir laquelle.

c- Les invariants

p-invariant :

Un p -invariant d'un réseau de Petri est un vecteur X d'entiers positifs de dimension $\text{card}(P) = n$ et qui vérifie :

$$X.C = 0$$

Où C est la matrice d'incidence du RdP.

En utilisant l'équation fondamentale des RdP, une propriété importante des p -invariants découle de leur définition, à savoir qu'un p -invariant X vérifie : $X^t.M = X^t.M_0$, et ce pour n'importe quel marquage accessible M . Ce qui signifie que la somme des marquages des places pondérées par le vecteur X est constante, quelle que soit l'évolution du RdP :

$$q_1M_1 + q_2M_2 + \dots + q_nM_n = \text{constante} \quad \forall M \in M \mid \forall i \in [1..n], q_i \in \mathbb{N}$$

t-invariant :

Un t -invariant est un vecteur Y d'entiers positifs de dimension $\text{card}(T) = m$ vérifiant :

$$C.Y^t = 0$$

Soit σ une séquence franchissable à partir de M_0 telle que $\vec{\sigma}$ est un t -invariant. Alors, en franchissant σ , on revient au marquage initial M_0 . Cette propriété découle directement de la définition des t -invariants et de l'équation fondamentale des RdP.

1.5 Les réseaux de Petri pour la supervision et le contrôle

Nous avons présenté dans la section 1.3 les problèmes de supervision où les spécifications s'expriment sous forme d'un comportement souhaité du procédé. Les actions de contrôle émises par le superviseur ne dépendaient que des séquences d'événements générées par le procédé. Cependant, nous définissons un problème de supervision par un ensemble d'états admissibles pour le procédé ou encore par un ensemble d'états interdits. Il est donc possible de synthétiser un superviseur qui interdit au procédé d'atteindre certains états interdits en se basant sur la séquence d'événements générés.

La politique la plus communément adoptée pour les problèmes d'états interdits est de type retour d'état. Le superviseur ne mémorise pas l'historique du système. L'action de contrôle admet toujours une liste d'événements tolérés. Elle est déterminée en observant uniquement l'état courant du système contrôlé.

Dans cette section, nous avons tout d'abord défini les problèmes d'états interdits et des transitions d'états interdites ; par la suite, nous présenterons les conditions nécessaires et suffisantes pour l'existence des contrôleurs RdP.

1.5.1 Problème d'états interdits (PEI)

Formellement, un contrôle par retour d'état d'un RdP (N, M_0) est une fonction $f: M \rightarrow \Gamma$ où Γ est l'ensemble des lois de contrôle défini par $\Gamma = \{\gamma \subseteq \Gamma \mid \gamma \supseteq T_{uc}\}$ (voir Figure 1.7).

Un réseau muni d'un contrôle f est noté N_f . Ainsi, une transition t est validée à partir d'un marquage M dans N_f si et seulement si $M[t >$ et $t \in f(M)$. Selon la définition de Γ , les transitions incontrôlables sont toujours autorisées par le contrôleur. On désignera par M_f l'ensemble des états atteignables dans N_f et R_f le graphe d'atteignabilité correspondant.

Dans un problème d'états interdits, notre objectif est de concevoir un contrôleur qui garantit que les états atteignables du système contrôlé soient dans un ensemble d'états admissibles spécifié, c'est-à-dire que le système contrôlé ne visitera jamais des états interdits bien définis.

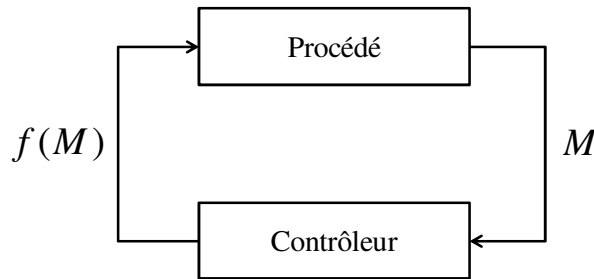


Figure 1.7- Principe du contrôle par retour d'état

Les états autorisés ou admissibles sont des états qui satisfont un prédicat Q défini comme suit : $Q: M \rightarrow \{0, 1\}$. Ces états admissibles sont identifiés aux éléments de l'ensemble :

$$M_Q = \{M \mid M \in M, Q(M) = 1\}$$

Par conséquent, l'ensemble complémentaire de M_Q est l'ensemble des états interdits.

Notation 1.2 : Soit $(M[\sigma > M')$ le chemin qui relie un état M à un autre état M' en franchissant dans un graphe la séquence de transitions σ .

1.5.2 Problème de transitions d'états interdites (PTEI)

Le problème d'états interdits représente un cas particulier des problèmes de transitions interdites (c'est-à-dire $PEI \subseteq PTEI$).

Définition 1.9 :

Un problème de transitions d'états interdites (ou PTEI) correspond à un comportement admissible R défini par :

- 1- $M_0 \in R$
- 2- $\forall M \in R, \exists \sigma \mid (M_0[\sigma > M) \in R$

L'admissibilité du marquage initial constitue la première condition nécessaire pour l'existence du contrôle. De plus, la deuxième condition exige l'atteignabilité de tout marquage admissible à partir de M_0 par un chemin dans R , ou encore que tous les marquages intermédiaires soient admissibles. On note que pour un PTEI, des transitions peuvent exister entre des états admissibles, mais qui n'appartiennent pas à R .

Le problème de transitions d'états interdites (PTEI) est un problème plus général que celui du problème d'état interdit (PEI). Un comportement admissible par rapport à des spécifications de type états interdits peut être défini comme suit :

Définition 1.10 :

Un problème d'états interdits (ou PEI) correspond à un comportement admissible qui satisfait les conditions 1 et 2 de la définition 1.9 et la condition supplémentaire suivante :

$$3- \forall (M[t > M'] \in R(N, M_0), M, M' \in R \Rightarrow M[t > M'] \in R$$

Remarque 1.3 : Un problème d'états interdits est un problème de transitions d'états interdites, mais l'inverse est faux.

1.5.3 Contrôlabilité et existence des contrôleurs RdP

Les travaux de Y.Li et W.M.Wonham [LW93] ont abouti à l'existence d'une solution pour les problèmes d'états interdits. Ces auteurs ont exprimé la propriété de contrôlabilité dans le contexte du contrôle par retour d'état. Leur recherche portait sur les *systèmes à évènements discrets vectoriels* (SEDV).

Définition 1.11 :

Un SEDV est un SED $G = (\Sigma, M, \delta, X_0)$ dont l'état est décrit par des variables d'état entières $x_0, x_1..x_n$ [LW93]. L'espace M des états du système est l'ensemble des vecteurs entiers de dimension n .

La fonction de transition d'un SEDV est $\delta: M \times \Sigma \rightarrow M$ telle que

$$\delta(M, e) = M + E_e$$

E_e est le vecteur de déplacement correspondant à l'évènement e . $\delta(M, e)$ est défini si $M \geq F_e$, où $F_e \in M$ est le *vecteur de condition d'occurrence*.

Les RdP constituent un modèle équivalent aux SEDV [LW93] [Pet81]. Les résultats démontrés pour les SEDV sont donc valides pour les réseaux de Petri.

Soit un RdP (N, M_0) et Q un prédicat de spécifications. On suppose que l'état initial satisfait Q , i.e. $M_0 \in M_Q$. Soit $R_Q(N, M_0)$ l'ensemble des états atteignables à partir de M_0 défini comme suit :

Définition 1.12 :

$R_Q(N, M_0)$ est tel que :

- $M_0 \in R_Q(N, M_0)$
- Si $M \in R_Q(N, M_0)$ et $M[t > M'] \in M_Q$, alors $M' \in R_Q(N, M_0)$
- Tout état de $R_Q(N, M_0)$ satisfait la première et la deuxième condition.

Il en découle que pour tout état $M \in R_Q(N, M_0)$, $\exists M_1, M_2..M_p \in M$ et $t_1, t_2..t_{p-1} \in \Sigma$ tels que :

- $\forall i = 0, 1, \dots, p - 1, M_i[t_i > M_{i+1} ;$

- $\forall i = 0, 1, \dots, p - 1, M_i \in R_Q(N, M_0)$;
- $M_p = M$.

Par ailleurs, pour tout marquage M qui satisfait Q , il existe toujours une séquence franchissable à partir de M_0 qui atteint M et telle que les états intermédiaires satisfont également le prédicat Q . La définition suivante introduit la notion de contrôlabilité d'une spécification d'états interdits Q [LW93].

Définition 1.13 :

Soit un réseau (N, M_0) et une spécification d'état interdits Q définie sur l'ensemble M . M_Q est contrôlable par rapport à (N, M_0) si :

- $M_Q = R_Q(N, M_0)$ et
- $\forall e \in T_{uc},$ si $M \in M_Q$ alors $M[t > M' \in M_Q$

On a donc toujours $R_Q(N, M_0) \subseteq M_Q$. La première condition de cette définition impose que tous les marquages admissibles par rapport à Q soient atteignables par l'intermédiaire d'états admissibles. La deuxième condition signifie que les états atteints sont admissibles de façon incontrôlable à partir des états admissibles. La condition de contrôlabilité de l'ensemble d'états est nécessaire et suffisante pour l'existence d'un contrôle par retour d'état, comme il est énoncé dans le théorème suivant [LW93].

Théorème 1.3 :

Soit une spécification Q sur les états d'un réseau de Petri (N, M_0) , telle que $M_0 \in M_Q$. Il existe un contrôle par retour d'état F tel que $M_F = M_Q$ si et seulement si M_Q est contrôlable.

Il existe un autre résultat concernant l'existence du plus grand sous-ensemble de marquages admissibles contrôlable d'un ensemble M_Q [LW93].

Théorème 1.4 :

Le plus grand élément de l'ensemble $F(M_Q) = \{K \mid K \subseteq M_Q \text{ et } K \text{ contrôlable}\}$ existe toujours.

1.6 Approches de synthèse des contrôleurs RdP

De nombreuses approches ont été développées pour le contrôle des systèmes à évènements discrets (voir par exemple [Kro87] [BBN95][Giua 96]). Beaucoup d'entre eux, tout comme la présente thèse, ont synthétisé leur contrôleur sous forme de places réseaux de Petri. En effet, ces auteurs ont bénéficié des nombreux avantages que présentent les RdP ; d'une part, on peut facilement étudier le comportement du processus contrôlé grâce aux outils mathématiques des RdP et implémenter par la suite l'agent de contrôle, et d'autre part, le calcul en ligne du *contrôleur RdP* se voit gagner en rapidité puisque la décision de contrôle est définie implicitement par l'état d'un RdP.

Toutefois, la synthèse d'un contrôleur RdP maximum permissif n'est pas du tout une tâche facile, vu la présence des transitions incontrôlables et la complexité de calcul des places de contrôles, en parcourant le graphe de marquages du RdP.

La méthode proposée par [GRX02a][GRX02b][GRX03a][GRX03b] dans la synthèse de contrôle est basée sur la théorie des régions [BD98]. Cette théorie est représentée par un système linéaire (*SL*) composé de trois types de conditions et sa résolution conduit à la synthèse d'un contrôleur RdP.

D'autres recherches ont été menées pour synthétiser des superviseurs RdP sous certaines hypothèses bornées afin de respecter des contraintes disjonctives [IA07]. En outre, des situations peuvent exister où le nombre de contrôleurs RdP est exponentiellement lié à la taille de la spécification. Dans [IWZA13], une approche améliorée qui génère des moniteurs RdP a été introduite dans laquelle le nombre des contrôleurs est linéairement lié à la taille des spécifications. Cependant, ces travaux ont été réalisés sous une contrôlabilité totale. De plus, dans [RCGH12], on synthétise un superviseur RdP avec des spécifications d'états interdits arbitraires. Leur politique de contrôle est déterminée en vérifiant certaines transitions contrôlables. Si on aboutit à un marquage réduit qui appartient aux marquages interdits, le contrôleur désactive la transition contrôlable correspondante.

Dans les travaux d'Uzam et al. [UGT15], une nouvelle méthode appelée TGALW a été présentée afin de synthétiser des contrôleurs RdP pour les systèmes de productions flexibles exposés aux problèmes d'évitement de blocage. En effet, le RdP original est d'abord transformé en un réseau équivalent. Ensuite, *une place source* est temporairement utilisée puis retirée une fois que le système obtenu est vivace. A chaque itération, les marquages dits *mauvais* sont identifiés puis contrôlés par des moniteurs RdP à travers une méthode basée sur la méthode des *p*-invariants. Si tous les mauvais marquages sont contrôlés de façon optimale, alors la méthode TGALW peut obtenir un contrôleur RdP optimal.

La politique de contrôle adoptée dans la présente thèse est basée principalement sur l'utilisation de la théorie des régions qui a été reconnue comme l'une des méthodes les plus puissantes pour la prévention de blocage et l'obtention des contrôleurs maximum permissifs. Plusieurs auteurs ont travaillé sur la réduction du coût de calcul de la théorie des régions pour la synthèse de contrôle, pareillement aux travaux présentés dans ce travail. En effet, dans [LZJ08], une approche appelée contrôle de siphons a été développée avant d'utiliser la théorie des régions pour calculer les contrôleurs RdP. En outre, dans [HP11], les auteurs proposent une autre issue pour le problème d'évitement de blocage qui diminue le nombre des contraintes de la théorie des régions afin de faciliter le calcul des superviseurs RdP. Les résultats de [HP11] semblent être les plus efficaces dans la politique de prévention de blocage par rapport aux travaux de [UZ07], [Uzam09], [PR00] et [QLZKM12]. Dans la suite, nous présentons la théorie des régions ainsi que les approches de Giua, Ghaffari et celle de Huang.

1.6.1 Théorie des régions

La théorie des régions (TR) a été développée originellement pour la synthèse des réseaux. En effet, on pourra approximer au mieux un langage préfixe clos non vide par le biais d'un langage d'un réseau de Petri [BBD97][BD98] et enfin déterminer ce réseau.

E.Badouel, L.Bernardinello et Ph.Darondeau ont construit un algorithme qui résout le problème de synthèse des RdP purs pour les automates finis [BBD95]. Cet algorithme est basé sur la notion de *région*. Cette notion a été définie par Ehrenfeucht et Rozenberg [ER90a][ER90b].

Définition 1.14 :

Une région peut être définie comme une propriété satisfaite par les nœuds/états qu'elle contient, et reste uniformément validée, invalidée, ou laissée inchangée par toutes les occurrences de chacune des actions qui étiquettent les arcs du graphe.

Pour plus de détails et de références bibliographiques relatifs à la théorie des régions, le lecteur peut consulter [Pro01].

a- Théorie des régions pour la synthèse des réseaux purs

Soit T un ensemble de transitions et G un graphe orienté tel que ses arcs sont étiquetés par des transitions de T . On suppose que pour chaque état du graphe, il existe au moins un chemin non orienté le reliant à chacun des autres états.

Le but de la TR est la synthèse d'un RdP pur (N, M_0) caractérisé par une matrice d'incidence C , tel que son graphe de marquage soit isomorphe à G .

Si on cherche une place p dans (N, M_0) , elle sera déterminée par le vecteur de ses coordonnées $C(p, .)$ dans la matrice C . Pour tout couple (M, t) où t est franchissable à partir de M , p doit vérifier :

$$M'(p) = M(p) + C(p, t), \forall (M', M) \in G \mid M[t > M' \quad (1.2)$$

Avec M' l'état de destination de l'arc étiqueté par t .

Si on considère maintenant les cycles non orientés tels que $\delta \in G$, l'application de la dernière équation le long du cycle aboutit à l'équation du cycle suivante :

$$\sum_{t \in T} C(p, t) \cdot \vec{\delta}[t] = 0, \forall \delta \in G \quad (1.3)$$

$\vec{\delta}[t]$ désigne la somme algébrique de toutes les occurrences de t dans δ .

En outre, pour chaque état $M \in G$, il existe au moins un chemin Γ_M non orienté le reliant à M_0 . En appliquant l'équation 1.2 à tous les arcs de Γ_M , on obtient :

$$M(p) = M_0(p) + C(p, .) \overrightarrow{\Gamma_M} \quad (1.4)$$

$\overrightarrow{\Gamma_M}$ est le vecteur d'occurrences de Γ_M . En appliquant l'équation des cycles (1.3), le produit $C(p, .) \overrightarrow{\Gamma_M}$ reste le même pour tout Γ_M . C'est pourquoi Γ_M pourrait être choisi arbitrairement. Donc, M est accessible dans (N, M_0) si l'équation d'atteignabilité suivante est satisfaite :

$$M(p) = M_0(p) + C(p, \cdot) \overrightarrow{\Gamma}_M \geq 0 \quad (1.5)$$

Ces conditions ne permettent pas d'obtenir un graphe d'atteignabilité G . Une condition supplémentaire doit être ajoutée. En effet, pour chaque couple (M, t) avec $M \in G$ et t une transition non validée, t doit être inhibée à partir de M (par le marquage d'une certaine place p). Cette condition est appelée *condition de séparation d'évènement* relative au couple (M, t) :

$$M_0(p) + C(p, \cdot) \overrightarrow{\Gamma}_M + C(p, t) < 0 \quad (1.6)$$

Définition 1.15 :

L'ensemble des couples (M, t) où M est atteignable et t est non validée par M , est appelé ensemble des instances de séparation d'évènements. Cet ensemble est noté Ω .

Enfin, une dernière condition, *la condition de séparation d'état* qui est nécessaire pour avoir G comme graphe d'atteignabilité. Cela signifie que les marquages sont deux à deux différents i.e. : $\forall M, M' \in G, \exists p \mid M(p) \neq M'(p)$ ou encore $\forall M, M' \in G, \exists p \mid C(p, t) \overrightarrow{\Gamma}_M \neq C(p, t) \overrightarrow{\Gamma}_{M'}$.

b- L'approche de Giua et al.

La méthode des invariants est basée sur l'expression de la spécification sous la forme d'un système *des contraintes généralisées d'exclusion mutuelle (CGEM)* ([Giua96][GDS92]).

Une contrainte de type *CGEM* est donnée par un couple (\vec{w}, K) , avec \vec{w} : un vecteur d'entiers, et K une constante entière, qui définit l'ensemble des états admissibles suivants :

$$M_{(\vec{w}, K)} = \{M \in M \mid \vec{w}^T M \leq K\} \quad (1.7)$$

Si on a un ensemble de contraintes *CGEM* (W, \vec{K}) où $W = [\vec{w}_1, \vec{w}_2, \dots, \vec{w}_n]$ et $\vec{K} = (k_1, \dots, k_n)$ alors l'ensemble des états admissibles correspondant est donné par :

$$M_{(W, \vec{K})} = \bigcap_{i=1}^n M_{(\vec{w}_i, k_i)}$$

En effet, les contraintes de type *CGEM* permettent de présenter l'exclusion mutuelle de l'utilisation des ressources communes partagées entre différents processus. Dans le cas où les transitions sont contrôlables, la contrainte *CGEM* est satisfaite en ajoutant au réseau une place de contrôle ou *moniteur* définie comme suit :

Définition 1.16 :

Soit un RdP (N, M_0) avec sa matrice d'incidence C et une *CGEM* (\vec{w}, K) , un moniteur qui respecte cette contrainte est une place de contrôle à ajouter au réseau résultant et à l'ensemble des places P . Par conséquent, la matrice d'incidence résultante C' devient : $C' = \begin{pmatrix} C \\ -\vec{w}C \end{pmatrix}$. La place de contrôle (moniteur) est pure, et entièrement définie par C' . De plus, le marquage

initial du réseau résultant M'_0 sera donné par : $M'_0 = \begin{pmatrix} M_0 \\ k - \vec{w}^T \cdot M_0 \end{pmatrix}$ car le marquage initial doit respecter la contrainte (\vec{w}, K) .

Ainsi, cette place de contrôle à ajouter au réseau peut avoir des arcs sortants vers (respectivement entrants depuis) des transitions d'entrée (respectivement de sortie) de toute place p appropriée au support \vec{w} $|\vec{w}[p] > 0$; contrairement aux places qui ont des poids négatifs. En effet, les poids des arcs sont définis à partir des coefficients \vec{w} et de la matrice C . Il est donc clair que pour une contrainte $CGEM$, on associe un moniteur. Les contraintes redondantes (où équivalentes) peuvent toujours être identifiées en délimitant un même ensemble d'états admissibles.

Remarque 1.4: Une spécification d'états interdits ne peut être traduite par un ensemble de $CGEM$ car l'ensemble d'états admissibles (1.7) correspondant à une $CGEM$ est nécessairement convexe.

Néanmoins, par les RdP conservatifs et saufs, il existe toujours un ensemble de contraintes $CGEM$ analogue à une liste quelconque d'états interdits [GDS92].

Théorème 1.5 :

Soit un RdP conservatif et sauf (N, M_0) , alors pour tout ensemble d'états interdits MI , il existe un ensemble de $CGEM$ (W, \vec{K}) $| M \setminus MI = M_{(\vec{w}, K)}$.

Il existe une classe d'états qui ne sont pas interdits par les spécifications, mais à partir desquels on peut franchir des transitions incontrôlables. Alors, il faut impérativement les exclure de l'ensemble d'états admissibles. Cet ensemble d'états admissibles correspondant à une $CGEM$ est donné par :

$$M_{(\vec{w}, K)}^c = \{M \in M \mid (\forall \sigma \in T^{uc*}, M[\sigma > M']). \vec{w}^T \cdot M' \leq K\} \quad (1.8)$$

Avec $M_{(\vec{w}, K)}^c$, l'ensemble d'états admissibles quelconque.

En conséquence, pour un RdP général, le problème de contrainte $CGEM$ ne peut pas être résolu avec des transitions incontrôlables. Cependant, en utilisant le théorème (1.5), ce problème peut être résolu avec un RdP sauf et conservatif par des moniteurs en présence des transitions incontrôlables [GDS92][GDS93].

c- L'approche de Ghaffari et al.

La méthode de Ghaffari et al. utilise les réseaux de Petri bornés, ordinaires ou généralisés pour la modélisation des procédés et les PEI/PTEI comme type de spécifications. En effet, cette approche se compose de deux étapes : dans la première étape, on procède par un nettoyage du graphe de marquage, à savoir l'identification des instances de séparation d'évènements (définition 1.15), les états admissibles et interdits, tandis que l'objectif de la deuxième étape est l'utilisation de la théorie des régions (TR) pour la synthèse des places de contrôle. Cet ensemble de places constitue le contrôleur RdP qui assure le respect des spécifications et du comportement désiré.

Afin de spécifier le marquage admissible R_c d'un RdP donné (N, M_0) , qui respecte les spécifications de contrôle, les auteurs proposent l'algorithme 1.1 suivant [GRX02a] :

Algorithme 1.1.

1. Détermination de l'ensemble des marquages interdits MI et des transitions d'état interdits TI .
2. Génération du graphe partiel d'atteignabilité $R_p(N, M_0)$
3. Identification de l'ensemble de transitions amenant à des états interdits D .i.e $D = \{M[t > M' \mid M \in R_p(N, M_0), M' \in MI]\}$; $TI = TI \cup D$.
4. Détermination de l'ensemble d'états dangereux MD , tel que $MD = \{M \in R(N, M_0) \mid \exists t \in T_{uc}, \exists \sigma \in T^* \wedge (M[\sigma > M'] \wedge (M'[t > M'']), M'' \in MI)\}$
5. Elimination des états appartenant à MD de $R_p(N, M_0)$
Soit R le graphe obtenu
6. Si R est co-atteignable, alors FIN. Sinon aller à 7.
7. Construction du plus grand sous-graphe co-atteignable $CA(R)$.
8. Détermination de l'ensemble d'états bloquants $MB = \{M \in CA(R) \mid \exists \sigma \in T^*, M[\sigma > M'], M' \notin CA(R)\}$
9. Elimination de R les états de MB , aller à 6.
FIN

Dans cet algorithme, les auteurs désignent initialement les ensembles d'états interdits et de transitions d'états interdites qui ne respectent pas une spécification de contrôle Q donnée. Ensuite, le graphe partiel d'atteignabilité est généré. En effet, dans ce graphe, les successeurs des états interdits ne sont plus générés. Après avoir déterminé l'ensemble TI , il faut chercher les transitions incontrôlables qui peuvent exister dans TI . C'est la raison pour laquelle les nœuds de MD sont identifiés dans un premier temps, et éliminés du graphe $R_p(N, M_0)$, par la suite. Le nettoyage du graphe concerne aussi l'élimination des états bloquants MB pour avoir enfin un graphe co-atteignable.

La synthèse de contrôle doit satisfaire le comportement du système contrôlé en inhibant toute transition interdite contrôlable. Les contraintes de vivacité et de contrôle doivent être satisfaites également, en interdisant tout marquage de sortie du graphe de marquage admissible de façon incontrôlable.

Dans la méthode de Ghaffari et al., les auteurs utilisent la théorie des régions pour la synthèse des contrôleurs RdP. En effet, à partir d'un modèle RdP initial représentant le procédé, on détermine les places de contrôle qui satisfont les spécifications de contrôle afin de les ajouter au modèle initial et avoir finalement le modèle RdP contrôlé.

Une place de contrôle pure notée p_c est caractérisée par son marquage initial $M_0(p_c)$ et son vecteur d'incidence $C_c(p_c, \cdot)$. Cette place doit satisfaire *les conditions d'atteignabilité*, c'est-à-dire que pour un système à superviser, chaque marquage $M \in R_c$ doit rester atteignable :

$$M_c(p_c) = M_0(p_c) + C_c(p_c, \cdot) \overrightarrow{\Gamma}_M \geq 0 \quad \forall M \in R_c \quad (1.9)$$

Γ_M est un chemin reliant M_0 à M .

De plus, p_c doit satisfaire les équations de cycle correspondant aux cycles existant dans R_c , on a alors :

$$\sum_{t \in T} C_c(p_c, t) \cdot \vec{\delta}[t] = 0, \forall \delta \in \Delta \quad (1.10)$$

$\vec{\delta}[t]$ et Δ sont respectivement la somme algébrique de toutes les occurrences de t dans δ et l'ensemble des cycles de base du graphe R_c .

Par ailleurs, la place de contrôle à synthétiser doit interdire au moins une transition d'état interdite appartenant à l'ensemble Ω d'instances de séparation d'évènements. Par conséquent, chaque nouvelle place p_c interdit une transition contrôlable t menant à un état interdit. Cette inhibition se fait par la condition de séparation suivante :

$$M'_c(p_c) = M_0(p_c) + C_c(p_c, \cdot) \overrightarrow{\Gamma}_M + C_c(p_c, t) < 0 \quad (1.11)$$

Ainsi, pour chaque instance de séparation d'évènements ou chaque transition contrôlable qui mène à un état interdit, on résout le système linéaire (SL) formé par les conditions (1.9), (1.10) et (1.11) pour calculer la place de contrôle p_c .

La synthèse de contrôle dans la méthode de Ghaffari et al. se base principalement sur l'utilisation de la théorie des régions pour déterminer des places de contrôle pures et impures. Pour plus d'exemples, le lecteur pourra voir [GRX02a][GRX02b][GRX03a][GRX03b].

d- L'approche de Huang et al.

La méthode de Huang et al. [HP11] propose une nouvelle méthodologie pour réduire le coût de calcul de la théorie des régions pour la synthèse de contrôle. En effet, cette méthode permet de réduire le nombre des instances de séparation d'évènements. Pour ce faire, les auteurs diminuent le nombre des conditions d'atteignabilité des états dans le système linéaire de la TR pour faire face aux problèmes d'évitement de blocage. L'avantage que présente cette méthode est la synthèse de contrôleurs RdP maximum permissifs avec une réduction de calcul considérable.

En comparant le travail de Huang et al. avec les méthodes existantes [UZ07][Uzam09][LZJ08], les résultats comparés dans [HP11] prouvent que leur politique de contrôle semble être la plus efficace pour les problèmes de blocage des RdP.

En effet, dans cette méthode, les auteurs estiment que pour chaque graphe de marquage, il existe au moins un état bloquant MB . L'ensemble de ces états sont regroupés dans une zone appelée *zone morte* Z_D . Une fois qu'un marquage entre dans Z_D , le système est dit *mort*. S'il n'existe aucune zone morte dans le graphe d'atteignabilité, le système est alors vivant. L'objectif de ce travail est donc de contrôler ces zones afin d'avoir un RdP vivant. Pour ce faire, les marquages de $G(N, M_0)$ doivent être divisés en catégories : des marquages légaux., marquages quasi-morts M_Q et marquages morts ou bloquants MB .

Définition 1.17 :

Une zone constituée de marquages quasi-morts $M_Q = \{M \in G(N, M_0), t \in T_c \mid (M[t > M'), M' \in MB\}$ est une zone quasi-morte notée Z_Q . Par conséquent, la zone légale $Z_L = G(N, M_0) - (Z_D + Z_Q)$ (voir l'exemple de la figure 1.8).

Dans la méthode de Huang et al., les auteurs procèdent d'abord par une approche de réduction des réseaux de Petri pour simplifier le modèle initial. Ainsi, la fusion des zones et l'élimination des conditions d'atteignabilité des états de Z_Q permettent de réduire le nombre d'équations du système linéaire de la théorie des régions, comme le montre l'exemple suivant de la figure 1.8 :

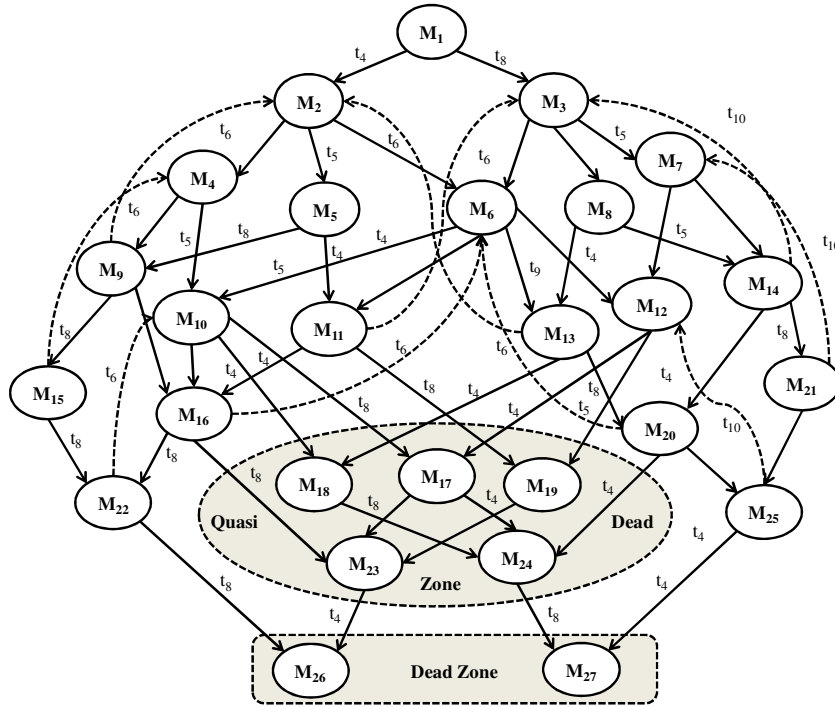


Figure 1.8- Graphe d'atteignabilité d'un RdP réduit [HP11]

Tous les marquages légaux peuvent être déterminés ; ainsi les marquages $M_1 - M_{11}$, $M_{20} - M_{22}$ et M_{25} représentent les marquages appartenant à ML . Les deux instances de séparation d'évènements sont :

$$M_{26}(p_c) = M_1(p_c) + 3C(p_c, t_4) + C(p_c, t_5) + 2C(p_c, t_8) < 0$$

$$M_{27}(p_c) = M_1(p_c) + 2C(p_c, t_4) + C(p_c, t_9) + 3C(p_c, t_8) < 0$$

De plus, il existe deux équations de cycle de base dans ce graphe d'atteignabilité .i.e dans la zone légale Z_L , ces deux équations sont :

$$C(p_c, t_4) + C(p_c, t_5) + C(p_c, t_6) = 0$$

$$C(p_c, t_8) + C(p_c, t_9) + C(p_c, t_{10}) = 0$$

L'application de la théorie des régions génère 4 places de contrôle dont une est redondante. Les contrôleurs RdP obtenus sont maximum permissifs. Cependant, on remarque la réduction

du coût de calcul de la TR en éliminant des équations d'atteignabilité du SL de la théorie des régions.

Dans la section suivante, nous introduisons des méthodes de dénombrement pour incorporer l'analyse combinatoire dans la synthèse de contrôleurs réseaux de Petri. En effet, l'analyse combinatoire nous permet de développer une approche de synthèse de commande qui calcule des superviseurs RdP sans générer le graphe de marquage $G(N, M_0)$.

1.7 Introduction à l'analyse combinatoire

Cette partie nous permettra de développer une nouvelle approche de synthèse de contrôleurs RdP sans calculer le graphe de marquage et en utilisant les méthodes d'analyse combinatoire.

L'analyse combinatoire représente une branche des mathématiques fournissant des méthodes de dénombrement particulièrement utiles en théorie des probabilités. Les différents modèles construits pour analyser les phénomènes hasardeux se basent sur la notion de probabilité qui exige des dénombrements d'ensembles finis.

En effet, toute suite d'éléments choisie parmi un ensemble fini peut être ordonnée ou pas, selon que l'on tient compte ou pas de la position occupée par ces éléments.

En résumé, l'analyse combinatoire peut se définir comme l'étude de différentes manières de ranger des objets en utilisant des méthodes de dénombrement [Riordan 12].

Dans cette partie, on présente les concepts de base de la théorie des combinaisons ainsi que quelques définitions utiles qui peuvent être utilisées dans cette thèse.

1.7.1 Définitions

a- Arrangement

Soit E un ensemble fini de n objets. On appelle *arrangement* de p objets, toute suite ordonnée de p objets pris parmi les n objets. Cependant, le nombre d'arrangement de p objets pris parmi n est noté A_n^p avec $1 \leq p \leq n$ et $n, p \in \mathbb{N}^*$.

En effet, lorsqu'un objet est observé plusieurs fois dans un arrangement, le nombre d'arrangements avec répétition de p objets parmi n est alors : $A_n^p = n^p$ avec $1 \leq p \leq n$.

Par contre, lorsque cet objet n'est observé qu'une seule fois dans un arrangement, le nombre d'arrangements sans répétition devient $A_n^p = \frac{n!}{(n-p)!}$ avec $1 \leq p \leq n$.

b- Permutations

Soit un ensemble E fini de n objets, on appelle permutations de n objets distincts, toute suite ordonnée de n objets ou tout arrangement n à n de ces objets.

Le nombre de permutations de n objets est noté $P(n, n) = n(n-1) \dots 1 = n!$

Les P permutations de n objets distincts peuvent être choisies en n façons, ou en $(n-1)$ et ainsi de suite jusqu'à $(n-p+1)$ façons. $P(n, p) = n(n-1) \dots (n-p+1)$; $n \geq p$.

La permutation de n objets présente un cas particulier d'arrangement sans répétition de p objets pris parmi n avec $p = n$. Par conséquent, le nombre de permutations de n objets est $A_n^n = \frac{n!}{(n-n)!} = n!$.

Effectivement, il existerait plusieurs répétitions k d'un même objet parmi les n objets, le nombre de permutations possibles des n objets doit être rapporté au nombre de permutations des k objets identiques. Le nombre de permutations de n objets est alors : $P_n = \frac{n!}{k!}$

En effet, les permutations de k objets similaires sont toutes équivalentes et ne comportent qu'une seule permutation.

1.7.2 Combinaisons

a- Combinaisons sans remise

On remarque qu'une combinaison donnée de p objets distincts peut être ordonnée en $p!$ façons, et si elle est ordonnée, c'est donc une p -permutation ; par conséquent, si $C(n, p)$ est le nombre en question, alors $p! C(n, p) = P(n, p) = n(n-1) \dots (n-p+1); n \geq p$.

Ainsi $C(n, p) = \frac{n(n-1)\dots(n-p+1)}{p!} = C_n^p = \frac{n!}{p!(n-p)!} = \binom{n}{p}$; $C(n, p) = \binom{n}{p}$ est le nombre de combinaisons de p objets parmi n et sans remise avec $1 \leq p \leq n$.

b- Combinaisons avec remise

Le nombre recherché pour les combinaisons avec remise est celui de p combinaisons parmi n objets choisis, dont chacune peut apparaître indéfiniment de 0 à p fois. Ceci est une fonction f de n et p notée $f(n, p)$.

Le nombre de combinaisons de p objets parmi n avec remise notée Γ_n^p est :

$$\Gamma_n^p = \frac{(n+p-1)!}{p!(n-1)!}$$

1.8 Conclusion

Dans ce chapitre, nous avons présenté les notions de base de la théorie de la supervision des systèmes à événements discrets, initiée par Ramadge et Wonham. Ces auteurs se sont basés sur les langages formels et les automates pour modéliser et synthétiser des superviseurs aptes à faire respecter par le système un certain comportement souhaité. En effet, ce superviseur est capable de tolérer ou pas des événements contrôlables et reste indifférent aux événements dits incontrôlables. Si le superviseur réalise un comportement désiré, alors il est dit maximum permissif.

Nous nous sommes focalisés sur les approches utilisant les réseaux de Petri (RdP), qui se sont révélés très avantageux par rapport aux autres outils de modélisation, grâce à leurs propriétés mathématiques et structurelles. Pour faire face aux problèmes de contrôle, la synthèse de contrôleurs RdP constitue une solution élégante et pratique pour les spécifications de type états interdits (PEI) et transitions d'états interdites (PTEI).

Des approches de synthèse de contrôleurs RdP ont été présentées afin de situer notre objectif parmi les travaux réalisés dans le domaine de synthèse de contrôle, notamment ceux qui utilisent la théorie des régions (TR) pour la synthèse des superviseurs RdP. En effet, dans la méthode de Ghaffari et al. [GRX02a][GRX02b][GRX03a][GRX03b], les auteurs partent d'un modèle RdP initial qui représente le procédé et applique la TR pour déterminer les places de contrôle à ajouter au RdP initial, afin de satisfaire les spécifications de contrôle souvent exprimées dans ce travail, sous forme de CGEM [Giua 96]. Par ailleurs, nous avons présenté la méthode de Huang et al. [HP11] qui s'intéresse à la réduction du coût de calcul de la théorie des régions, tout comme la présente thèse, pour synthétiser des contrôleurs RdP.

Enfin, une introduction à l'analyse combinatoire et à la théorie des combinaisons a été réalisée dans ce chapitre. Ces concepts nous seront utiles dans les prochains chapitres.

Chapitre 2

Synthèse de contrôleurs RdP basée sur les Coupes Minimales

***Résumé:** Dans ce deuxième chapitre, nous présentons une méthode de synthèse de contrôleurs réseaux de Petri. Elle est basée sur les coupes minimales et la théorie des régions. Cette théorie a été nouvellement interprétée et utilisée dans la synthèse des contrôleurs RdP maximum permissifs, s'ils existent. Cependant, comme toute méthode de synthèse de commande basée sur les graphes de marquage, la théorie des régions souffre du problème d'explosion combinatoire des états. De plus, cette théorie peut ne pas toujours aboutir à une solution. Dans ce travail, nous exposons une méthode efficace qui réduit le coût de calcul de la théorie des régions. En effet, la méthode proposée minimise le nombre d'équations du système linéaire de cette théorie et, par conséquent, réduit le temps de calcul des contrôleurs RdP. Deux approches différentes sont présentées dans ce chapitre pour l'identification des coupes minimales dans le graphe de marquage et la synthèse des superviseurs RdP. Chaque contrôleur synthétisé aura la tâche de superviser sa coupe. Grâce à un switch entre les places de contrôle, la coordination entre les contrôleurs RdP est faite par l'intermédiaire d'une fonction booléenne de contrôle. Les approches développées sont appliquées à un système de production flexible (FMS) avec des événements non contrôlables. Les résultats numériques obtenus sont par la suite comparés à d'autres travaux qui utilisent la théorie des régions dans la synthèse de contrôle. Les contributions de ce chapitre ont fait l'objet des publications suivantes : [RARK14], [RAR14] et [RARK15].*

2.1 Introduction

Dans ce chapitre, nous présentons une méthode de synthèse de contrôleurs RdP. Les problèmes que nous considérons incluent les problèmes d'états interdits (PEI) et les problèmes de transitions d'états interdites (PTEI). Dans un PEI, une transition d'état est inhibée si elle génère un état interdit. Dans un PTEI, une transition d'état est interdite par les spécifications de contrôle sans que l'état destinataire ne soit interdit.

La politique de contrôle proposée dans ce chapitre est valable pour tous les modèles réseaux de Petri généralisés bornés. Nous rappelons que l'ensemble des transitions T est constitué de deux sous-ensembles ; les transitions contrôlables T_c et les transitions incontrôlables T_{uc} ; $T = T_c \cup T_{uc}$.

La synthèse de contrôle est basée sur la théorie des régions (TR). En effet, cette théorie a été développée dans le domaine des sciences de l'informatique par Ehrenfeucht et Rosenberg. Une nouvelle interprétation de la TR a été proposée par Ghaffari et al. pour des problèmes de synthèse de commande. Théoriquement, la TR est représentée par un système linéaire (SL) composé de trois types de conditions et sa résolution conduit à la synthèse d'un superviseur RdP. Cependant, un état peut être une combinaison convexe d'autres états du graphe et le SL devient impossible à résoudre. De plus, l'explosion combinatoire des états dans le graphe généré alourdit fortement le calcul des contrôleurs RdP.

Dans cette partie, l'application de la théorie des régions est restreinte aux coupes minimales CM et non pas sur la totalité du graphe. Ces CM sont identifiées dans le graphe de marquage $G(N, M_0)$ avec des critères spécifiques. Chaque contrôleur synthétisé supervise une coupe minimale dans le graphe de marquage. Par la suite, il existe une coordination entre tous les contrôleurs RdP qui se fait grâce à une fonction booléenne de contrôle.

Dans ce qui suit, nous présentons la théorie des régions et ses limites, dans certains cas, avec l'approche classique dans la synthèse de contrôle de Ghaffari et al. Ensuite, nous définissons le nouveau concept de coupes minimales dans un graphe de marquages avec deux approches différentes proposées pour l'identification des CM . Des algorithmes traduisant notre politique de contrôle vont être présentés par la suite. La section 2.4 va définir la fonction booléenne de contrôle qui gère le switch et la coordination entre les contrôleurs RdP. Enfin, nous clôturons ce chapitre par l'application de notre méthode sur un système de production flexible (FMS) et d'autres exemples qui nous serviront à comparer nos approches avec celles de Ghaffari et al. et de Huang et al.

2.2 Théorie des régions pour la synthèse de contrôle

Dans cette section, nous exposons le problème de synthèse de contrôleur RdP de type $PTEI$ et PEI . Cette méthode de synthèse est fondée sur la théorie des régions. Cette théorie a été adaptée pour la synthèse de contrôle par Ghaffari et al. [GRX02a][GRX02b].

2.2.1 Synthèse de places de contrôle pures

L'application de la théorie des régions permet de synthétiser des contrôleurs RdP à ajouter au modèle réseaux de Petri initial. Une place de contrôle pure p_c assure un comportement souhaité qui satisfait une spécification de contrôle. Son marquage initial est noté $M_0(p_c)$ et son vecteur d'incidence, qui lie p_c à toutes les transitions du modèle RdP initial, est ainsi noté $C(p_c, \cdot)$.

Définition 2.1 :

Un comportement admissible/désiré R correspondant à un problème de type *PTEI* ou *PEI* vérifie la propriété suivante : $\forall t \in T$, si $M \in R$ et $M[t > M'$ alors $M \xrightarrow{t} M' \in R$.

Les problèmes qu'on peut rencontrer dans les systèmes de production manufacturiers sont tous des problèmes de type *PTEI*. L'exemple de la figure 2.1 illustre le choix d'un comportement admissible d'un RdP donné. En effet, à partir de l'état initial, les deux transitions t_1 et t_2 sont contrôlables et validées, menant à un même état M_1 . Ces deux transitions peuvent aussi représenter une même opération qui peut être effectuée avec l'une des deux machines identiques. Pour ces raisons d'équilibrage des charges, ces machines doivent être utilisées alternativement. Supposons que la priorité est donnée à t_1 à partir de M_0 . Le comportement non contrôlé du modèle RdP est donné par son graphe d'atteignabilité (Figure 2.1.b) Les événements interdits sont $(M_0[t_2 > M_1])$ et $(M_1[t_1 > M_2])$. La figure 2.2 montre le comportement admissible R du modèle contrôlé.

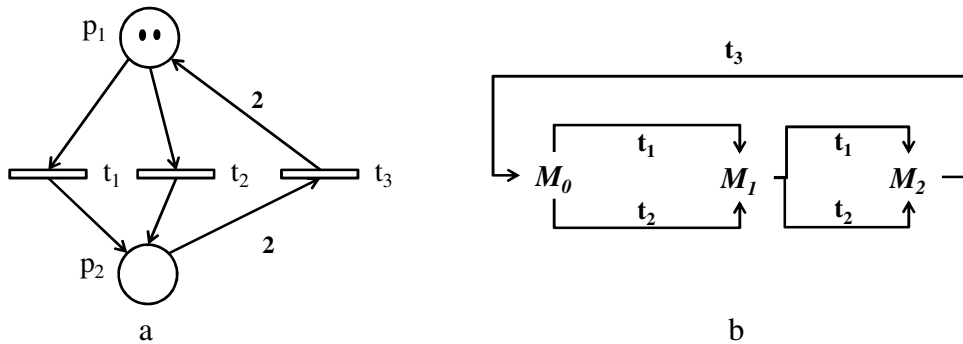


Figure 2.1- Un exemple de PTEI

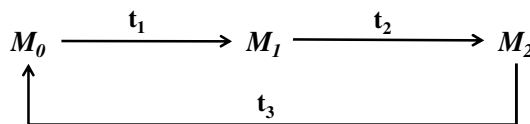


Figure 2.2- Comportement désiré du RdP de la figure 2.1

L'ensemble de ces places de contrôle synthétisées forme les contrôleurs RdP du modèle contrôlé, qui assureront à leur tour un comportement désiré à satisfaire.

Chaque place de contrôle p_c doit satisfaire trois conditions pour assurer un comportement admissible, notamment les *conditions d'atteignabilité* :

$$M_0(p_c) + C(p_c, \cdot) \overline{\Gamma}_M \geq 0 \tag{2.1}$$

En outre, p_c doit résoudre au moins une *condition de séparation d'évènements* (M, t) de l'ensemble Ω des instances de séparation d'évènements :

$$M_0(p_c) + C(p_c, \cdot) \overline{\Gamma}_M + C(p_c, t) < 0 \quad (2.2)$$

Finalement, cette place de contrôle doit vérifier les *équations de cycle* :

$$\sum_{t \in T} C(p_c, t) \cdot \vec{\sigma}[t] = 0 \quad \forall \sigma \in R \quad (2.3)$$

La figure 2.3 présente un exemple d'application de la théorie des régions (TR) pour un problème PEI de blocage. En effet, on va déterminer le contrôleur nécessaire à ajouter au modèle initial pour éviter le blocage du réseau de Petri et assurer un comportement vivant. Le marquage bloquant à interdire est $M_7 = [1 \ 2 \ 0 \ 0 \ 0 \ 0]^T$.

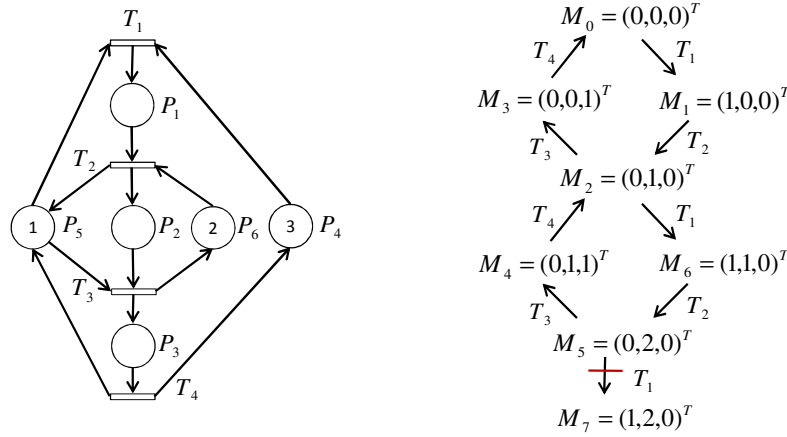


Figure 2.3- Modèle RdP initial et son graphe d'atteignabilité

On a une seule condition de séparation d'évènements à interdire dans cet exemple :

$$M_7(p_c) = M_0(p_c) + 2 \cdot C(p_c, t_1) + 2 \cdot C(p_c, t_2) + C(p_c, t_1) < 0$$

Le comportement souhaité est défini par les états $M_0 - M_6$ dont les conditions d'atteignabilité sont :

$$\begin{aligned} M_0(p_c) &\geq 0 \\ M_1(p_c) &= M_0(p_c) + C(p_c, t_1) \geq 0 \\ M_2(p_c) &= M_0(p_c) + C(p_c, t_1) + C(p_c, t_2) \geq 0 \\ M_3(p_c) &= M_0(p_c) + C(p_c, t_1) + C(p_c, t_2) + C(p_c, t_3) \geq 0 \\ M_4(p_c) &= M_0(p_c) + C(p_c, t_1) + 2C(p_c, t_2) \geq 0 \\ M_5(p_c) &= M_0(p_c) + 2C(p_c, t_1) + 2C(p_c, t_2) \geq 0 \\ M_6(p_c) &= M_0(p_c) + 2 \cdot C(p_c, t_1) + 2 \cdot C(p_c, t_2) + C(p_c, t_3) \geq 0 \end{aligned}$$

Par ailleurs, le graphe d'atteignabilité (Figure 2.3) contient deux cycles qui ont la même équation suivante :

$$C(p_c, t_1) + C(p_c, t_2) + C(p_c, t_3) + C(p_c, t_4) = 0$$

La résolution du système linéaire constitué par ces trois types de conditions nous génère la place de contrôle suivante :

$$p_c: \begin{cases} M_0(p_c) = 2 \\ C(p_c, \cdot) = (-1, 0, 0, 1) \end{cases}$$

La figure 2.4 montre le modèle RdP contrôlé par l'ajout du contrôleur RdP.

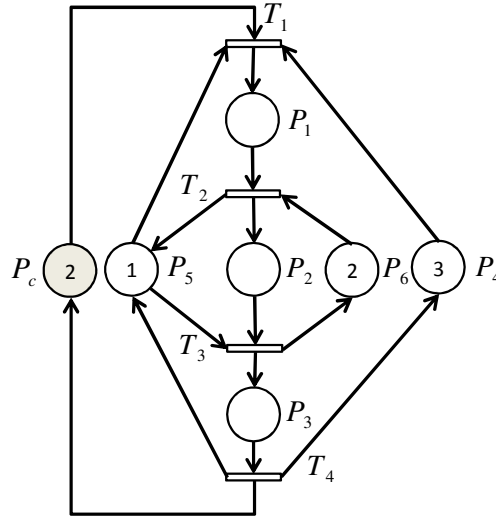


Figure 2.4- Modèle RdP contrôlé

Le nombre de places de contrôle supplémentaires qui peuvent exister est au plus égal à $card(\Omega)$. En pratique, dans l'approche de [GRX02a], le nombre de places de contrôle est beaucoup plus petit que $card(\Omega)$.

Néanmoins, l'application de la théorie des régions est difficile pour un graphe de marquage $G(N, M_0)$ considérable. En effet, le coût de calcul des places de contrôle est énorme dans le cas où on a une explosion combinatoire des états. De plus, le système linéaire de la TR peut contenir des combinaisons convexes, c.-à-d. qu'on peut trouver un marquage qui est une combinaison convexe d'autres marquages, rendant le système linéaire insoluble.

Nous présentons dans la figure 2.5 un exemple, où l'application de la théorie des régions dans la synthèse de contrôle ne donne pas de solution.

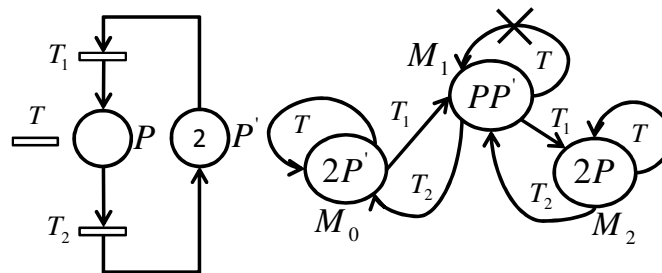


Figure 2.5- Problème insoluble par la théorie des régions

Cet exemple présente un problème de type PTEI. En appliquant la théorie des régions, il n'existe pas une place de contrôle pour interdire la transition d'état ($M_1[T > M_1]$) et réaliser le comportement souhaité.

En écrivant le système linéaire suivant (SL), la théorie des régions est insoluble et la synthèse d'un contrôleur RdP est impossible :

$$\begin{aligned}
 M_0(p_c) &= M_0(p_c) \geq 0 \\
 M_1(p_c) &= M_0(p_c) + C(p_c, T_1) \geq 0 \\
 M_2(p_c) &= M_0(p_c) + 2C(p_c, T_1) \geq 0 \\
 M_0(p_c) + C(p_c, T_1) + C(p_c, T) &< 0 \\
 C(p_c, T) &= 0 \\
 C(p_c, T_1) + C(p_c, T_2) &= 0
 \end{aligned}$$

En effet, il est clair que dans SL , le marquage $M_1(p_c)$ forme une combinaison convexe des marquages $M_0(p_c)$ et $M_2(p_c)$ c.à.d $M_1(p_c) = \frac{1}{2}M_0(p_c) + \frac{1}{2}M_2(p_c)$. Par conséquent, l'évènement (M_1, t) ne peut pas être interdit avec l'application de TR dans la synthèse de contrôle.

Dans la section suivante, on va introduire une méthode efficace afin réduire le nombre total d'équations dans SL . Pour plus de commodité, la méthodologie proposée suivra l'interprétation de la théorie des régions étudiée dans la section précédente.

2.3 Synthèse de contrôle basée sur les coupes minimales

Ce travail est axé sur la réduction du coût de calcul de la théorie des régions en réduisant au maximum le nombre d'équations dans les systèmes linéaires. Par conséquent, on peut nettement gagner le temps de calcul de la TR .

Afin de réaliser cet objectif, un nouveau concept de Coupe Minimale (CM) est introduit dans le graphe de marquage en utilisant deux approches.

Définition 2.2 :

Un état interdit est un état qui ne respecte pas une spécification de contrôle donné, exprimée sous forme de $CGEM(\vec{w}, K)$ (voir figure 2.6.a). L'ensemble des états interdits est noté $MI_i = \{M \in M \mid M \neq M_{(\vec{w}, K)}\}$.

Définition 2.3 :

L'ensemble des instances de séparation d'évènements que le superviseur doit interdire est $\Omega = \{(M, t) \mid \exists (M[t > M']) \wedge M \in G(N, M_0) \wedge M' \in MI\}$ (voir figure 2.6.c)

Définition 2.4 :

Un état est dit dangereux s'il mène à un état MI en franchissant une séquence de transitions incontrôlables. Soit $MD_i = \{M \in G(N, M_0) \mid \exists t \in T_{uc}, \exists \sigma \in T_{uc}^*, (M[\sigma > M']) \wedge (M'[t > M'']) \wedge (M'', t) \in \Omega\}$ (voir figure 2.6.a).

Définition 2.5 :

Un *état bloquant* est un état qui n'appartient pas au sous-graphe vivant (*LSG*). *LSG* est un sous-graphe du graphe de marquage $G(N, M_0)$ qui contient tous les états $M \in G(N, M_0)$ menant à l'état initial M_0 . Soit *MB* l'ensemble des états bloquants :

$$MB_i = \{M \in G(N, M_0) \mid \nexists \sigma \in T^*, (M[\sigma > M_0])\} \text{ (voir figure 2.6.b)}$$

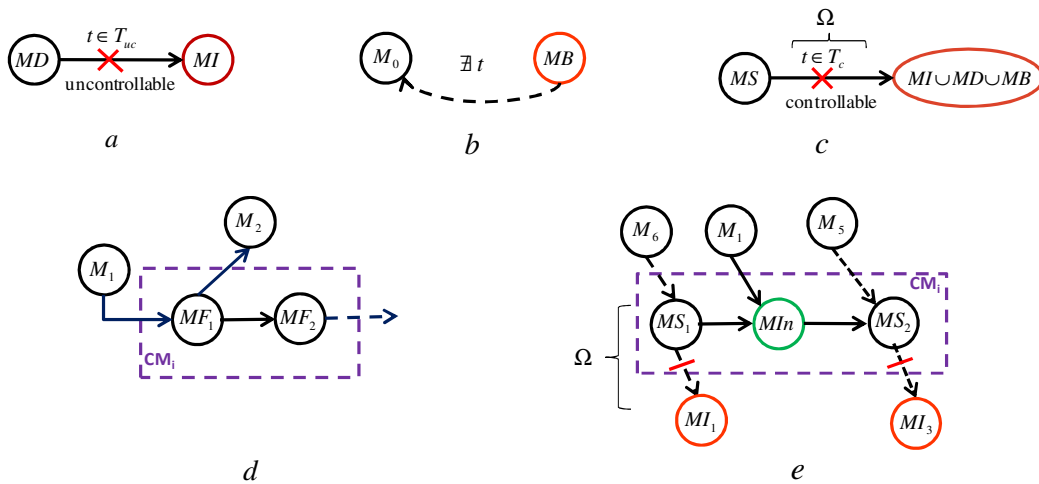
Définition 2.6 :

Un *état source* est un état admissible dans une coupe minimale CM_i , à partir duquel on interdit une transition appartenant à Ω . Soit MS_i l'ensemble des états sources (voir figure 2.6.c), tel que : $MS_i = \{M \in G(N, M_0) \mid \exists (M[t > M']), (M, t) \in \Omega, M' \in (MI_i \cup MD_i \cup MB_i), t \in T_c\}$

Définition 2.7 :

Un *état frontière* MF_i est un état qui appartient au graphe de marquages $G(N, M_0)$, à partir duquel on entre/sort dans la coupe minimale CM_i . (voir figure 2.6.d).

$$MF_i = \{M \in CM_i \mid \exists (M_i[t > M]) M_i \notin CM_i\}$$



- a. Etat dangereux et état interdit
- b. Etat bloquant
- c. Etat source
- d. Etat frontière
- e. Etat intermédiaire

Figure 2.6- Six types d'états

Nous proposons ci-après les deux approches d'identification des coupes minimales CM dans un graphe de marquage pour la synthèse des contrôleurs RdP, en utilisant les définitions de la section précédente.

2.3.1 Approche 1: Coupe minimale à instances de séparation d'évènements corrélées

L'identification d'une coupe minimale CM_i dans un graphe de marquage $G(N, M_0)$ est caractérisée par la corrélation des instances de séparation d'évènements. La coupe minimale choisie garantit un comportement maximum permissif et non bloquant. Cependant,

l'application de la théorie des régions dans la synthèse de contrôle se restreint dans CM_i et non pas sur tout le graphe et génère un seul contrôleur P_{ci} pour chaque CM_i .

Définition 2.8 :

Une coupe minimale CM_i est une zone du graphe de marquage $G(N, M_0)$ avec un nombre minimal de nœuds, qui s'identifie par une place de contrôle P_{ci} , garantissant l'atteignabilité de tous les états contenus dans cette zone, et qui permet de résoudre une ou plusieurs instances de séparation d'évènements.

$$CM_i = MS_i \cup \{(M[\sigma > M'] \mid M, M' \in MS_i, \sigma \in T^*\}$$

En d'autres termes, la coupe minimale regroupe tous les états sources MS_i et les chemins les reliant. La figure 2.7(a) et 2.7(b) montre une coupe minimale identifiée dans un graphe de marquage.

Définition 2.9 :

Deux instances de séparation d'évènements sont corrélées si leurs transitions d'état interdites correspondantes sont les mêmes. Soit $\Omega_{11} = (M, t_i)$ et $\Omega_{12} = (M', t_j) \in \Omega$, si $t_i = t_j$ alors Ω_{11} et Ω_{12} sont corrélées (voir Figure 2.7.c).

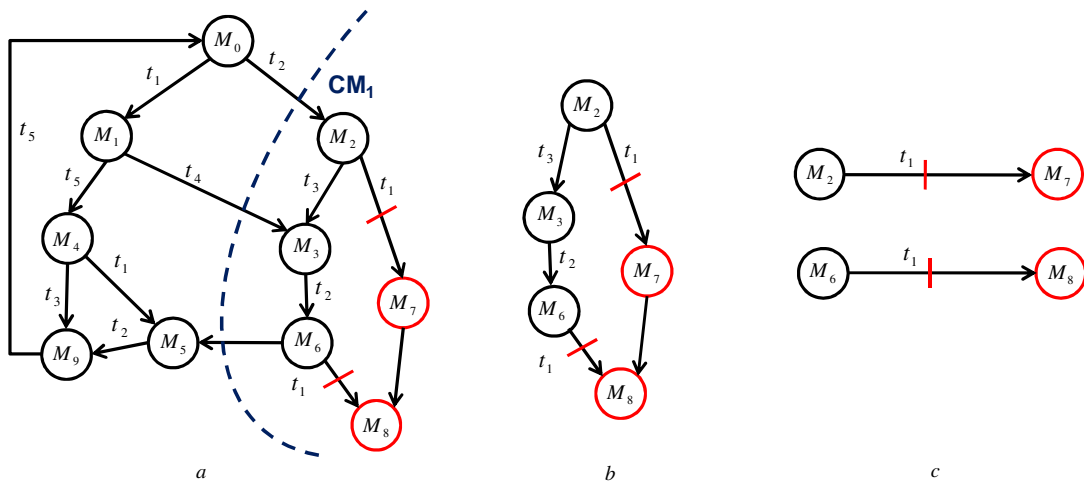


Figure 2.7- Un exemple de coupe minimale à instances de séparation d'évènements corrélées avec t_1

Néanmoins, dans une coupe minimale, on peut trouver des états intermédiaires qui peuvent exister dans le chemin qui relie deux états sources. L'ensemble de ces états est noté Min_i .

Définition 2.10 :

Un marquage intermédiaire est un état frontière appartenant à une coupe minimale CM_i et qui n'est pas un état source : $Min_i = \{M \in CM_i \mid M \in MF_i/MS_i\}$ (voir figure 2.6.e où l'état intermédiaire est dessiné en vert).

Proposition 2.1 :

L'état initial M_{0i} d'une coupe minimale CM_i appartient à l'ensemble des états sources MS_i . Il est généré par le plus petit vecteur de transition x dans le graphe de marquage $G(N, M_0)$ à partir de M_0 .

Preuve :

Soit x et x' deux vecteurs de transition des états M et M' ($M, M' \in G(N, M_0)$) et soit C la matrice d'incidence du réseau de Petri correspondant. $\vec{\sigma}$ et $\vec{\sigma}'$ sont les deux séquences de franchissement associées à M et M' , telles que : $M = M_0 + C \cdot \vec{\sigma}$ et $M' = M_0 + C \cdot \vec{\sigma}'$. A partir de [DX03], si $\sigma' \geq \sigma$, il existe une séquence de franchissement de transition : $\sigma'' = \sigma' - \sigma \mid M[\sigma'' > M'$, alors x est plus petit que x' .

Définition 2.11 [DX03] :

Un vecteur de transition x d'un état $M \in G(N, M_0)$ est le plus petit vecteur de transition si son état correspondant M est calculé avec la plus petite séquence de franchissement σ à partir de M_0 (i.e. $i = |\sigma|$ appelé la longueur d'une séquence, c 'est le nombre de transitions dans σ).

L'identification des coupes minimales dans un graphe de marquage $G(N, M_0)$ est décrite dans l'algorithme 2.1 suivant. Nous générons ce graphe de marquage en utilisant un logiciel de simulation et d'analyse des réseaux de Petri colorés *Design/CPN* [DES].

Algorithme 2.1 : Identification des coupes minimales dans $G(N, M_0)$.

Soit Q une spécification de contrôle donnée et $MA = MI \cup MD \cup MB$ l'ensemble des états à éviter.

- 1 Déterminer les ensembles des états et de transitions d'états, respectivement MI et T_F qui ne respectent pas Q .
- 2 Spécifier les instances de séparation d'évènements : $t_f = \{(M[t > M'), M \in G(N, M_0), M' \in MI\}$; $T_F = T_F \cup t_f$.
- 3 Identifier l'ensemble des états dangereux MD dans $G(N, M_0)$.
- 4 Identifier les transitions contrôlables qui mènent à MD .
- 5 Déterminer l'ensemble des états bloquants MB .
- 6 Identifier les transitions contrôlables qui mènent à MB .
- 7 Spécifier les instances de séparation d'évènements $\Omega = \{(M[t > M'), M \in G(N, M_0), t \in T_c, M' \in MA\}$
- 8 Définir les états sources $MS = \{(M \in G(N, M_0) \mid \exists (M, t) \in \Omega, t \in T_c\}$
- 9 Si $M \notin (MS \cap MB)$, alors aller à l'étape suivante
Sinon éliminer $(M[t >)$ de l'ensemble Ω et aller à l'étape 5
- 10 Déterminer chaque ensemble Ω_i d'instances de séparation d'évènements corrélées
- 11 Regrouper les états sources MS_i de chaque ensemble Ω_i et les chemins les reliant dans une coupe minimale CM_i .
- 12 Définir l'état initial M_{0i} de la coupe CM_i .

FIN

MI , ainsi que T_F désignent initialement l'ensemble des états interdits (respectivement des transitions d'états interdites) qui ne respectent pas Q .

A l'étape 2, on ajoute à T_F toute transition d'état qui génère un état appartenant à l'ensemble des états interdits MI . Maintenant, l'ensemble de toutes les transitions d'état interdites du

modèle est dans T_F . Si ces transitions sont occasionnées par le franchissement de transitions incontrôlables, alors il faut détecter en amont les états dangereux MD . Ensuite, nous identifions les transitions qui mènent aux marquages MD . De même pour les états bloquants MB ; une fois identifiés, on détermine les transitions contrôlables menant à MB . Une fois que les états à éviter sont identifiés $MA = MI \cup MD \cup MB$, nous déterminons l'ensemble des instances de séparation d'évènements Ω . Plus formellement, cet ensemble est donné par : $\Omega = \{(M[t > M'], M \in G(N, M_0), t \in T_c, M' \in MA)\}$.

A l'étape 8, nous définissons tous les états sources MS à partir de l'ensemble Ω . Si chaque état dans MS n'est pas bloquant alors on cherche les instances de séparation d'évènements corrélées et on les regroupe dans une même coupe minimale CM_i . Sinon on revient à l'étape 5 d'identification de MB et on refait le même processus.

Finalement, après avoir identifié la coupe minimale CM_i , on lui définit son état initial M_{0i} . Consécutivement, après un nombre nécessaire d'itérations, on obtient toutes les coupes minimales dans le graphe de marquage $G(N, M_0)$.

Remarque 2.1 :

L'application de la théorie des régions dans la méthode de Ghaffari et al. [GRX02a] dans la synthèse de contrôle, considère la totalité du graphe $G(N, M_0)$. Cependant, dans notre présente méthode, l'union des coupes minimales ne forme pas la totalité du graphe .i.e $\cup CM_i \subset G(N, M_0)$. Ce qui approuve la réduction de la complexité de calcul de la théorie des régions.

Théorème 2.1 :

Si un marquage entre dans CM_i par un état source $M \in MS_i \mid M \neq M_{0i}$, le marquage initial de P_{ci} n'a pas besoin d'être recalculé et la théorie des régions reste valide.

Preuve (démonstration par l'absurde) :

Soit CM_i une coupe minimale dans le graphe de marquage $G(N, M_0)$ corrélée par une transition t_1 (voir figure 2.8) et MI_i l'ensemble des états interdits correspondant, $MI_i = (M', M'')$. Hypothèse : Si un marquage entre dans une coupe CM_i par un état $M \in MS_i \mid M \neq M_{0i}$ tel que $(M_{0i}[t_1 > M'])$ et $(M[t_1 > M''])$ alors avec l'application de la théorie des régions calculée auparavant à partir de M_{0i} , la transition t_1 ne peut pas être interdite si un marquage entre dans CM_i par M et on écrit $M''(P_{ci}) = M(P_{ci}) + C(P_{ci}, t_1) \geq 0$ et la théorie des régions n'est pas valide. Supposons qu'un marquage M_k entre dans CM_i en franchissant t_1 (figure 2.8) tel que $M_k(P_{ci}) = M_0(P_{ci}) + 3 \cdot C(P_{ci}, t_2)$, $M = M_k + C(P_{ci}, t_1) = M_0(P_{ci}) + 3C(P_{ci}, t_2) + C(P_{ci}, t_1)$. Puisque $M''(P_{ci}) = M(P_{ci}) + C(P_{ci}, t_1) = M_0(P_{ci}) + 2C(P_{ci}, t_1) + 3C(P_{ci}, t_2) \geq 0$ or M_{0i} est l'état calculé avec le plus petit vecteur, donc M'' peut être exprimé par M_{0i} de CM_i . A partir de la figure 2.8, $M_{0i}(P_{ci}) = M_0(P_{ci}) + C(P_{ci}, t_1) + 2C(P_{ci}, t_2)$; $M''(P_{ci}) = M_{0i}(P_{ci}) + C(P_{ci}, t_1) + C(P_{ci}, t_2) \geq 0$ Ce qui est *absurde*. En effet, l'application de la théorie des régions, à partir de l'état initial de la coupe M_{0i} , donne $M''(P_{ci}) = M_{0i}(P_{ci}) + C(P_{ci}, t_1) + C(P_{ci}, t_2) < 0$. Enfin, si un marquage entre dans une coupe minimale CM_i par n'importe quel

état $M \in MS_i$, le marquage initial de la place de contrôle $M_{0i}(P_{ci})$ n'a pas besoin d'être recalculé et la théorie des régions reste valide.

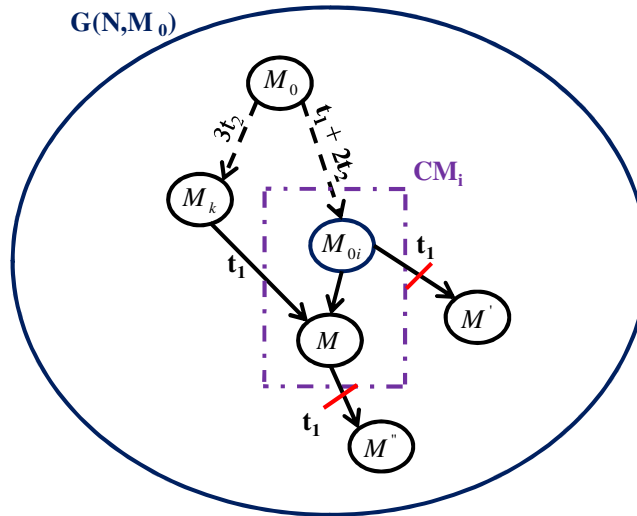


Figure 2.8- Un exemple de coupe minimale

Une fois que la coupe minimale CM_i est déterminée, les relations (2.1), (2.2) et (2.3) rendent possible la détermination de la place de contrôle correspondante P_{ci} ; l'algorithme 2.2 suivant résume le calcul du superviseur RdP.

Algorithme 2.2 : Synthèse de P_{ci} pour chaque CM_i

Soit M_{0i} l'état initial de CM_i , avec $n_i = \text{card}(\Omega)$.

- 1 Initialisation $i := 1$
 - 2 Si $n_i = 1$
 - 2.1 Ecrire l'équation d'atteignabilité de M_{0i}
 - 2.2 Ecrire les conditions de séparation d'évènements de $(M_{0i}[t > M']) \in \Omega_i$
 - 2.3 Résoudre le système linéaire SL_i et soit $[M_{0i}(P_{ci}), C(P_{ci}, \cdot)]$ la solution de CM_i .
 - 3 Sinon si $n_i > 1$
 - 3.1 Identifier les équations de cycle de base en utilisant la méthode de Gauss
 - 3.2 Ecrire les conditions d'atteignabilité des états $M \in CM_i$ à partir de M_{0i}
 - 3.3 Ecrire les conditions de séparation d'évènements de l'ensemble Ω_i à partir de M_{0i} .
 - 3.4 Résoudre le système linéaire de la théorie des régions et soit $[M_{0i}(P_{ci}), C(P_{ci}, \cdot)]$ la solution de CM_i
 - 4 Tant que $i < n, i := i + 1$. Retourner à l'étape 2
 - 5 Eliminer les places de contrôles redondantes
- FIN

Notre approche est programmée avec un langage C et a été réalisée avec l'aide d'un élève ingénieur lors de son projet de fin d'études. Le programme récupère en entrée le réseau de Petri et la spécification de contrôle exprimée sous forme de CGEM. Il nous retourne en sortie les coupes minimales et les contrôleurs RdP correspondants. Afin de résoudre les systèmes linéaires de la théorie des régions, le programme fait appel aux fonctions de la bibliothèque du logiciel de programmation mathématique CPLEX 7.1. L'application de notre méthode sur des

exemples traités à la fin de ce chapitre permet d'avoir une idée sur la performance de cet algorithme.

Après avoir défini la première approche d'identification des coupes minimales dans un graphe de marquage, on attaque dans la section suivante une deuxième approche de sélection des coupes minimales CM_i dans $G(N, M_0)$.

2.3.2 Approche 2: Coupe minimale à instances de séparation d'évènements quelconques

Dans cette approche, la conception d'une coupe minimale CM_i ne s'intéresse qu'aux transitions d'état interdites qui ne respectent pas une spécification de contrôle Q donnée. La coupe minimale identifiée garantit un comportement maximum permissif et non bloquant. L'application de la théorie des régions se restreint aussi à CM_i générant une seule place de contrôle P_{ci} .

Définition 2.12 :

Une coupe minimale CM_i est une zone du graphe de marquage $G(N, M_0)$ s'identifiant par une seule place de contrôle P_{ci} et ne s'intéresse qu'aux instances de séparation d'évènements.

$$CM_i = \{(M[t > M'] \mid M \in MS, M' \in MA)\}$$

La figure 2.9(a), 2.9(b) et 2.9(c) nous montre l'identification d'une coupe minimale dans un graphe de marquage, choisie selon les instances de séparation d'évènements. Quelle que soit la transition qui constitue l'évènement à interdire, les instances de séparation adjacentes forment la coupe minimale.

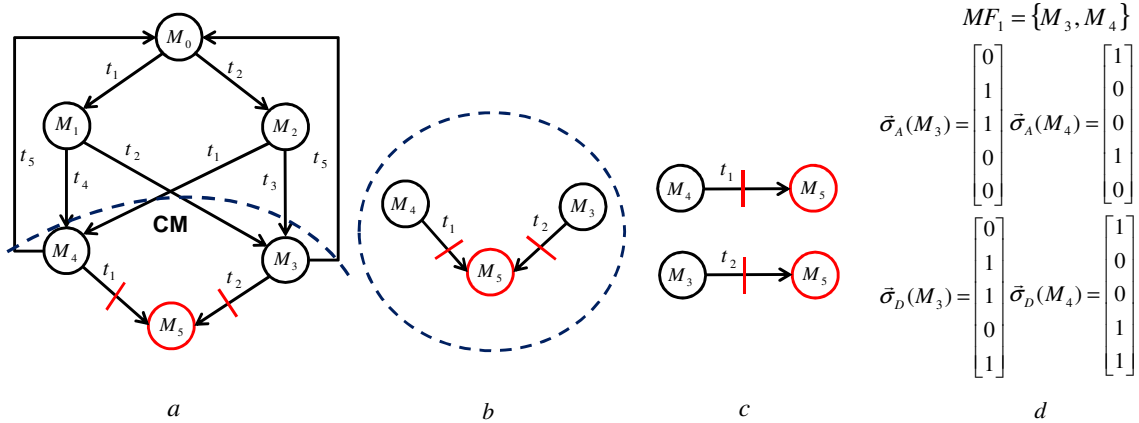


Figure 2.9- Identification d'une CM en utilisant l'approche 2

La conception des coupes minimales CM_i avec instances de séparation d'évènements quelconques est décrite dans l'algorithme 2.3 suivant. En effet, les étapes (1-9), qui déterminent, d'une part, les états interdits, bloquants et dangereux et, d'autre part, les transitions d'état interdites sont les mêmes que dans l'algorithme 2.1 ; la différence est la façon avec laquelle on construit la coupe définie dans les étapes 10, 11 et 12.

Algorithme 2.3 : Identification des CM_i dans $G(N, M_0)$.

Soit $T_\alpha(M)$ l'ensemble des arcs d'un état source $M \in MS_i$

- 10 Déterminer pour chaque état $M \in MS_i$, les arcs

$$T_\alpha(M) = \{t \in T \mid (M[t > M''), M \in MS_i, M'' \notin MA\}$$
 - 11 Associer à chaque état $M \in MS_i$ et ses instances de séparation d'évènements $((M[t > M'], M \in MS_i, M' \in MA)$ une coupe minimale CM_i
 - 12 Si $M' \in MA \mid (M[t > M'], M \in MS_i)$ est un état commun entre deux CM_i et $t \in T \mid (M[t > M']) \notin T_\alpha(M)$, alors fusionner les coupes minimales
- FIN

Une fois que les états à éviter (MA) et les transitions d'état interdites sont identifiés, on détermine pour chaque état source $M \in MS_i$, les arcs $T_\alpha(M)$. Cet ensemble représente les arcs sortant d'un état source qui ne mènent pas à un état interdit. Ensuite, dans l'étape 11, on associe pour chaque état source $M \in MS_i$ et ses instances de séparation d'évènements une coupe minimale. Si l'état à éviter est un état commun entre les coupes et que les transitions à interdire n'appartiennent pas à $T_\alpha(M)$ alors on fusionne les coupes minimales en une seule CM_i .

Remarque 2.2 :

La fusion des coupes minimales dans l'approche 2 peut nettement réduire le nombre des CM_i et par conséquent, le nombre de contrôleurs RdP sera diminué.

Après avoir déterminé toutes les coupes minimales dans le graphe de marquage $G(N, M_0)$, l'application de la théorie des régions se restreint aux zones CM_i pour synthétiser les contrôleurs correspondants. L'algorithme 2.4 suivant illustre la synthèse de contrôle de l'approche 2.

Algorithme 2.4 : Synthèse de contrôleurs RdP pour les coupes minimales

Soit CM l'ensemble des coupes minimales, M_{0i} l'état initial de chaque CM_i et $n_i = \text{card}(\Omega_i)$.

- 1 Initialisation $i := 1$
 - 2 Si $n_i = 1$
 - 2.1 Ecrire l'équation d'atteignabilité de l'état initial M_{0i}
 - 2.2 Ecrire les conditions de séparation d'évènements $M_{0i}[t > M', M' \in MA]$.
 - 2.3 Résoudre le système linéaire SL_i et soit $[M_{0i}(P_{ci}), C(P_{ci}, \cdot)]$ la solution de CM_i .
 - 3 Sinon si $n_i > 1$
 - 3.1 Ecrire l'équation d'atteignabilité de l'état M_{0i} de CM_i
 - 3.3 Ecrire les conditions de séparation d'évènements de l'ensemble à partir de M_{0i} .
 - 3.4 Résoudre le système linéaire de la théorie des régions et soit $[M_{0i}(P_{ci}), C(P_{ci}, \cdot)]$ la solution de CM_i
 - 4 Tant que $i < n, i := i + 1$. Retourner à l'étape 2
 - 5 Eliminer les places de contrôles redondantes
- FIN

Reprenons l'exemple de la figure 2.5 où le problème est insoluble par l'application de l'approche classique de la théorie des régions dans la synthèse de contrôle. On applique alors notre politique de contrôle en utilisant la deuxième approche (voir figure 2.10(a)) :

$SL: M_{01}(P_{c1}) \geq 0, M_{01}(P_{c1}) + C(P_{c1}, T) < 0$. La résolution de SL nous donne $M_{01}(P_{c1}) = 1; C(P_{c1}, T) = -2$. Le système contrôlé est donné par la figure 2.10(b).

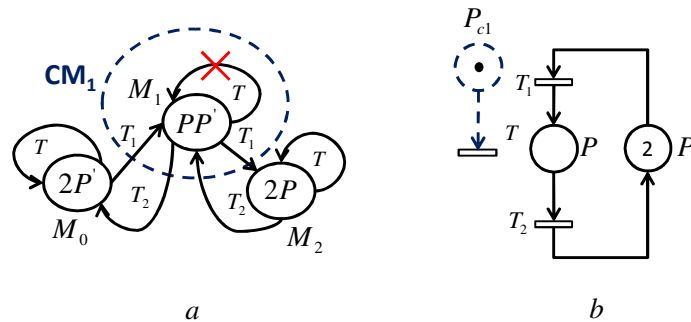


Figure 2.10- Résolution du problème insoluble de la théorie des régions

En dépit de la réduction du temps de calcul et du nombre d'équations, notre politique de contrôle permet de résoudre des problèmes insolubles avec l'approche classique de la théorie des régions. Il est également nécessaire de calculer les vecteurs d'activation/désactivation de l'ensemble des états frontières $MF_1 = \{M_1\}$ pour activer/désactiver le contrôleur P_{c1} . Une fois que les contrôleurs RdP sont synthétisés pour les coupes minimales CM , on doit déterminer la fonction booléenne de contrôle F_{BC} qui autorise l'activation et la désactivation des superviseurs. Ainsi, le rôle de chaque moniteur P_{ci} est de contrôler sa zone CM_i dans le graphe de marquage. Les vecteurs d'activation/désactivation et la fonction booléenne de contrôle F_{BC} sont décrits dans la section suivante.

Grâce à l'activation/désactivation des contrôleurs RdP, le contrôle s'effectue en ligne, et la fonction booléenne de contrôle nous indiquera dépendamment de l'état dans lequel se trouve le système à l'instant t , si au moins une place de contrôle est activée dans le graphe de marquage.

2.4 Coordination et Switch entre les contrôleurs RdP

Dans cette section, une politique d'alternance entre les contrôleurs RdP est développée. Un switchage entre les places moniteurs sera réalisé dans le graphe de marquage $G(N, M_0)$, dépendamment de la coupe minimale dans laquelle le processus existe, en calculant les vecteurs d'activation et de désactivation. En outre, la fonction booléenne de contrôle F_{BC} sera introduite dans ce qui suit.

2.4.1 Les vecteurs d'activation et de désactivation

Après avoir identifié les coupes minimales dans le graphe de marquage et leurs places de contrôle correspondantes, nous procédons à l'étape d'activation/désactivation des contrôleurs RdP. Une fois que le processus entre dans une coupe minimale CM_i , son superviseur P_{ci} correspondant aura la tâche de contrôler sa zone dans le graphe de marquage $G(N, M_0)$. En effet, un contrôleur RdP est initialement inactif (actif) jusqu'à ce qu'on entre (sorte) d'une coupe minimale par l'intermédiaire des états frontières MF_i . Ci-après, on définit formellement des vecteurs d'activation et de désactivation.

Définition 2.13 :

On associe pour chaque état frontière $M \in MF_i$ un vecteur d'activation (désactivation), qui présente un chemin entre l'état initial du graphe de marquage M_0 et $M \in MF_i$, en éliminant les éventuels cycles qui peuvent réactiver (re-désactiver) le contrôleur P_{ci} .

- Vecteur d'activation : $\vec{\sigma}_A(M) = \sigma \in T^* \mid \exists M_0[\sigma > M, M \in MF$
- Vecteur de désactivation : $\vec{\sigma}_D(M) = \sigma \in T^* \mid \exists M_0[\sigma > M \wedge M[t >, M \in MF, t \notin CM_i$

Théorème 2.2 :

Une fois qu'un marquage entre dans une coupe minimale CM_i par un état intermédiaire $M \in MIn_i$, le marquage initial de la place de contrôle $M_{0i}(P_{ci})$ change comme suit : $M_{0i}(P_{ci}) = M_{0i}(P_{ci}) + C(P_{ci}, \cdot)[\vec{\sigma}_A(M) - \vec{\sigma}_A(M_{0i})]$ (voir figure 2.11).

Preuve :

Soit les séquences de tir $\sigma_A(M_{0i})$ et $\sigma_A(M)$, qui sont validées à partir de l'état initial du graphe de marquage M_0 et qui conduisent aux marquages M_0 et $M \in MIn_i$ d'une coupe minimale CM_i . En tenant compte de la proposition 2.1 ($\sigma_A(M_{0i}) < \sigma_A(M)$) et de [DX03], $\exists \sigma'' \mid M_{0i}[\sigma'' >, \sigma'' = \sigma_A(M) - \sigma_A(M_{0i})$ le marquage obtenu après avoir franchi la séquence σ'' à partir de M_{0i} est égal à : $M_{0i} + C[\vec{\sigma}_A(M) - \vec{\sigma}_A(M_{0i})] = (M_0 + C\vec{\sigma}_A(M_{0i})) + C[\vec{\sigma}_A(M) - \vec{\sigma}_A(M_{0i})] = M_0 + C.\vec{\sigma}_A(M) = M. M \in MIn_i$. On peut déduire que $M_0[\sigma_A(M_{0i})\sigma'' > M, M \in MIn_i$ et $\sigma'' = \sigma_A(M) - \sigma_A(M_{0i})$ (voir figure 2.11).

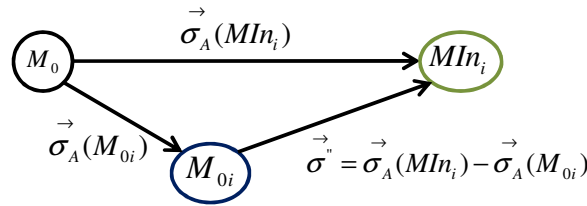


Figure 2.11- Relation entre M_{0i} et MIn_i

2.4.2 Fonction booléenne de contrôle

On constate que tous les contrôleurs RdP synthétisés peuvent contrôler le système (S), soit simultanément, soit alternativement. Plus précisément, en fonction de l'état dans lequel se trouve le processus, le contrôleur RdP est activé ou désactivé. Ainsi, on peut définir formellement la fonction booléenne de contrôle F_{Bc} comme suit :

$$F_{Bc} = F(P_{c1}, P_{c2} \dots P_{cm}) = P_{c1} + P_{c2} + \dots P_{cm} = \sum_{i=1}^m P_{ci}$$

Cette fonction logique **OU** caractérise l'union des contrôleurs RdP. En effet, F_{Bc} prend la valeur de 1 si le système (S) est contrôlé, et 0 sinon :

$$F_{Bc}: \begin{cases} 1 & \text{Si } S \text{ est contrôlé} \\ 0 & \text{Sinon} \end{cases}$$

Dans la figure 2.12, on présente un exemple de réseau de Petri (figure 2.12(a)) et son graphe de marquage (figure 2.12(b)), qui modélise le partage de ressources de deux machines. On applique la *CGEM* suivante sur le RdP pour illustrer notre approche :

$$M(P_2) + M(P_3) + M(P_5) + M(P_6) \leq 1$$

En appliquant notre politique de contrôle, on identifie deux coupes minimales (CM_1 et CM_2) dans le graphe de marquage $G(N, M_0)$ et par conséquent, on synthétise leurs deux contrôleurs RdP correspondants (P_{c1} et P_{c2}).

Dans cet exemple, les superviseurs RdP interviennent alternativement. La fonction booléenne de contrôle correspondante est : $F_{BC} = P_{c1} + P_{c2}$. Si les deux superviseurs sont désactivés (activés), alors F_{BC} prend la valeur de 0 (1), sinon, si l'un des deux superviseurs est activé, alors le système est considéré comme contrôlé et F_{BC} prend toujours la valeur de 1.

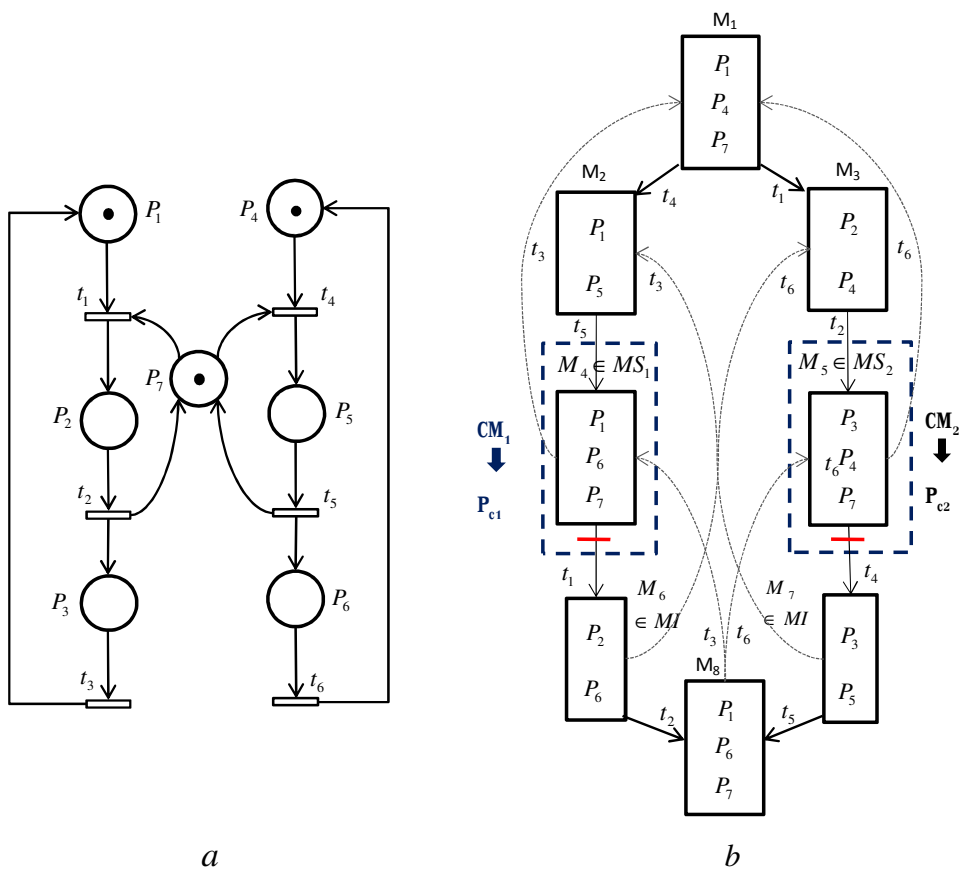


Figure 2.12- Partage de ressources

2.5 Application : Commande d'un système de production flexible (FMS)

Afin d'illustrer nos approches développées tout au long de ce chapitre, nous allons appliquer notre politique de contrôle sur un système de production flexible *FMS* implémenté dans la plateforme de notre laboratoire. Plus précisément, nous comptons commander deux stations

de l'atelier flexible qui font la même tâche, à savoir *l'assemblage des pièces en verre gravées* avec des socles de différentes couleurs.

Le système physique ainsi que sa modélisation par réseaux de Petri sont représentés dans la figure 2.13. Afin de toujours satisfaire la commande du client, nous imaginons le scénario suivant : La première station modélisée par P_2 représente notre autoproduction, tandis que la deuxième station (P_3) modélise un sous-traitant. Les places C_M et C_S indiquent la capacité de trois pièces des deux machines. De plus, la place P_1 présente notre stock de production (4 pièces).

Les évènements (t_1, t_2) et (t_4, t_5) modélisent respectivement les évènements d'entrée et de sortie des palettes dans les deux stations à travers leurs convoyeurs annexes, sinon la palette peut continuer son chemin dans le convoyeur principal en franchissant la transition t_3 . On note que seules les transitions (t_1, t_2) sont contrôlables. Le graphe de marquage $G(N, M_0)$ contient 13 états (voir figure 2.14).

La spécification de contrôle à satisfaire est de maintenir le marquage de P_2 plus le double du marquage de P_3 moins de trois jetons :

$$CGEM: M(P_2) + 2M(P_3) \leq 3$$

Cette contrainte exprime la spécification (Q) suivante : On souhaite confier au minimum notre production à un sous-traitant dans le cadre d'un travail de sous-œuvre.

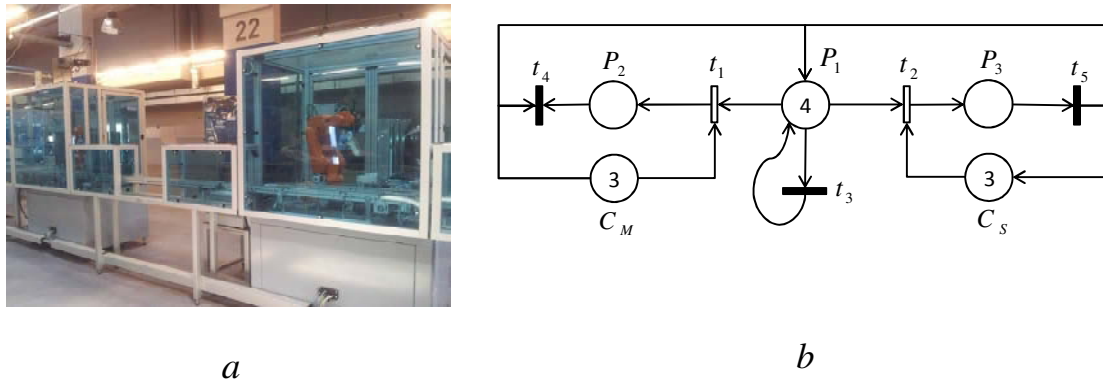


Figure 2.13- Environnement expérimental

2.5.1 Application de l'approche 1

Ce travail propose une approche efficace pour déterminer une coupe minimale CM_i avec un nombre minimal de nœuds dans laquelle la synthèse de contrôle est appliquée. Selon la définition 2.9, l'ensemble Ω contient cinq instances de séparation d'évènements corrélées soit par t_2 soit par t_1 :

$$\Omega_1 = \{(M_2[t_2 > M_4]), (M_5[t_2 > M_8]), (M_{10}[t_2 > M_{13}])\}.$$

$$\Omega_2: \{M_5[t_1 > M_9]\}.$$

Ce qui donne deux coupes minimales dans $G(N, M_0)$ (voir figure 2.14) à savoir : $CM_1 = \{(M_2[t_2 >]), (M_2[t_1 >]), (M_3[t_1 >]), (M_5[t_2 >]), (M_5[t_5 >]), (M_6[t_2 >]), (M_6[t_1 >]), (M_{10}[t_2 >])\}$; $CM_2 = \{(M_5[t_1 >])\}$.

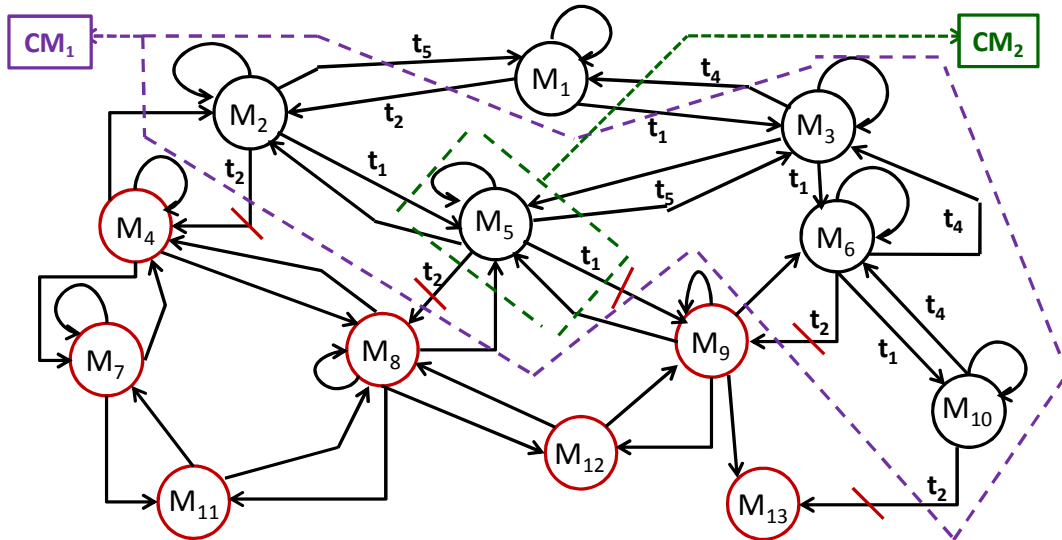


Figure 2.14- Graphe de marquage 1 avec des coupes minimales

Tableau 2.1 : Informations sur les marquages des coupes minimales

Coupe minimale	Marquage	Classification	Marquage $[p_1, p_2, p_3]^T$
CM_1	M_2	MS_1	$[3,0,1]^T$
	M_4	MI_1	$[2,0,2]^T$
	M_5	MS_1	$[2,1,1]^T$
	M_8	MI_1	$[1,1,2]^T$
	M_6	MS_1	$[2,2,0]^T$
	M_9	MI_1	$[1,2,1]^T$
	M_{10}	MS_1	$[1,3,0]^T$
	M_{13}	MI_1	$[0,3,1]^T$
	M_3	Min_1	$[3,1,0]^T$
CM_2	M_5	MS_2	$[2,1,1]^T$
	M_9	MI_2	$[1,2,1]^T$

Les informations détaillées sur les marquages de chaque coupe minimale sont données dans le tableau 2.1. Les états initiaux des coupes CM_1 et CM_2 sont : $M_{01} = M_2$ et $M_{02} = M_5$. En outre, en appliquant l'algorithme 2.2 de synthèse de contrôle, on obtient les deux contrôleurs RdP P_{c1} et P_{c2} qui seront ajoutés au modèle initial de la figure 2.13 (b) pour avoir le modèle contrôlé présenté dans la figure 2.15 :

$$P_{c1}: \begin{cases} M_{01}(P_{c1}) = M_2(P_{c1}) = 1 \\ C(P_{c1}, \cdot) = (-1, -2, 0, 1, 2) \end{cases}$$

$$P_{c2}: \begin{cases} M_{02}(P_{c2}) = M_5(P_{c2}) = 1 \\ C(P_{c2}, \cdot) = (-2, 0, 0, 0, 0) \end{cases}$$

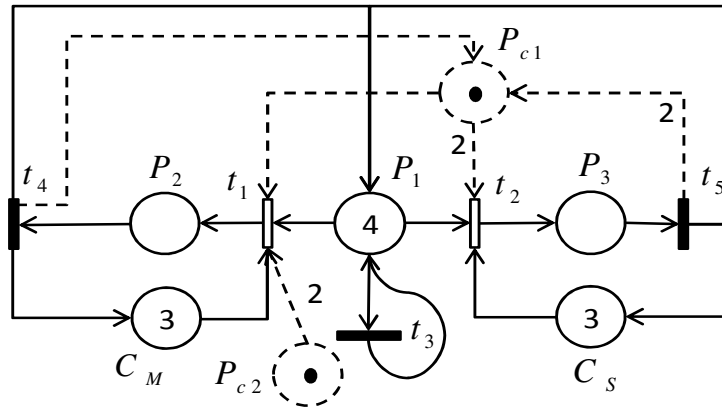


Figure 2.15- Modèle RdP contrôlé 1

Selon notre politique de contrôle, le Switch entre les contrôleurs RdP P_{c1} et P_{c2} est fait par l'intermédiaire des états frontières : $MF_1 = \{M_2, M_3, M_5\}$; $MF_2 = \{M_5\}$. On peut déduire qu'il est seulement dans l'état M_5 où les superviseurs sont impliqués simultanément, sinon ils agissent alternativement. La fonction booléenne de contrôle correspondante est la suivante:

$$F_{Bc} = P_{c1} + P_{c2}$$

Si le processus entre dans la coupe minimale CM_1 par l'état intermédiaire de $M_3 \in Min_1$, le marquage initial de P_{c1} change comme suit :

$$M_{01}(P_{c1}) = M_{01}(P_{c1}) + C(P_{c1}, \cdot)[\sigma_A(M_3) - \sigma_A(M_2)] = 2$$

Pour coordonner entre les contrôleurs, des informations détaillées des vecteurs d'activation et de désactivation, des états frontières de P_{c1} et P_{c2} sont listés dans le tableau 2.2.

Tableau 2.2 : Les vecteurs d'activation et de désactivation

Marquage	Vecteur d'activation	Vecteur de désactivation
M_2	$[0 \ 1 \ 0 \ 0 \ 0]^T$	$[0 \ 1 \ 0 \ 0 \ 1]^T$
M_3	$[1 \ 0 \ 0 \ 0 \ 0]^T$	$[1 \ 0 \ 0 \ 1 \ 0]^T$
M_5	$[1 \ 1 \ 0 \ 0 \ 0]^T$	$[1 \ 1 \ 0 \ 1 \ 1]^T$

2.5.2 Application de l'approche 2

La conception des coupes minimales, dans ce travail, diffère de la dernière approche et ne s'intéresse qu'aux instances de séparation d'évènements avec une possibilité de fusion des coupes minimales selon certains critères.

En effet, en utilisant l'algorithme 2.3, on génère l'ensemble des états sources $MS = \{M_2, M_5, M_6, M_{10}\}$ et des arcs $T_a(M_i)$ montrés dans la figure 2.16 :

$$\begin{aligned} T_a(M_2) &= \{(M_2[t_1 >), (M_2[t_3 >), (M_2[t_5 >)\} \\ T_a(M_5) &= \{(M_5[t_3 >), (M_5[t_4 >), (M_5[t_5 >)\} \\ T_a(M_6) &= \{(M_6[t_1 >), (M_6[t_3 >), (M_6[t_4 >)\} \\ T_a(M_{10}) &= \{(M_{10}[t_3 >), (M_{10}[t_4 >)\} \end{aligned}$$

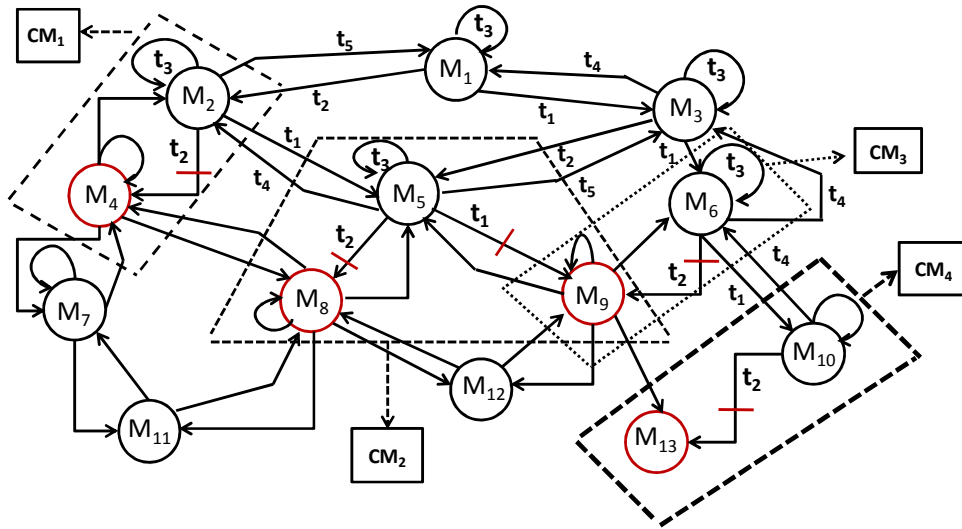


Figure 2.16- Graphe de marquage 2 avec des coupes minimales

L'ensemble des arcs sortant des états sources $M_i \in MS$, qui ne font pas partie de l'ensemble des arcs $T_a(M_i)$, constituent initialement les coupes minimales :

$$\begin{aligned} CM_1 &= \{(M_2[t_2 > M_4])\} \\ CM_2 &= \{(M_5[t_2 > M_8]), (M_5[t_1 > M_9])\} \\ CM_3 &= \{(M_6[t_2 > M_9])\} \\ CM_4 &= \{(M_{10}[t_2 > M_{13}])\} \end{aligned}$$

On remarque que les états sources M_5 et M_6 partagent un état interdit commun (M_9), mais la fusion entre CM_2 et CM_3 n'est pas possible. En effet, la transition t_1 entre M_5 et M_9 appartient à $T_a(M_6)$. Ensuite, l'application de l'algorithme 2.4 nous génère les contrôleurs RdP suivants :

$$\begin{aligned} P_{c1}: & \begin{cases} M_{01}(P_{c1}) = 1 \\ C(P_{c1}, \cdot) = (0, -2, 0, 0, 0) \end{cases} \\ P_{c2}: & \begin{cases} M_{02}(P_{c2}) = 1 \\ C(P_{c2}, \cdot) = (-2, -2, 0, 0, 0) \end{cases} \\ P_{c3}: & \begin{cases} M_{03}(P_{c3}) = 1 \\ C(P_{c3}, \cdot) = (0, -2, 0, 0, 0) \end{cases} \\ P_{c4}: & \begin{cases} M_{04}(P_{c4}) = 1 \\ C(P_{c4}, \cdot) = (0, -2, 0, 0, 0) \end{cases} \end{aligned}$$

Notamment, P_{c1} , P_{c3} et P_{c4} sont identiques. Par conséquent, les contrôleurs redondants P_{c3} et P_{c4} sont éliminés. Ainsi, le modèle RdP contrôlé est présenté dans la figure 17 suivante :

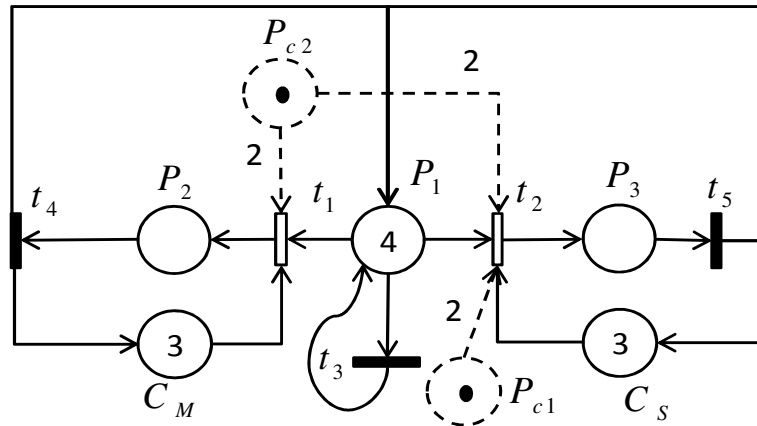


Figure 2.17- Modèle RdP contrôlé 2

La fonction booléenne de contrôle est donnée comme suit :

$$F_{Bc} = P_{c1} + P_{c2}$$

Les informations concernant les marquages des coupes minimales sont données dans le tableau 2.3 suivant :

Tableau 2.3 : Informations sur les marquages des coupes minimales

Coupe minimale	Marquage	Classification	Marquage $[p_1, p_2, p_3]^T$
CM_1	M_2	MS_1	$[3,0,1]^T$
	M_4	MI_1	$[2,0,2]^T$
CM_2	M_8	MI_2	$[1,1,2]^T$
	M_5	MS_2	$[2,1,1]^T$
	M_9	MI_2	$[1,2,1]^T$
CM_3	M_6	MS_3	$[2,2,0]^T$
	M_9	MI_3	$[1,2,1]^T$
CM_4	M_{10}	MS_4	$[1,3,0]^T$
	M_{13}	MI_4	$[0,3,1]^T$

Dans cette approche, la coordination entre les superviseurs RdP P_{c1} et P_{c2} est assurée par le billet des états frontières : $MF_1 = \{M_2\}$, $MF_2 = \{M_5\}$, $MF_3 = \{M_6\}$ et $MF_4 = \{M_{10}\}$. Les vecteurs d'activation et de désactivation des contrôleurs sont donnés dans le tableau 2.4 suivant :

Tableau 2.4 : Les vecteurs d'activation et de désactivation

Marquage	Vecteur d'activation	Vecteur de désactivation
M_2	$[0\ 1\ 0\ 0\ 0]^T$	$[0\ 1\ 0\ 0\ 1]^T$
M_5	$[1\ 1\ 0\ 0\ 0]^T$	$[1\ 1\ 0\ 1\ 1]^T$
M_6	$[2\ 0\ 0\ 0\ 0]^T$	$[2\ 0\ 0\ 1\ 0]^T$
M_{10}	$[3\ 0\ 0\ 0\ 0]^T$	$[3\ 0\ 0\ 1\ 0]^T$

2.6 Comparaison avec les travaux précédents

Dans cette section, on présente trois tableaux qui nous permettent de comparer notre méthode avec les travaux précédents. D'abord, on compare ce travail avec l'approche classique de la

théorie des régions dans la synthèse de contrôle de Ghaffari et al [GRX02ab] avec l'exemple 1.

En outre, dans l'exemple 2, on confronte notre méthode avec les politiques de prévention de blocage des RdP et, en particulier, avec la méthode de Huang et Pan [HP11]. En effet, cette méthode présente de nombreux avantages et résout le problème de transition d'état interdite. Leurs résultats expérimentaux prouvent que leur politique de contrôle est la plus efficace en la comparant avec les méthodes existantes de prévention de blocage, tels que les travaux de recherche de Li et al. [LZJ08], Park et al. [PR00], Uzam et al. [UZ07] et Pirrodi et al. [PCF08]. Bien qu'il faille noter que toutes ces méthodes ne considèrent pas les transitions incontrôlables dans leurs modèles RdP utilisés dans l'expérimentation.

Enfin, on compare dans l'exemple 3 notre travail avec l'approche des contrôleurs partiels de Ghaffari et al. [GRX03ab]. Cette approche permet de résoudre le problème de non-existence de contrôleurs RdP en utilisant l'approche classique de la théorie des régions. Pour plus de commodité, les deux approches de notre méthode sont dénotées par S_1 et S_2 . Par ailleurs, les travaux de recherche de [GRX02ab], [GRX03ab] et [HP11] sont appelés respectivement, G_1 , G_2 et H .

2.6.1 Exemple 1 (FMS, G_1)

Le tableau 2.5 énumère les résultats expérimentaux de synthèse de contrôleurs RdP avec l'approche G_1 . Pour l'exemple 1 du FMS , on peut remarquer que pour le même nombre de marquages admissibles et d'instances de séparation d'évènements, on peut synthétiser le même nombre de superviseurs RdP maximum permissifs avec un nombre (L) d'équations de la théorie des régions très réduit.

En effet, les deux places de contrôle trouvées avec G_1 sont obtenues en résolvant 40 équations avec un temps de calcul de 593 ms, alors qu'avec notre méthode de coupe minimale, le nombre L est divisé par 3. Cette politique de contrôle traduite par S_1 et S_2 réduit énormément le temps de calcul des contrôleurs RdP de 593 ms à 219 ms et 190 ms (voir tableau 2.5). Les systèmes contrôlés qui traduisent les trois approches G_1 , S_1 et S_2 sont obtenus et décrits dans le tableau 2.6.

Par conséquent, d'après ces résultats expérimentaux, on peut déduire que notre méthode traduite par S_1 et S_2 peut considérablement réduire la complexité de calcul de la théorie des régions dans la synthèse de commande.

Tableau 2.5 : Les comparaisons de l'exemple 1

Approche	Nb de Moniteurs	Marquages admissibles	Ω	L	Temps de calcul (ms)	Méthode
S_1	2	6	5	15	219	Coupe minimale
S_2	2	6	5	13	190	Coupe minimale
G_1	2	6	5	40	593	Basée sur $G(N, M_0)$

Tableau 2.6: Contrôleurs RdP additionnés

Approche	Superviseurs	$M_{0i}(P_{ci})$	$C(P_{ci}, \cdot)$
S_1	P_{c1}	1	(-1,-2,0,1,2)
	P_{c2}	1	(-2,0,0,0,0)
S_2	P_{c1}	1	(0,-2,0,0,0)
	P_{c2}	1	(-2,-2,0,0,0)
G_1	P_{c1}	1	(0,-1,0,0,1)
	P_{c2}	3	(-1,-2,0,1,2)

2.6.2 Exemple 2 (H)

L'exemple 2 pris de [HP11] est dédié à la prévention de blocage des RdP. En se basant sur le tableau 2.7, il nous faut 4 instances de séparation d'évènements et 46 équations à résoudre pour la théorie des régions, afin que la méthode H soit opérative.

D'autre part, notre nouvelle méthode nécessite 10 instances de séparation d'évènements ainsi que 22 et 18 équations à résoudre respectivement pour les approches S_1 et S_2 . Dans cet exemple, trois contrôleurs RdP sont utilisés pour la méthode H contre trois et quatre superviseurs pour S_1 et S_2 . Les modèles RdP contrôlés pour les approches S_1 , S_2 et H sont donnés dans le tableau 2.8.

L'avantage principal des algorithmes de S_1 et S_2 est le gain du temps de calcul qui passe de 489 ms à 234 et 191 ms respectivement pour S_1 et S_2 . En effet, pour le même nombre de marquages admissibles (soit 20 états), on ne résout que 22 et 18 équations pour S_1 et S_2 dans notre politique de contrôle comparée à 46 équations pour la méthode H . Il existe un autre facteur très important dans la comparaison entre les trois méthodes, qui ne figure pas dans le tableau 2.7. En fait, la synthèse de commande pour la méthode H ne considère que des transitions contrôlables (méthode sous contrôlabilité totale) dans les modèles RdP, ce qui présente une faiblesse dans ce travail.

Par conséquent, on peut déduire que notre nouvelle méthode est plus efficace que la méthode H en terme de facilité de calcul des contrôleurs RdP avec un temps de calcul nettement plus réduit.

Tableau 2.7 : Les comparaisons de l'exemple 2

Approche	Nb de Moniteurs	Marquages admissibles	Ω	L	Temps de calcul (ms)	Méthode
S_1	3	20	10	22	234	Coupe minimale
S_2	4	20	10	18	191	Coupe minimale
H	3	20	4	46	489	Basée sur $G(N, M_0)$

Tableau 2.8: Contrôleurs RdP additionnés

Approche	Superviseurs	$M_{0i}(P_{ci})$	$C(P_{ci}, \cdot)$
S_1	P_{c1}	0	(-2,0,0,0,0,0,0,0,0,0)
	P_{c2}	0	(0,-2,0,0,0,0,0,0,0,0)
	P_{c3}	0	(0,0,0,0,0,-2,0,0,0,0)
S_2	P_{c1}	1	(0,0,0,0,0,0,0,0,-2,0)
	P_{c2}	1	(0,0,0,-2,0,0,0,0,0,0)
	P_{c3}	1	(0,0,0,0,0,0,0,-2,-2,0)
	P_{c4}	1	(0,0,0,-2,-2,0,0,0,0,0)
H	P_{c1}	2	(0,0,0,0,1,-1,0,1,-1,0)
	P_{c2}	3	(0,0,0,1,-1,0,0,1,-1,0)
	P_{c3}	2	(0,0,0,1,-1,0,0,0,1,-1)

2.6.3 Exemple 3(G_2)

L'exemple 3 est pris de [GRX03a] qui présente un problème de type PTEI insoluble avec la théorie des régions (TR). En effet, la méthode G_2 est utilisée quand le système linéaire (SL) de la TR est insoluble ; c'est la méthode des contrôleurs partiels. L'idée globale est que chaque contrôleur partiel contient un ensemble de places de contrôle qui traduisent plusieurs SL relaxés de certaines équations qui forment une combinaison convexe dans SL .

En se basant sur le tableau 2.9, la méthode G_2 nécessite 236 équations à résoudre et 8 instances de séparation d'évènements pour contrôler cet exemple. Par contre, la réduction de calcul dans nos approches S_1 et S_2 est très claire. En effet, chaque approche traite seulement 16 équations, ce qui prouve aussi la forte réduction du temps de calcul qui passe de 1540 ms à 110 ms et 113 ms pour S_1 et S_2 .

En outre, cinq places moniteurs sont obtenues avec le procédé G_2 , ce qui définit trois contrôleurs partiels contre 2 superviseurs RdP pour chacune des approches S_1 et S_2 . Les modèles contrôlés, qui nous permettent d'obtenir des superviseurs RdP maximum permissifs, garantissant l'atteignabilité de 30 états admissibles pour les trois approches S_1, S_2 et G_2 sont indiqués dans le tableau 2.10.

Ainsi, non seulement la complexité de résolution est réduite dans cet exemple, mais le problème PTEI insoluble est également résolu avec nos approches, en appliquant une méthode plus simple. Par conséquent, on déduit que les approches S_1 et S_2 se montrent plus efficaces que la méthode G_2 .

Tableau 2.9 : Les comparaisons de l'exemple 3

Approche	Nb de Moniteurs	Marquages admissibles	Ω	L	Temps de calcul (ms)	Méthode

S_1	2	30	8	16	10	Coupe minimale
S_2	2	30	8	16	113	Coupe minimale
G_2	5	30	8	236	1540	Contrôleurs partiels

Tableau 2.10: Contrôleurs RdP additionnés

Approche	Superviseurs	$M_{0i}(P_{ci})$	$C(P_{ci}, \cdot)$
S_1	P_{c1}	0	(-2,0,0,0)
	P_{c2}	0	(0,-2,0,0)
S_2	P_{c1}	1	(0,-2,0,0)
	P_{c2}	1	(-2,0,0,0)
G_2	P_{c1}	4	(-1,-1,0,2)
	P_{c2}	7	(-2,-1,-2,5)
	P_{c3}	0	(1,-3,0,2)
	P_{c4}	9	(-3,-2,2,3)
	P_{c5}	0	(1,-3,2,0)

2.7 Conclusion

Dans ce chapitre, nous avons présenté deux approches originales de synthèse de contrôleurs RdP pour un problème de transitions d'état interdites en présence de transitions incontrôlables. Un problème de type PTEI est plus général qu'un problème PEI. Les contrôleurs synthétisés satisfont nos spécifications de contrôle données sous forme de contraintes linéaires de type *CGEM*, tout au long de ce chapitre.

Grâce aux comparaisons effectuées dans la section précédente, nous avons pu montrer que notre méthode est plus efficace et simple que les travaux antérieurs cités [GRX02], [HP11] et [GRX03]. En effet, pour un même nombre de places moniteurs, nous synthétisons des contrôleurs avec un nombre d'équations très réduit dans les systèmes linéaires *SL* de la théorie des régions (*TR*) et un temps de calcul nettement plus inférieur en le comparant avec d'autres méthodes.

Dans nos approches, l'application de la théorie des régions est confiée aux coupes minimales CM_i et non pas sur la totalité du graphe de marquage $G(N, M_0)$; i.e. l'union de toutes les coupes minimales ne constitue pas la totalité du graphe ; ce qui prouve la forte réduction de la complexité de calcul de la théorie des régions.

En outre, notre méthode résout également les problèmes insolubles avec l'approche classique de la théorie des régions dans la synthèse de contrôle, grâce à l'explosion combinatoire des

états dans $G(N, M_0)$ ou la convexité des équations d'état qui peuvent exister dans le système linéaire de la théorie des régions.

Nous avons pu appliquer notre politique de contrôle sur un système de production flexible (*FMS*) et implémenter nos contrôleurs RdP avec la présence des transitions incontrôlables sur des automates programmables. Ce travail a été réalisé sur la plateforme de notre laboratoire dans le cadre d'un stage avec une élève ingénieur, lors de son projet de fin d'études.

Enfin, nous avons repris d'autres exemples de la littérature, qui utilisent la théorie des régions dans la synthèse de contrôle, dans la littérature comme les exemples de [GRX02], [GRX03] et [HP11], afin de montrer la considérable réduction du coût de calcul de la théorie des régions. Notre politique de contrôle fournit une méthode efficace pour synthétiser des contrôleurs maximum permissifs.

Dans le chapitre suivant, nous développerons d'autres méthodes de synthèse de commande qui nous permettent également de réduire la complexité de calcul des superviseurs RdP.

Chapitre 3

Commande Optimale des RdP via les marquages canoniques et les contraintes de chemins admissibles

***Résumé:** Dans ce troisième chapitre, nous présentons deux nouvelles approches de synthèse de la commande basées sur les réseaux de Petri et la théorie des régions. La première est fondée sur la détermination des marquages canoniques pour synthétiser un contrôleur maximum permissif, alors que la deuxième se base sur l'identification des chemins admissibles dans le graphe de marquage. En effet, les marquages canoniques sont conçus à partir d'un état interdit partiellement connu et forment un ensemble d'états sur lesquels s'applique la théorie des régions. La deuxième approche, nécessite l'utilisation du graphe de marquage pour la synthèse de contrôle. Effectivement, certaines spécification de contrôle ne peuvent pas être exprimées par des Contraintes Généralisées d'Exclusion Mutuelle; pour cette raison, de nouvelles contraintes basées sur des séquences de transition et appelées contraintes de chemins interdits sont introduites. Les deux approches vont être appliquées sur des stations d'un système de production flexible. Les contributions de ce chapitre ont fait l'objet des publications suivantes : [RRKA14] et [RAR15].*

3.1 Introduction

Les méthodes d'analyse des systèmes à évènements discrets (SED) comprennent des analyses structurelles du RdP donné et des analyses de son graphe de marquage $G(N, M_0)$. De nombreuses méthodes ont été développées pour le contrôle de ces systèmes en utilisant l'analyse structurelle [GRX02a] [GNRX02] [GRX02c] [ECM95] [HP11]. Certains de ces problèmes de contrôle par supervision ont été résolus en utilisant la théorie des régions.

Cette théorie a été utilisée pour la synthèse de la commande afin de synthétiser des superviseurs RdP en identifiant l'ensemble des états admissibles, interdits, bloquants et un ensemble d'instances de séparation d'évènements. Néanmoins, la méthode proposée dans [GRX02a] est maximale permissive lorsque la solution existe, mais le calcul devient fastidieux lorsque le graphe de marquage $G(N, M_0)$ est assez conséquent.

Plusieurs travaux de recherche ont tenté de réduire le coût de calcul des contrôleurs RdP. En effet, dans [LZZMJ08], les auteurs ont utilisé le concept des siphons comme une approche de réduction avant d'utiliser la théorie des régions pour calculer les contrôleurs RdP. En outre, [HP11] propose une issue pour la prévention de blocage en minimisant le nombre de conditions d'atteignabilité de la théorie des régions. Il est à remarquer que la méthode de [HP11] s'avère plus efficace en termes de réduction du temps de calcul en comparant ses résultats expérimentaux avec ceux de [UMZM07] [UM10], [PJRS01] et [QMLZZ12]. Dans ce travail, une nouvelle approche de réduction du coût de calcul de la théorie des régions est développée. Cette approche est basée sur les marquages canoniques.

De surcroît, le comportement admissible des SED suscite un grand intérêt pour les chercheurs. En particulier dans [GDS92], il a été montré que si l'ensemble des marquages admissibles est exprimé par un ensemble de contraintes d'inégalité linéaires appelées Contraintes Généralisées d'Exclusion Mutuelle (*CGEM*) et si tous les évènements sont contrôlables, alors une solution à base de réseaux de Petri existe et est, au maximum, permissive. Cependant, certaines *CGEM* n'arrivent pas à exprimer le vrai sens d'une spécification de contrôle donnée.

Dans ce chapitre, on propose deux approches qui ont deux objectifs : en premier lieu, la détermination d'un ensemble de marquages canoniques sur lequel s'applique la théorie des régions, sans utiliser le graphe de marquage $G(N, M_0)$, en s'inspirant du concept des coupes minimales (développé dans le chapitre 2) et en second lieu, la détermination des chemins admissibles, en développant un nouveau type de contraintes qui ne se fonde pas sur la combinaison de marquages comme les *CGEM*, mais sur des combinaisons de séquences de transition. Ces deux approches ont un point commun : c'est la synthèse d'un contrôleur RdP qui garantit un comportement maximum permissif désiré.

Dans ce qui suit, nous présentons la première approche des marquages canoniques dans la synthèse de contrôle (section 3.1) en appliquant la théorie des régions sur un exemple réel. Ensuite, nous développons le nouveau type de contraintes dans la section 3.3, appelées Contraintes de Chemins Admissibles (*CCA*) afin d'obtenir un comportement légal des SED.

Les approches présentées dans ce chapitre seront appliquées sur un système de production flexible.

3.2 Approche 1 : Synthèse de contrôleurs RdP basée sur les marquages canoniques

La théorie des régions pour la synthèse de la commande est maximale permissive si la résolution de son système linéaire possède une solution. Néanmoins, l'explosion combinatoire des états, présente le désavantage majeur de l'approche de Ghaffari et al. [GRX02a] [GRX02b].

Dans nos précédents travaux [RARK14] [RAR14] et [RARK15], nous avons présenté une nouvelle politique de contrôle basée sur les coupes minimales dans un graphe de marquage. La présente méthode se concentre aussi sur la réduction du coût de calcul de la théorie des régions en s'inspirant de la méthode des coupes minimales (*CM*). En effet, ce concept est fusionné dans la présente approche de telle sorte qu'on peut *temporairement* se déposséder du graphe de marquage pour synthétiser le superviseur RdP.

Dans ce travail, la théorie des régions est appliquée sur l'ensemble des marquages canoniques qui substitue le graphe de marquage pour concevoir les contrôleurs RdP. Néanmoins, nous n'aurons besoin de $G(N, M_0)$ qu'au moment du calcul des vecteurs d'activation/désactivation des contrôleurs RdP. Cette approche allège le contrôle, et en même temps, réduit la complexité de calcul de la théorie des régions et le temps de calcul des contrôleurs.

Définition 3.1 : (Chainage arrière CH_b)

Le chainage arrière CH_b est une méthode qui consiste à remonter depuis les conclusions pour arriver aux axiomes, c.-à-d. qu'en partant d'un état, on réussit à générer d'autres états prédécesseurs.

$$CH_b = \sum_i^{(t)} p_i \mid p_i \in t^{(p)}, M(p_i) = w(p_i, t)$$

Définition 3.2 : (Chainage avant CH_f)

Par opposition au chainage arrière, le chainage avant CH_f est une méthode de déduction qui applique les règles de la logique séquentielle des états successifs, par le franchissement des transitions d'états correspondantes.

$$CH_f = \sum_i^{(t)} p_i \mid p_i \in {}^{(p)}t, M(p_i) \geq w(p_i, t)$$

Définition 3.3 : (Etat interdit \widetilde{MI})

Un état interdit \widetilde{MI}_i est un état primitif qui ne respecte pas une spécification de contrôle de type *CGEM*. A partir de cet état, on effectue des chainages arrière et des chainages avant pour générer d'autres états qui peuvent exister dans le graphe de marquage $G(N, M_0)$.

$$\widetilde{MI} = \{\widetilde{M} \in M \mid \widetilde{M} \neq M_{(w,k)}\}$$

La conception des marquages canoniques nécessite un état primitif (initialement l'état interdit \widetilde{MI}_i) à partir duquel on effectue des chaînages arrière (CH_b) et des chaînages avant (CH_f) pour générer l'ensemble des états sur lesquels on applique la théorie des régions pour la synthèse de contrôle.

3.2.1 Conception des marquages canoniques

Dans cette partie, on s'intéresse à la conception des marquages canoniques sur lesquels on applique la théorie des régions afin de synthétiser le superviseur RdP.

Définition 3.4 : (Marquages canoniques \widehat{Mc})

Un marquage canonique est un pseudo-état avec la forme de base d'un état mais qui n'est pas un état complet c.à.d. le marquage de certaines places p_i est inconnu. Soit \widehat{Mc} l'ensemble des états canoniques tel que :

$$\widehat{Mc} = \widetilde{MI} \cup \{\widehat{M} \in M, t \in CH_f \mid \exists (\widehat{M}[t > M'], M' \in \widetilde{MI})\}$$

En s'inspirant de la méthode de CM, l'ensemble des marquages canoniques synthétisés forme une zone appelée Coupe Minimale Canonique CMC. La théorie des régions sera par la suite utilisée et sera limitée à cette CMC pour déterminer le contrôleur RdP correspondant.

Définition 3.5: (Coupe Minimale Canonique CMC)

Une coupe minimale canonique est un ensemble de marquages canoniques \widehat{Mc} sur lesquels on applique la théorie des régions pour synthétiser une place de contrôle P_{ci} en se dépossédant du graphe de marquage $G(N, M_0)$ [RAR15].

$$CMC = \{M [t > M' \mid M \in \widehat{Mc}, M' \in \widetilde{MI}\}$$

La conception des marquages canoniques est décrite dans l'algorithme 3.1 suivant.

Algorithme 3.1 : Conception des marquages canoniques \widehat{Mc}

Soit (w, K) une CGEM, \widetilde{MI} l'ensemble des états interdits, \widehat{Mc} l'ensemble des états canoniques, β_n le poids de chaque transition franchie durant un chainage et $\alpha \in \mathbb{N}$.

- 1 Initialisation $i := 1, \widehat{Mc} = \emptyset$
Déterminer l'état interdit $\widetilde{M} \in \widetilde{MI}$ qui ne respecte pas (w, K) dans sa forme canonique suivante : $\widetilde{M} = \sum_{l,j} (\alpha_l w_l) p_j \mid \sum_l \alpha_l w_l = K + \beta_n$. avec α_l : le marquage de p_j , et w_l : la composante du vecteur $\vec{w}^T \in (w, K)$.
- 2
- 3 Faire $\widetilde{MI} \subset \widehat{Mc}$
- 4 Effectuer un chainage arrière CH_b à partir de $\widetilde{M} \in \widetilde{MI}$; $CH_b = \sum^{(t)} p \mid p \in t^{(p)}$ et soit $\widehat{Mc} = \sum_{j=i+1} \widehat{M}_j$ l'ensemble résultant des marquages canoniques.
- 5 Définir les transitions résultant(es) du chainage CH_b , $T \in CH_b$.
- 6 Regrouper tous les marquages $\widehat{M} \in \widehat{Mc}$ dans une Coupe Minimale Canonique CMC
$$CMC = \{(\widehat{M}[t > \widetilde{M}'] \mid \widehat{M} \in \widehat{Mc}, \widetilde{M}' \in \widetilde{MI})\}$$
- 7 Faire $i := i+1$
- 8 Effectuer un chainage avant CH_f , $CH_f = \sum p^{(t)} \mid p \in {}^{(p)}t$ et un chainage arrière CH_b

à partir de $\widehat{M} \in \widehat{M}_c$.

- 9 Définir les transitions résultant(es) des chainages CH_f et CH_b ; $T \in CH_b, T \in CH_f$
FIN

A partir d'un état qui ne respecte pas une spécification de contrôle de type $CGEM(w, K)$, nous déterminons un état interdit sous une forme canonique $\widetilde{MI} = \sum_{l,j} (\alpha_l, w_l) \cdot p_j$ avec la condition suivante : Les coefficients α_l, w_l de chaque marquage de p_j doivent être égaux à la constante K de la $CGEM$ plus β_n le poids correspondant(s) à la transition du chainage arrière CH_b .

Une fois cet état ($\widetilde{M} \in \widetilde{MI}$) choisi, nous effectuons un chainage arrière CH_b et un chainage avant CH_f pour synthétiser les marquages canoniques. Cet ensemble de marquages \widehat{M}_c forme une zone appelée coupe minimale canonique CMC qui peut exister partiellement ou entièrement dans le graphe de marquage $G(N, M_0)$.

En effet, toutes les instances de séparation d'évènements de $G(N, M_0)$ seront retrouvées dans CMC . Par contre, on peut retrouver dans CMC des instances de séparation qui n'existe pas dans $G(N, M_0)$. Ces instances sont de toutes les manières interdites par le procédé.

Finalement, après avoir conçu tous les marquages canoniques ainsi que l'ensemble des transitions T issues des chainages CH_b et CH_f , la théorie des régions peut être appliquée sur CMC pour synthétiser le contrôleur P_c .

Remarque 3.1 :

La coupe minimale CMC contient la forme générique de toutes les instances de séparation d'évènements Ω .

Une fois que les marquages canoniques sont déterminés et forment une coupe minimale canonique, la théorie des régions peut être appliquée selon l'algorithme 3.2 suivant :

Algorithme 3.2 : Synthèse de contrôleur RdP

Soit \widehat{M}_0 l'état initial de la coupe minimale canonique CMC et \widetilde{MI} l'ensemble des états interdits.

- 1 Ecrire les conditions d'atteignabilité des marquages canoniques à partir de \widehat{M}_0
- 2 Ecrire les conditions de séparation d'évènements dans CMC : $\widehat{M}_0[t > \widetilde{M}, \widetilde{M} \in \widetilde{MI}$
- 3 Ecrire les équations de cycles de base s'ils existent dans CMC
- 4 Résoudre le système linéaire et soit $(\widehat{M}_0(P_c), C(P_c, \cdot))$ la solution de la correspondante CMC .

FIN

3.2.2 Exemple théorique

Nous présentons dans cette partie, un système physique composé de trois parties : un train, un ordinateur ou le périphérique de contrôle et une barrière de passage. Soit le réseau de Petri illustré dans la figure 3.1 modélisant le trafic ferroviaire décrit comme suit [LS87] :

Les places P_1, P_2, P_3 et P_4 représentent les différentes positions du train (c.à.d. en approche, juste avant le passage, passage du train et après le passage du train)

De même, les transitions t_1, t_2 et t_3 désignent les évènements suivants : signalisation de l'approche du train, entrée du train dans le croisement et la signalisation du départ de train.

Le périphérique ou le logiciel de commande est représenté par l'ensemble des places ($P_5, P_6, P_7, P_8, P_9, P_{10}$).

Les états de la barrière sont modélisés par les deux places : P_{11} (barrière levée) et P_{12} (barrière en position basse). Les transitions t_6 et t_7 représentent respectivement les évènements de la montée et de la descente de la barrière.

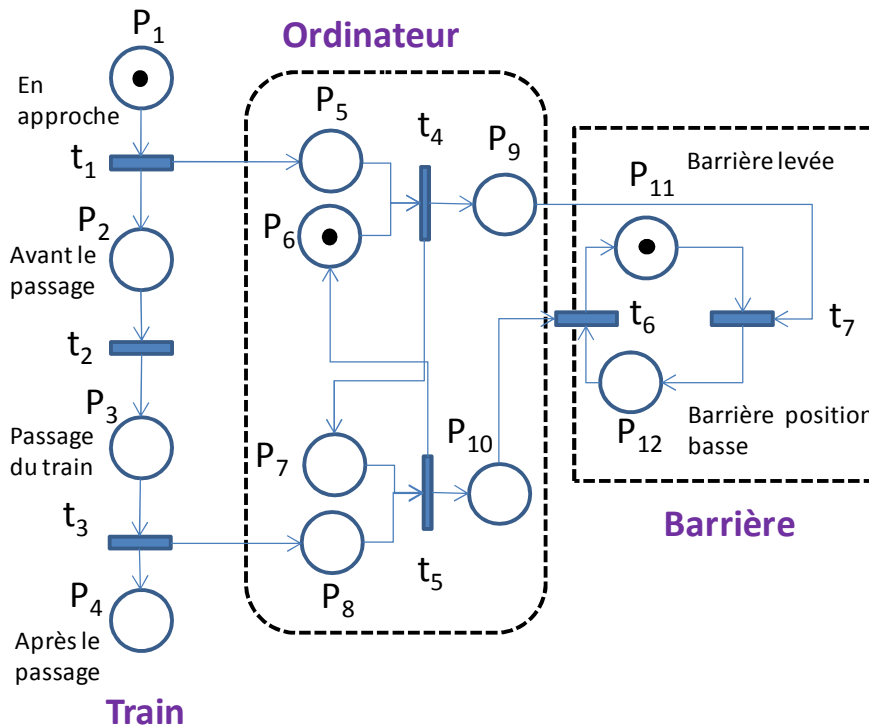


Figure 3.1- Modélisation du trafic ferroviaire par un RdP

Le périphérique de contrôle peut être affecté par des défaillances ou par des évènements environnementaux. En effet, les accidents sont souvent le résultat de multiples séquences de défaillance qui impliquent le matériel, les logiciels et les erreurs humaines. De plus, les problèmes de sécurité deviennent importants lorsque ces applications comprennent des systèmes avec de graves séquences de défaillance, ce qui présente un danger pour la vie humaine et des biens.

Par conséquent, afin d'assurer un système sûr et garantir la sécurité humaine, on propose de respecter la spécification de contrôle suivante (*CGEM*) qui diminue la probabilité de risque d'un danger (hasardeux) :

$$M(P_3) + M(P_{11}) \leq 1$$

En effet, la place P_3 modélise la situation où le train est entrain de passer la barrière ou le croisement, à ce moment-là, la place P_{11} (barrière levée) ne doit pas être marquée. Notre objectif principal est la synthèse d'un système sécurisé et l'élimination de tout risque pouvant menacer la vie humaine. Le RdP contrôlé doit donc satisfaire la *CGEM* et garantir un comportement souhaité.

3.2.2.1 Détermination de la forme canonique de $\tilde{M}I$

A partir de la *CGEM*, on peut déduire que $w = (0,0,1,0,0,0,0,0,0,1,0)^t$ et $K = 1$. L'état interdit à concevoir est aussi un état canonique $\tilde{M}_1 \in \tilde{M}I$. \tilde{M}_1 a la forme canonique montrée (en rouge) dans la figure 3.2 suivante :

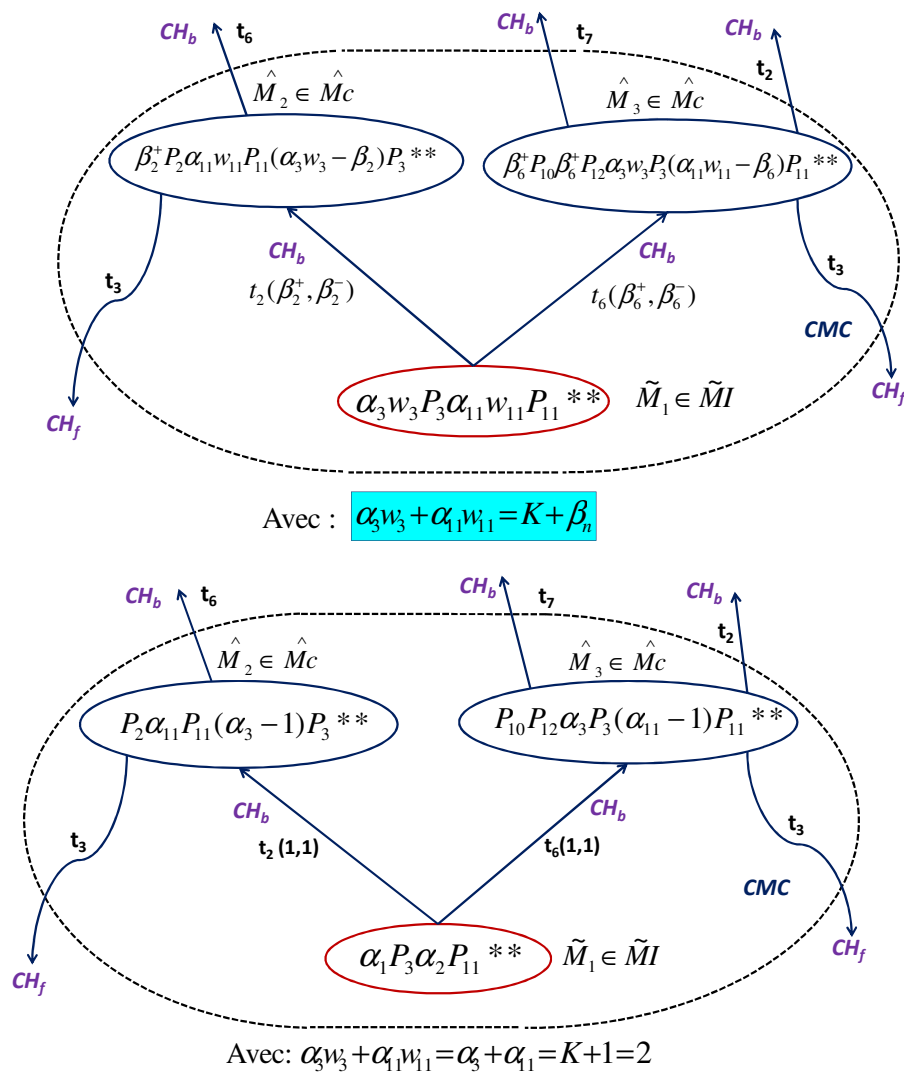


Figure 3.2- Synthèse des marquages canoniques

Proposition 3.1 :

Dans un RdP ordinaire, l'addition ou la soustraction de β^+ , β^- jetons des places p_i est limitée à 1 c.à.d. $\beta^+ = \beta^- = 1$ avec $\beta_i^- = B(p, t_i) | p \in t_i^{(p)}$ et $\beta_i^+ = B(p, t_i) | p \in {}^{(p)}t_i$. Par conséquent, l'état interdit $\widetilde{M}_1 \in \widetilde{MI}$ sur lequel est basée la conception des marquages canoniques est écrit comme suit : $\widetilde{M}_1 = \sum_{l,j} (\alpha_l w_l) \cdot p_j$ avec $\sum_{l,j} \alpha_l w_l = K + \beta = K + 1$.

Preuve :

Soit (w, K) est une spécification de contrôle donnée, $\alpha_L \in \mathbb{N}$, $\beta_i^- = B(p, t_i) | p \in t_i^{(p)}$, et $\beta_i^+ = B(p, t_i) | p \in {}^{(p)}t_i$. Si P est l'ensemble des places de T est l'ensemble des transitions tel que $P \cap T = \emptyset$, alors pour un RdP ordinaire : $C^+ : P * T \rightarrow \{0, 1\}$ est la matrice d'incidence avant du RdP de telle sorte que sur un arc connecté p_i à t_j alors $C^+(p_i, t_j) = 1$, sinon $C^+(p_i, t_j) = 0$. De plus, on a $C^- : P * T \rightarrow \{0, 1\}$ la matrice d'incidence arrière telle que si un arc connecté t_j à p_i , alors $C^-(p_i, t_j) = 1$, sinon $C^-(p_i, t_j) = 0$. Par conséquent, nous avons $B(p_i, t_j) | p_i \in t_j^{(p)} = B(p_i, t_j) | p_i \in {}^{(p)}t_j = \beta^- = \beta^+ = \beta = 1 \Rightarrow \sum_{l,j} \alpha_l w_l = K + \beta = K + 1$. En effet, en effectuant un chainage arrière (CH_b), cette condition garantit que nous ne pouvons plus trouver d'autres états interdits dans CMC et assure que tous les types d'instances de séparation d'évènements Ω sont inclus dans CMC .

3.2.2.2 Détermination des marquages canoniques

L'application de l'algorithme 3.1 génère l'ensemble des marquages canoniques $\widehat{M} \in \widehat{MC}$ qui ont la forme générique des pseudo-états qui peuvent exister dans le graphe de marquage $G(N, M_0)$ du réseau de Petri.

En effet, on effectue des chainages arrière/avant (CH_b et CH_f) à partir de $\widetilde{M}_1 \in \widetilde{MI}$ avec $\widetilde{M}_1 = \sum_{l,j} (\alpha_l w_l) p_j$. Les coefficients α_l et w_l de la place p_j représentent respectivement le marquage de la place et la composante du vecteur \vec{w}^T de la $CGEM(w, k)$ avec la condition suivante : $\sum_l \alpha_l w_l = K + \beta_n$. K est la constante de la spécification $CGEM$ tandis que β_n représente le poids de chaque transition franchie durant le chainage. β_n est égal à 1 si le RdP est ordinaire, sinon $\beta_n = \beta_i$ ce qui correspond au chainage arrière effectué à partir de \widetilde{MI} .

Cette condition minimise les coefficients des marquages de p_j . Après avoir effectué tous les chainages arrière/avant nécessaires, nous obtenons la coupe minimale canonique sur laquelle s'applique la théorie des régions.

3.2.2.3 Application de la théorie des régions

Une fois que les marquages canoniques et la coupe minimale canonique sont déterminés, la théorie des régions est appliquée pour la synthèse de contrôle en utilisant l'algorithme 3.2. La place de contrôle générée est ajoutée au RdP initial (voir figure 3.3) :

$$P_c: M_0(P_c) = 1; C(P_c, \cdot) = (0, -2, 0, 0, 0, -2, 0)$$

On remarque dans cet exemple que le contrôleur RdP est synthétisé sans utilisation de $G(N, M_0)$. La coupe minimale canonique présente une forme générique de toutes les coupes minimales qui peuvent exister dans le graphe de marquage si on utilise les approches développées dans le chapitre 2 [RARK14], [RAR14] et [RARK15].

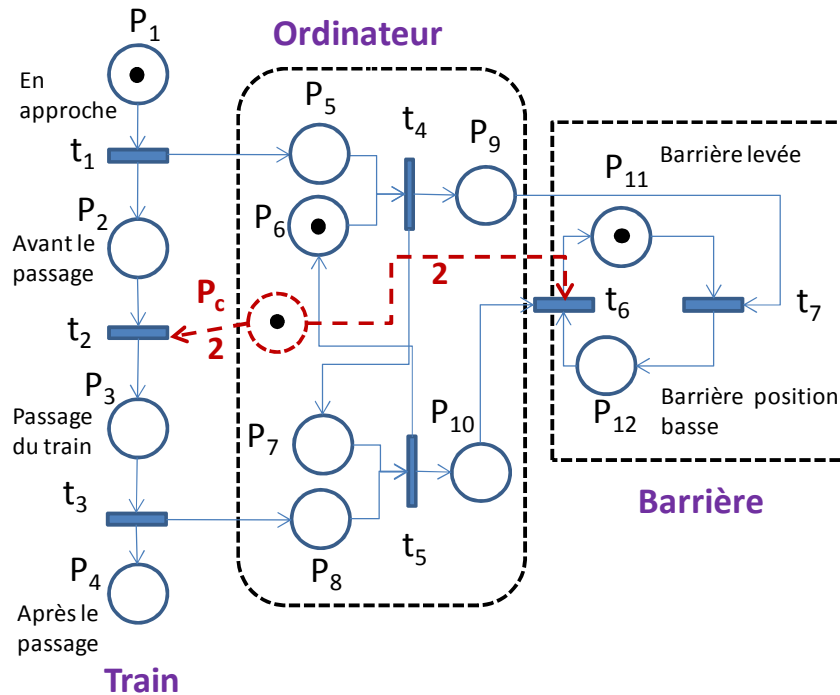


Figure 3.3- Modèle RdP contrôlé

En utilisant les propriétés du chapitre 2, le contrôleur RdP synthétisé s'active et se désactive en fonction de l'état dans lequel se trouve le procédé dans $G(N, M_0)$. Dans la partie qui suit, nous procédons à la reconnaissance de la coupe minimale canonique et des instances de séparation d'évènements dans $G(N, M_0)$ afin d'activer et désactiver le contrôleur RdP.

3.2.3 Identification des instances de séparation d'évènements dans $G(N, M_0)$

Une fois le réseau de Petri contrôlé, nous pouvons garantir le comportement maximum permissif de notre système. En effet, en identifiant les instances de séparation d'évènements dans le graphe de marquage, le contrôleur ajouté au RdP initial permet de garantir le comportement souhaité qui respecte la spécification de contrôle.

3.2.3.1 Les vecteurs d'activation et de désactivation

En s'inspirant des approches développées dans le deuxième chapitre, [RARK14][RAR14] et [RARK15], nous calculons les vecteurs d'activation et de désactivation du superviseur RdP synthétisé à partir de la coupe minimale canonique CMC .

La place de contrôle aura la tâche d'inhiber tous les états interdits présents dans $G(N, M_0)$. Ainsi, le contrôleur RdP sera initialement inactif (actif) jusqu'à ce qu'un marquage entre (sorte) dans la coupe minimale canonique. L'activation/désactivation du superviseur RdP est réalisée par le calcul des vecteurs $\vec{\sigma}_A(M)$ et $\vec{\sigma}_D(M)$ tels que [RARK15] :

$$\begin{aligned}\vec{\sigma}_A(M) &= \sigma \in T^* \mid \exists M_0[\sigma > M, M \in \Omega = M[t > M'] \\ \vec{\sigma}_D(M) &= \sigma \in T^* \mid \exists M_0[\sigma > M \wedge M[t >, M \in \Omega, t \notin \Omega.\end{aligned}$$

Remarque 3.2:

Le calcul des vecteurs d'activation/désactivation ($\vec{\sigma}_A(M)$ et $\vec{\sigma}_D(M)$) nécessite l'élimination des éventuels cycles qui peuvent réactiver (re-désactiver) le contrôleur P_{ci} .

3.2.3.2 Calcul du comportement maximum permissif

Après avoir identifié les instances de séparation Ω qui appartiennent à la coupe minimale canonique (CMC), nous pouvons déterminer le comportement maximum permissif désiré. L'identification des instances de séparation et le calcul du comportement souhaité sont décrits dans l'algorithme 3.3:

Algorithme 3.3 : Identification des CMC et calcul du comportement souhaité.

Soit $G(N, M_0)$ le graphe de marquage d'un RdP donné, (w, K) la spécification de contrôle exprimée sous de CGEM et $n = card(\Omega)$.

- 1 Initialisation $i := 1$
 - 2 Déterminer les ensembles des états et des transitions d'états respectivement MI et T_f qui ne respectent pas (w, K) .
 - 3 Identifier les instances de séparation:

$$\Omega = \{(M[t > M'), M \in G(N, M_0), t \in T_f, M' \in MI\}$$
 - 4 Calculer les vecteurs d'activation/désactivation
 - 5 Si $n \geq 1$
 Activer pour chaque $\Omega_i \in CMC$ le contrôleur RdP
 - 6 Eliminer les états/transitions d'états interdits ne respectant pas la spécification (w, K) .
 - 7 Tant que $i < n$
 $i := i + 1$
 Retourner à l'étape 5
- FIN

Dans les premières étapes, nous identifions les états et les transitions d'état qui ne respectent pas la CGEM. Ainsi, on pourra déterminer les instances de séparation. Ensuite, pour chaque $\Omega_i \in \Omega$ qui appartient à CMC et qui existe dans le graphe de marquage $G(N, M_0)$, on active/désactive notre contrôleur RdP afin d'avoir le comportement désiré du système contrôlé.

L'identification des CMC et l'obtention du comportement maximum permissif sont décrites dans les figures 3.4 et 3.5 :

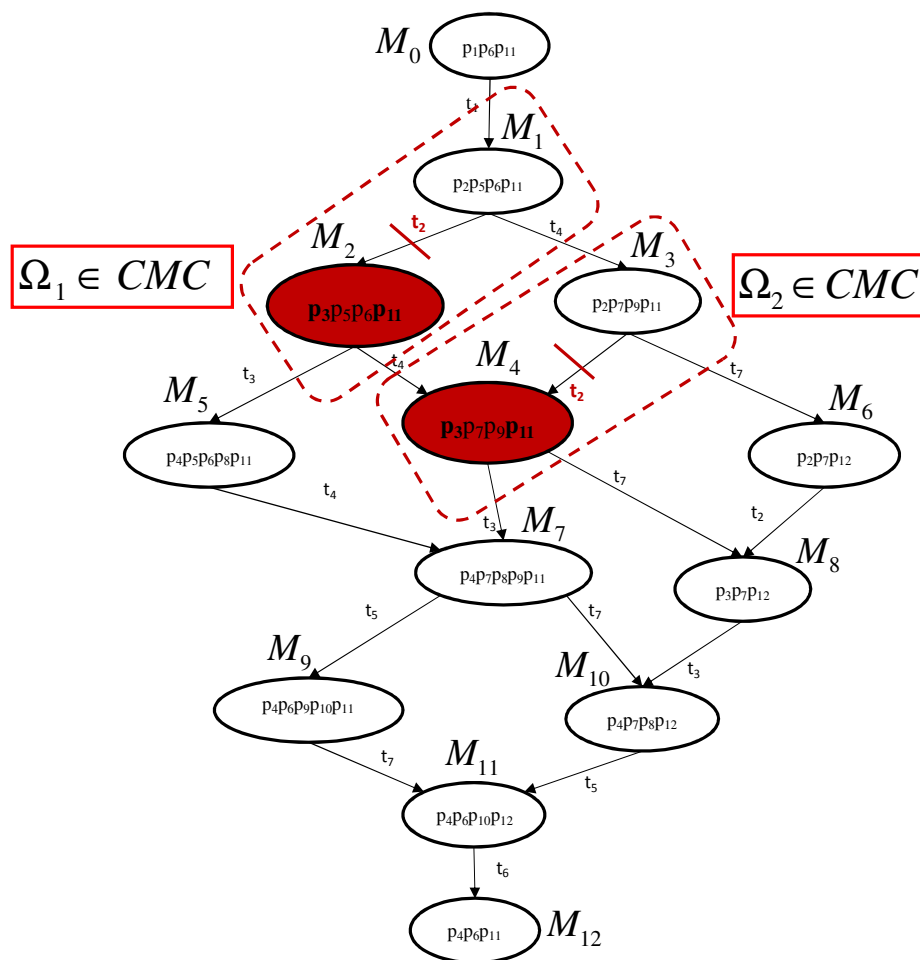


Figure 3.4- Identification des instances de séparation d'évènements

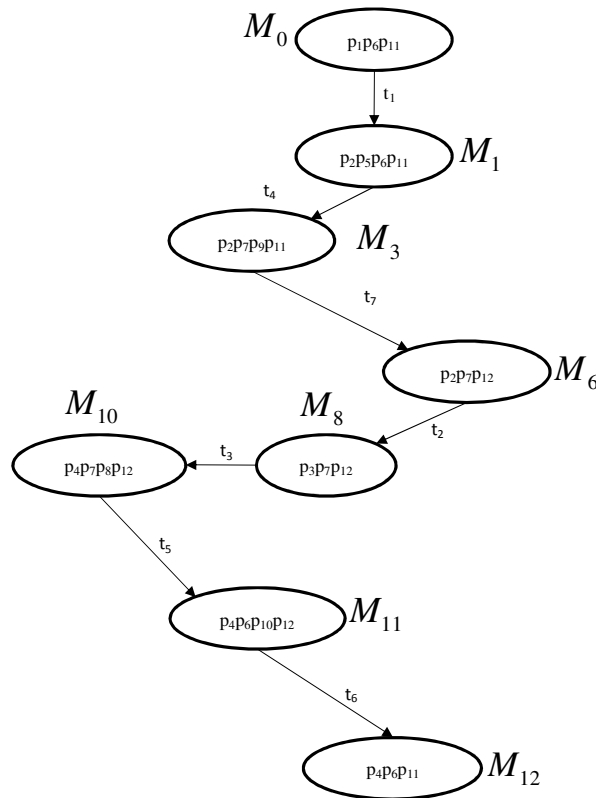


Figure 3.5- Calcul du comportement maximum permissif

3.3 Approche 2 : Synthèse de contrôleurs RdP basée sur les chemins admissibles dans un graphe de marquage

Nous définissons dans cette deuxième partie du chapitre un nouveau type de contrainte nommé contrainte de chemins admissibles (CCA), puis nous présentons notre approche de synthèse de la commande pour ce type de spécification. Contrairement à la première approche développée, cette dernière est basée sur l'utilisation du graphe de marquage $G(N, M_0)$.

Dans la littérature, il existe beaucoup de méthodes où les spécifications de contrôle s'expriment sous forme de combinaisons de marquages des places dans le graphe. Cependant, ces spécifications ne peuvent pas toujours satisfaire les exigences de contrôle demandées.

Dans cette présente approche, nous introduisons un nouveau type de contraintes qui se fonde sur des combinaisons de séquences de transitions dans le graphe de marquage. Autrement dit, dans ce travail, le graphe $G(N, M_0)$ sera déchiffré et lu comme un graphe de séquences de transitions. Cette nouvelle contrainte est appelée Contrainte de Chemins Admissibles (CCA).

3.3.1 Contrainte de chemins admissibles CCA

Considérons un ensemble de marquages admissibles, tels que ces marquages légaux correspondent à un ensemble de contraintes $CGEM(\vec{w}, K)$, ce qui définit l'ensemble d'états suivants :

$$M_{(\vec{w}, K)} = \{M \in M \mid \vec{w}^T M \leq k\}$$

Ce type de contraintes nous permet de traduire naturellement l'exclusion mutuelle pour l'utilisation de ressources partagées entre différents processus, ce qui est très répandu(s) dans les systèmes industriels.

Toutefois, les spécification de contrôle ne peuvent pas toutes être exprimées par des $CGEM$. Ainsi, à partir des définitions basiques des RdP et des perspectives générales, on introduit l'approche des chemins admissibles pour la synthèse de contrôle.

Définition 3.6 : [RRK14]

Soit $\langle N, M_0 \rangle$ un réseau avec un ensemble de séquences de transition. Une Contrainte de Chemin Admissible (CCA)(\vec{v}, C) définit un ensemble de séquences de transition admissibles :

$$T_{(\vec{v}, C)} = \{\sigma \in \sigma^* \mid \vec{v}^T \cdot \vec{\sigma} \leq C\}$$

\vec{v} est un vecteur d'entiers, et C est une constante entière. L'ensemble des séquences admissibles correspond à un ensemble de contraintes $CCA(V, \vec{C})$, avec $V = [\vec{v}_1 \dots \vec{v}_r]$ et $\vec{C} = (k_1 \dots k_r)$ donné par : $T_{(V, \vec{C})} = \bigcap_{i=1}^r T(\vec{v}_i, C_i)$.

Dans ce qui suit, nous allons mettre en œuvre l'utilité de la contrainte CCA pour certains exemples comme celui de la figure 3.6 :

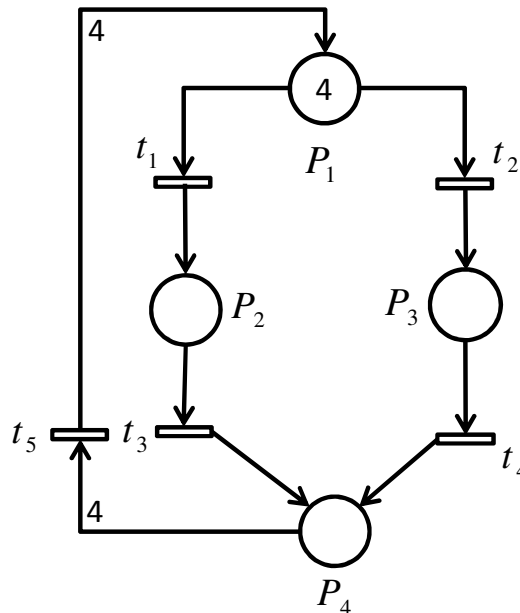


Figure 3.6- Un exemple RdP

Les places P_2 et P_3 modélisent deux machines différentes. Nous proposons d'implémenter la spécification de contrôle suivante dans cet exemple :

Le nombre de jetons dans la place P_2 doit être au minimum égal au double de celui de P_3 .

Nous souhaitons exprimer cette spécification de contrôle sous forme de *CGEM* [Giua96]:

$$\alpha M(P_2) + \beta M(P_3) \leq K$$

Le graphe de marquage $G(N, M_0)$ du RdP est présenté dans la figure 3.7 suivante.

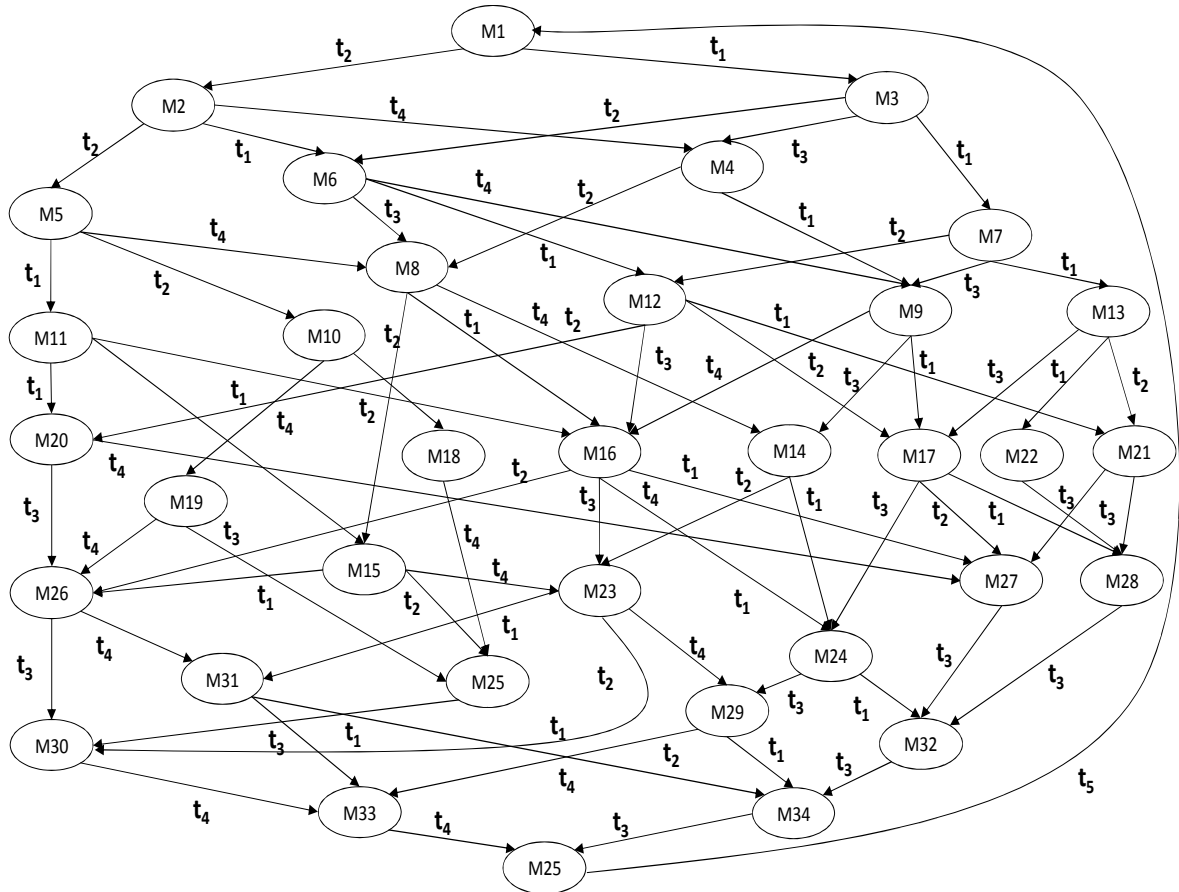


Figure 3.7- Graphe de marquage généré

On remarque que nous ne pouvons pas modéliser la spécification en utilisant des *CGEM*. En effet, la combinaison linéaire n'exprime pas le vrai sens de la spécification car cette dernière est basée sur le taux de production entre les deux machines.

En conséquence, nous devons traduire cette spécification en une combinaison de séquences de transition. Notre nouvelle approche résout les problèmes de transitions d'état interdites (*PTEI*). La contrainte de chemins admissibles *CCA* est donnée comme suit :

$$2\sigma[t_2] - \sigma[t_1] \leq 0$$

En appliquant la contrainte *CCA* sur le graphe de marquage $G(N, M_0)$, on obtient les chemins admissibles présentés dans la figure 3.8.

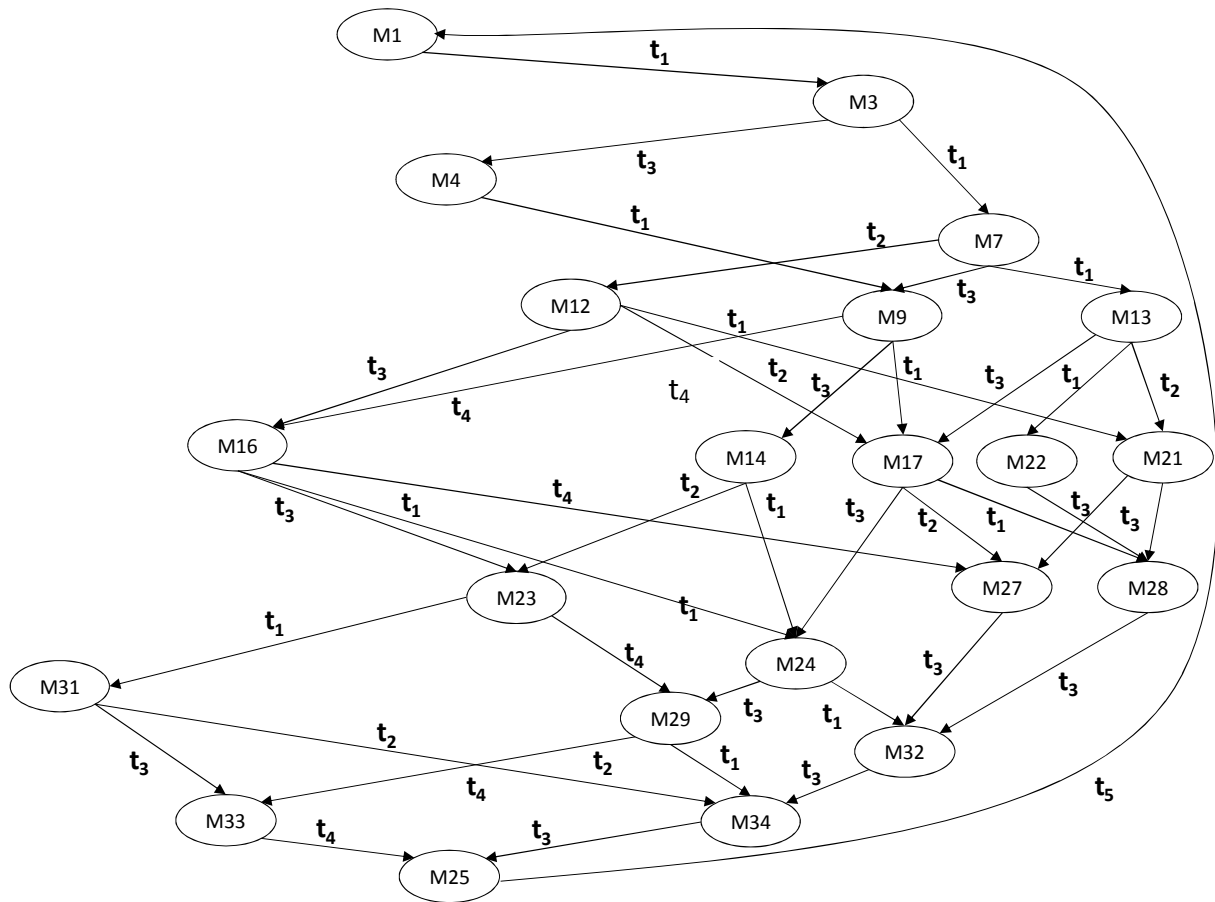


Figure 3.8- Chemins admissibles

Une fois la contrainte *CCA* appliquée, la théorie des régions nous permet de synthétiser le contrôleur RdP. L'algorithme 3.4 résume la présente approche.

Algorithme 3.4 : Soit un RdP $\langle N, M_0 \rangle$ et une spécification de contrôle donnée Q [RRK14] :

- 1 Générer le graphe de marquage $G(N, M_0)$
- 2 Déterminer l'ensemble des chemins interdits (FP)
- 3 Si $G(N, M_0)$ est fortement connexe alors aller à l'étape 7
Sinon aller à l'étape 4
- 4 Calculer les marquages fortement connexes (MFC) du graphe admissible (GA) qui contient M_0 .
- 5 Calculer $M_b = \{M \mid \exists \sigma \in T_u^*, M[\sigma > M' \wedge M \in MFC \wedge M' \notin MFC]\}$
- 6 Eliminer tous les marquages M_b du graphe et aller à l'étape 3
- 7 Soit GA le graphe admissible qui présente le comportement souhaité, M_L les marquages légaux dans le graphe, et W l'ensemble des instances de séparation (M, t) où $M \in GA, M[\sigma > M'$ et $M' \notin GA$.
- 8 Résoudre à plusieurs reprises le système linéaire de la théorie des régions pour chaque évènement interdit et soit $(M_0(P_c); C(P_c, \cdot))$ la solution correspondante.
- 9 Eliminer les places de contrôles redondantes
FIN

Plusieurs Outils informatique permettent de générer le graphe de marquage à partir d'un réseau de Petri. Pour cette partie, nous avons choisi d'utiliser Design/CPN [DES] qui est un logiciel de simulation et d'analyse des RdP colorés.

La première étape consiste à générer le graphe de marquage $G(N, M_0)$. En effet, le logiciel permet de spécifier une contrainte d'arrêt de la génération du graphe. Si un marquage vérifie la condition en question, ses successeurs ne sont pas calculés. Ainsi, le graphe admissible (GA) obtenu définit le comportement légal souhaité.

L'application de la théorie des régions via l'algorithme 3.4 génère une place de contrôle définie comme suit :

$$M_0(P_{c1}) = 0; C(P_{c1}, \cdot) = (1, -2, 0, 0)$$

Le contrôleur synthétisé garantit le comportement désiré qui satisfait la spécification de contrôle énoncée (CCA).

La figure 3.9 représente le réseau de Petri contrôlé de cet exemple.

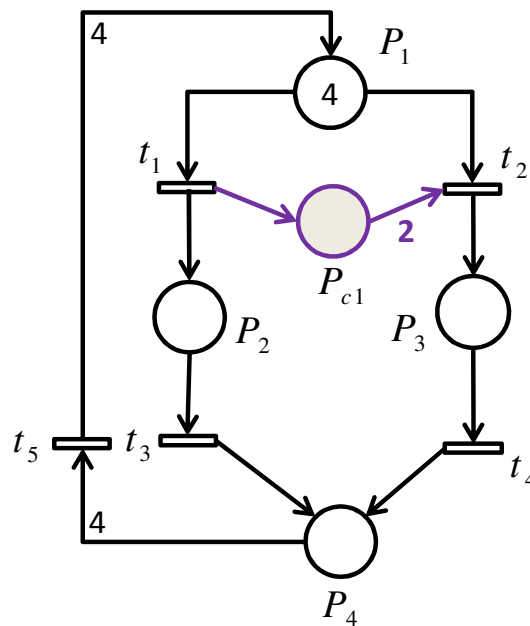


Figure 3.9- Modèle RdP contrôlé

3.4 Application sur un système de production flexible

Nous illustrons par cet exemple l'application de la méthode de synthèse de la commande présentée dans la première partie du chapitre. Considérons un système physique présenté par deux stations d'un atelier flexible et son modèle RdP sont présentés dans les figures 3.10 et 3.11.

Comme mentionné dans le chapitre 2, ces deux stations assurent la tâche d'assemblage des pièces en verre déjà gravées avec des socles de différentes couleurs. On propose de satisfaire la *CGEM* suivante : $M(P_2) + 2M(P_3) \leq 3$.

Le scénario du chapitre 2 est adapté dans cet exemple ; la place P_2 représente notre production et P_3 modélise la machine d'un sous-traitant. Chaque machine a une capacité de trois ($C_M = C_S = 3$). Les événements t_1, t_2, t_4 et t_5 modélisent l'entrée et la sortie des produits dans chaque machine. Sinon, la palette continue son chemin via la transition t_3 sur le convoyeur principal.



Figure 3.10- Système physique

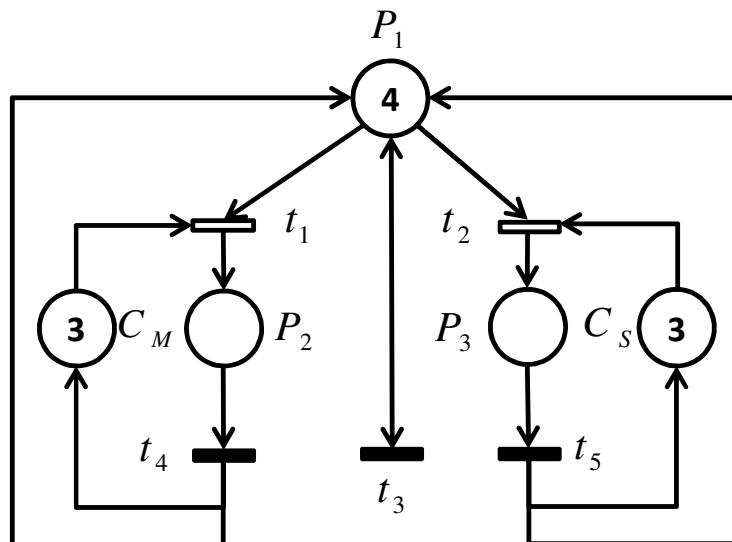


Figure 3.11- Modèle RdP

Application de l'algorithme 3.1 :

Les données qu'on possède jusqu'à présent sont : le modèle RdP et la spécification de contrôle exprimée sous forme de *CGEM*. L'application de l'algorithme 3.1 de conception des

marquages canoniques est donnée dans la figure 3.12. En effet, la forme généralisée de la coupe minimale canonique *CMC* est décrite dans cette figure.

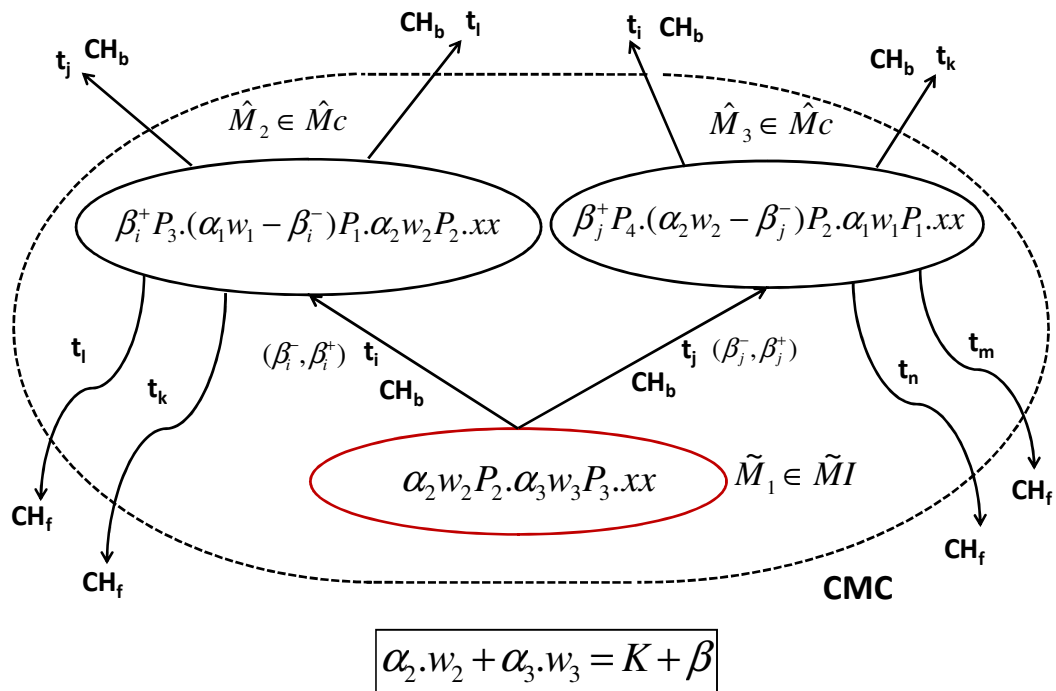


Figure 3.12- La forme généralisée de *CMC*

Le réseau de Petri utilisé pour modéliser le système de production flexible est ordinaire, donc conforme à la proposition 3.1 : $\beta = \beta^+ = \beta^- = 1$. En outre, à partir de la *CGEM*, on déduit que $K = 3$ et $w = (P_1, P_2, P_3, C_M, C_S)^t = (0, 1, 2, 0, 0)^t \Rightarrow w_2 = 1, w_3 = 2$. Par conséquent, les marquages canoniques synthétisés sont décrits dans la figure 3.13 suivante :

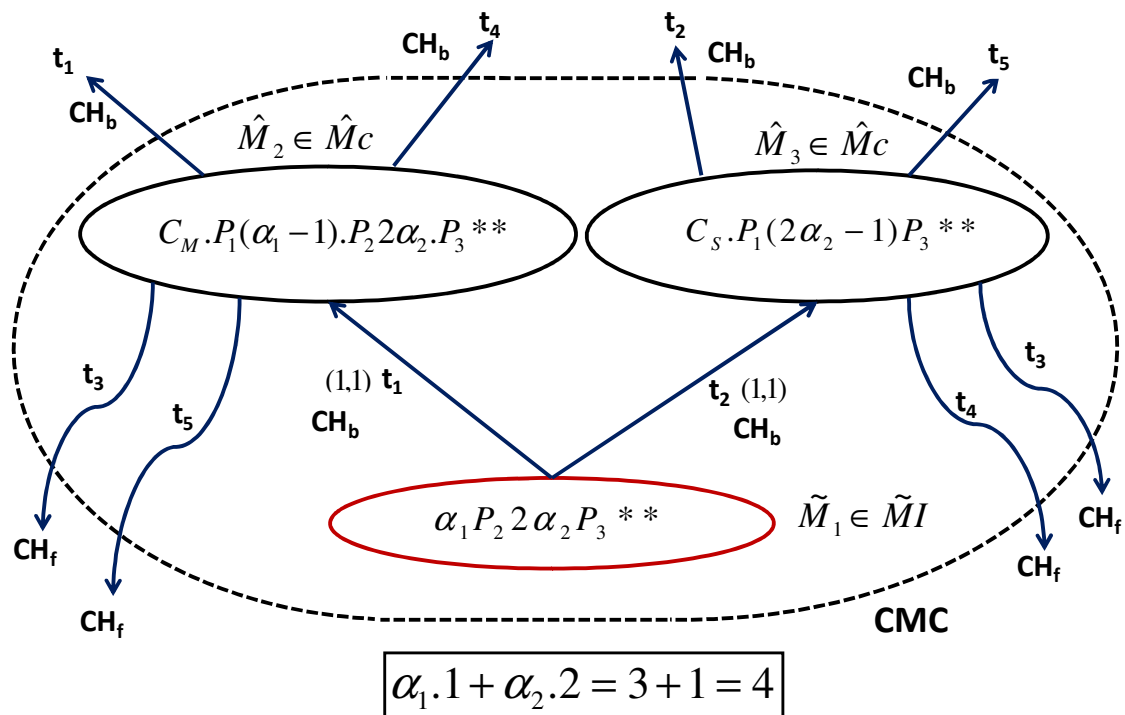


Figure 3.13- Marquages canoniques

L'application de l'algorithme 3.1 génère l'ensemble des marquages canoniques $\hat{M} \in \hat{M}c$ qui constituent des pseudo-états et qui forment une coupe minimale canonique dans laquelle on trouve tous les types d'instances de séparation d'événements appartenant à l'ensemble Ω .

En effet, à partir de la spécification de contrôle (CGEM) et le réseau de Petri, nous concevons initialement l'état interdit $\tilde{M} \in \tilde{M}I$ comme suit : $\tilde{M} = \sum_{l,j} \alpha_l w_l P_j$. Les coefficients α_l et w_l de la place P_j représentent respectivement : le marquage de P_j et la composante du vecteur \vec{w}^T de la spécification (w, K) . Ces coefficients doivent remplir la condition suivante : $\sum_l \alpha_l w_l = K + \beta_n \cdot K$ est la constante de la CGEM et β_n est le poids pour chaque transition durant le chainage. β_n est égal à 1 dans cet exemple $\Rightarrow \sum_l \alpha_l w_l = K + \beta_n = \alpha_1 \cdot 1 + \alpha_2 \cdot 2 = 3 + 1 = 4$.

Ainsi, grâce aux chainages arrière/avant, nous pouvons déterminer tous les marquages canoniques et les transitions correspondantes.

Application de l'algorithme 3.2 :

La coupe minimale canonique et les marquages canoniques substituent le graphe de marquage $G(N, M_0)$ pour la synthèse de contrôle et la conception du contrôleur RdP. L'application de la théorie des régions est décrite dans l'algorithme 3.2 et nous génère la place de contrôle suivante :

$$P_c: \begin{cases} M_0(P_c) = 1 \\ C(P_c, \cdot) = (0, -2, -2, 0, 0) \end{cases}$$

L'ajout de la place de contrôle au RdP initial (figure 3.11) nous donne le RdP contrôlé suivant :

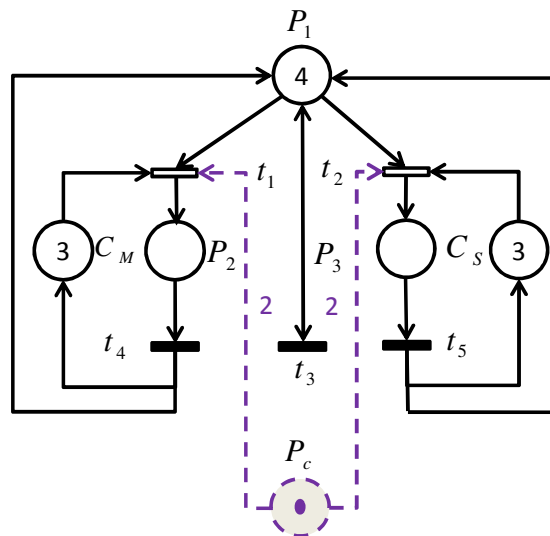


Figure 3.14- Modèle RdP contrôlé

Conformément à notre politique de contrôle, nous obtenons un contrôleur RdP qui permet d'interdire toutes les instances de séparation d'évènements. Ce superviseur RdP sera activé/désactivé en fonction de l'état/marquage dans lequel se trouve le processus dans $G(N, M_0)$. Dans ce qui suit, nous identifions les instances de séparation d'évènements dans le graphe de marquage afin d'activer/désactiver le contrôleur RdP (voir la figure 3.15).

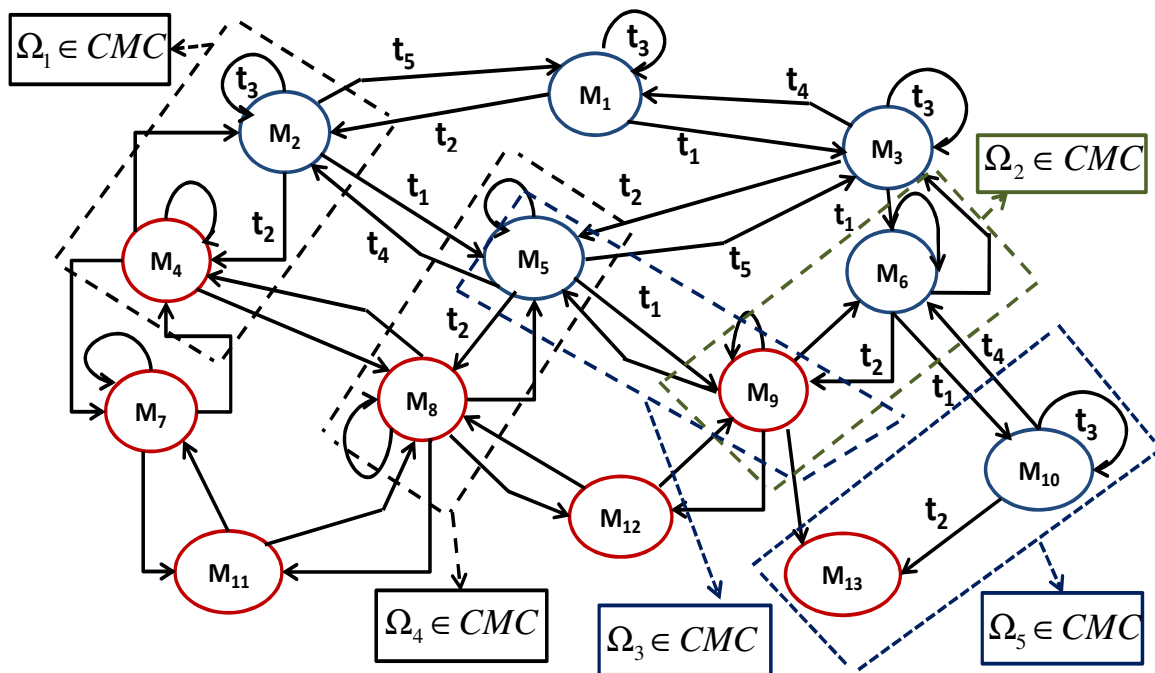


Figure 3.15- Identification des instances de séparation d'évènements

En outre, pour activer/désactiver le contrôleur RdP, il est nécessaire de calculer les vecteurs d'activation et de désactivation des états $M_2 \in \Omega_1, M_6 \in \Omega_2, M_5 \in \Omega_3 \cup \Omega_4$ et $M_{10} \in \Omega_5$. Les informations détaillées de ces vecteurs sont données dans le tableau 3.1:

Tableau 3.1- Vecteurs d'activation et de désactivation

Etat	Vecteur d'activation $[t_1, t_2, t_3, t_4, t_5]^T$	Vecteur de désactivation $[t_1, t_2, t_3, t_4, t_5]^T$
M_2	$[0,1,0,0,0]^T$	$[1,1,0,0,0]^T$
M_6	$[2,0,0,0,0]^T$	$[3,0,0,0,0]^T$
M_5	$[1,1,0,0,0]^T$	$[1,1,0,0,1]^T$
M_{10}	$[3,0,0,0,0]^T$	$[3,0,0,1,0]^T$

L'activation et la désactivation du contrôleur RdP nous permettent de calculer le comportement maximum permissif souhaité qui satisfait la spécification de contrôle (*CGEM*). Ce comportement est donné dans la figure 3.16:

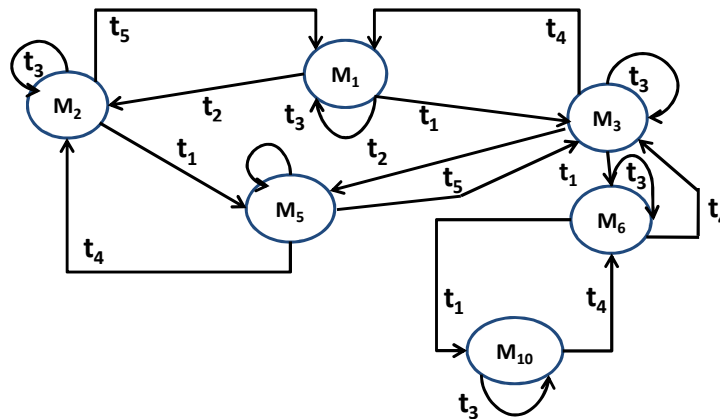


Figure 3.16- Calcul du comportement maximum permissif

3.5 Conclusion

Nous proposons dans ce chapitre une extension de la méthode exposée au chapitre précédent. En effet, deux approches de synthèse de contrôleur RdP ont été présentées dans ce chapitre. Elles se basent sur la théorie des régions et nous permettent de trouver une solution optimale aux problèmes d'état interdit et transitions d'état interdites.

Dans la première approche, la synthèse de contrôle s'appuie sur les marquages canoniques. Cet ensemble d'états \widehat{MC} forme une zone de marquages appelée coupe minimale canonique *CMC*, qui substitue le graphe de marquage $G(N, M_0)$ pour la synthèse de contrôle, c.à.d. que la théorie des régions s'applique sur *CMC* et non pas sur $G(N, M_0)$. En effet, à partir de la spécification de contrôle exprimée sous forme de *CGEM* et les caractéristiques du modèle RdP, on parvient à déterminer une forme générique de l'état interdit \widetilde{MI} . A partir de cet état primitif, on effectue des chainages arrière/avant pour déterminer les autres états de la coupe

minimale canonique. Ces états présentent une forme générique des états et des instances qui existent dans le graphe $G(N, M_0)$. Une fois le contrôleur RdP synthétisé, on utilise $G(N, M_0)$ pour la reconnaissance des événements interdits, l'activation/désactivation du superviseur RdP et le calcul du comportement maximum permissif désiré.

De surcroît, la deuxième approche développée présente un nouveau type de contraintes qui permettent de mieux satisfaire et exprimer une spécification de contrôle (Q) donnée. En effet, certaines spécifications de contrôle, qui se fondent sur le taux de production des systèmes manufacturiers par exemple, ne s'expriment pas sous forme de $CGEM$, c.à.d. que la combinaison entre les marquages de place n'arrive pas à traduire le vrai sens de la spécification. C'est pour cette raison qu'on a développé un nouveau type de spécification de contrôle qui se base sur des séquences de transitions et non pas sur des marquages de places.

Le travail de recherche accompli dans ce chapitre nous ouvre encore une autre extension et un nouveau champ de recherche pour réduire la complexité de calcul des superviseurs RdP. Dans le chapitre suivant, nous proposons de nous déposséder définitivement du graphe de marquage $G(N, M_0)$ pour réaliser le comportement désiré tout en respectant les règles des réseaux de Petri.

Chapitre 4

Synthèse des contrôleurs RdP sans génération du graphe de marquage

***Résumé:** L'objectif du travail réalisé dans ce chapitre, est de réduire au maximum la complexité de calcul des contrôleurs RdP générés en utilisant la théorie des régions. En effet, l'explosion combinatoire est une des limites des méthodes de synthèse basées sur la génération des graphes de marquages. L'explosion combinatoire rend la détermination du superviseur assez complexe. Pour contourner ce problème, nous proposons dans ce chapitre deux approches originales de synthèse de la commande pour les systèmes à événements basées sur la théorie des régions sans la génération du graphe de marquage.*

4.1 Introduction

La théorie des régions nouvellement interprétée dans la synthèse de contrôle [GRX02a][GRX02b] a été d'un grand apport pour résoudre les problèmes d'états interdits et de transitions d'état interdites. Cette théorie est reconnue comme l'une des méthodes les plus efficaces dans la synthèse de la commande et de prévention de blocage afin d'obtenir un contrôleur RdP maximum permissif [GRX02c].

Malgré les différents travaux menés dans la recherche des méthodes de réduction de calcul du superviseur RdP, la théorie des régions reste complexe à résoudre si le graphe de marquage $G(N, M_0)$ est très large [UZ07][Uzam09][LZJ08][HP11][QLZKM12].

Nous proposons dans ce travail de développer des approches de synthèse de contrôle qui nous permettent de nous déposséder complètement du graphe de marquage, afin de calculer le superviseur RdP. En effet, les méthodes d'exploration du graphe et la recherche du comportement maximum permissif, qui satisfait les spécifications de contrôle, suscitent un grand intérêt pour les chercheurs. Les méthodes définies dans ce chapitre se basent, d'une part, sur des notions mathématiques et algébriques, et d'autre part, sur les propriétés des réseaux de Petri.

Ce travail est basé sur l'utilisation de nouvelles approches qui se fondent sur les concepts de la théorie des régions sans générer le graphe. L'idée clé de notre politique de contrôle est la transformation de notre problématique en un exercice mathématique qui se résout en se basant sur les propriétés des RdP. Les résultats de ce chapitre démontrent l'enjeu important de nos méthodes qui révèlent un grand avantage dans le domaine de synthèse de la commande.

Dans ce chapitre, nous présentons dans les sections 4.2 et 4.3 les deux approches de synthèse de la commande sans utiliser le graphe de marquage. Les deux approches sont détaillées et illustrées par des exemples didactiques. Enfin, dans la section 4.4, nous clôturons ce chapitre par une partie expérimentale dans laquelle nous implémentons des contrôleurs RdP sur les automates programmables d'un système de production flexible.

4.2 Approche 1 : Théorie des régions sans graphe de marquage

Dans cette partie, une méthode efficace de synthèse de contrôle basée sur la théorie des régions et sans utilisation du graphe de marquage est introduite. En effet, la génération du graphe et son exploration afin d'identifier les états interdits/admissibles et les cycles de base, représente la partie la plus fastidieuse lors de la synthèse de la commande. L'objectif principal de ce travail est la réduction de la complexité de calcul des contrôleurs RdP. Nous définissons dans la suite deux outils mathématique nécessaires à la compréhension de la démarche.

Définition 4.1 : (Combinaison sans remise)

Une combinaison k sans remise d'un ensemble A à n éléments est un sous-ensemble arbitraire de A ayant k éléments. Si nous prenons k objets parmi n sans remise ou sans répétition et indépendamment de l'ordre d'apparition, nous pouvons représenter ces k éléments par une partie des k éléments à partir d'un ensemble de n éléments. Pour déterminer le nombre de ces

dispositions, nous déterminons le nombre d'arrangements de k objets et nous le divisons par le nombre de dispositions obtenues entre elles par une permutation [Riordan 12] :

$$C_n^k = \binom{n}{k} = \frac{A_k^n}{k!} = \frac{n!}{k!(n-k)!}$$

Définition 4.2 : (Combinaison avec remise)

Une combinaison k avec remise d'un ensemble A à n éléments est un multi-ensemble de k éléments de A . Si nous prenons k objets parmi n sans remise et indépendamment de l'ordre d'apparition, ces objets peuvent apparaître plusieurs fois et nous ne pouvons pas les représenter, ni avec une partie de k éléments, ni avec un k -uplet, car leur ordre de placement n'est pas impliqué. Cependant, il est possible de représenter ces dispositions avec des combinaisons avec remise. Le nombre de k -combinaisons avec répétition d'un ensemble de n éléments est désigné par Γ_n^k , et formellement [Riordan 12]:

$$\Gamma_n^k = \binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!}$$

Nous rappelons que notre politique de contrôle est basée sur l'application de la théorie des régions qui se traduit dans notre travail par les conditions d'atteignabilité des états, les conditions de séparation d'évènements et les équations de cycles de base formant ainsi le système linéaire noté SL . Toutefois, dans l'approche classique de la théorie des régions développée par Ghaffari et al. [GRX02a, b], SL est obtenu en définissant les cycles de base, les états interdits, les états admissibles, les états dangereux et l'ensemble des instances de séparation d'évènements (Ω) [GRX03 a, b].

4.2.1 Procédé de synthèse de commande

En nous inspirant de nos travaux précédents présentés dans les chapitres 2 et 3, nous développons une méthode efficace pour la détermination du système linéaire SL de la théorie des régions en définissant les cycles de base, les conditions d'atteignabilité et de séparation d'évènements.

4.2.1.1 Détermination des équations de cycles de base

Soit un réseau de Petri $N = \langle P, T, Pre, Post \rangle$ de matrice d'incidence C et X_t un vecteur d'entiers positifs ou nuls, solution de l'équation $C \cdot X_t = 0$. Le t -sous-réseau de Petri généré par le support de X_t est une composante conservative positive.

$\Leftrightarrow X_t$ est un vecteur solution de $C \cdot X_t = 0$ et dont les composantes sont des nombres entiers positifs ou nuls.

Soit s une séquence de franchissement de transitions appelée invariant de transitions [KSTW10] :

$$C \cdot \begin{bmatrix} X_{t_1} \\ \vdots \\ X_{t_n} \end{bmatrix} = 0$$

La résolution de ce système d'équations génère les invariants de transitions qui nous permettent de déterminer les équations de cycles de base. Soit T_{cycle} l'ensemble des transitions existantes dans les équations de cycles.

4.2.1.2 Détermination des conditions d'atteignabilité et de séparation

Une fois les équations de cycles exprimées, il ne reste qu'à déterminer les conditions d'atteignabilité et les conditions de séparation pour définir le système linéaire de la théorie des régions sans avoir recours au graphe de marquage $G(N, M_0)$. Les spécifications de contrôle sont exprimées dans ce travail sous forme de contraintes généralisées d'exclusion mutuelle (CGEM) : $M_{(\vec{w}, k)} = \{M \in M | \vec{w}^T \leq k\}$.

A partir de cette équation, on peut déterminer les marquages interdits MI qui ne satisfont pas la CGEM. Cependant, notre hypothèse de travail est la connaissance de ces états MI qui ne répondent pas aux exigences de contrôle.

Jusqu'à présent, pour synthétiser le superviseur RdP sans générer le graphe de marquage, les informations suivantes sont supposées connues : Le réseau de Petri du procédé, l'état initial M_0 , l'ensemble des états interdits MI , et l'ensemble T_{cycle} des transitions existantes dans les équations de cycles.

Remarque 4.1 :

Déterminer les conditions d'atteignabilité et de séparation d'évènements revient à rechercher les chemins qui relient l'état initial M_0 et les autres états $M \in M$.

⇒ Pour déduire le chemin entre l'état initial M_0 et un état M_i , il est nécessaire de déterminer la séquence de franchissement des transitions σ_i .

Pour identifier des conditions d'atteignabilité et les conditions de séparation d'évènements sans avoir recours au graphe de marquage, on utilise les méthodes d'analyse combinatoire.

Théorème 4.1 :

Pour un RdP sauf, le nombre possible de combinaisons des transitions .c.à.d. de séquences de tir des transitions, correspond mathématiquement à une combinaison non ordonnée sans

remise dite simple : $S = \sum_{i=0}^{card(T_{cycle})} C_{card(T_{cycle})}^i$.

Preuve :

Un RdP sauf est un RdP 1-borné, pour une séquence donnée chaque transition ne peut être tirée qu'une fois au maximum. De la même manière, une combinaison sans répétition est une collection de p objets pris simultanément parmi n , donc sans tenir compte de l'ordre d'apparition. Elle est dite simple si on ne peut prendre chaque objet qu'au plus une fois. Le nombre de combinaisons de i éléments parmi T_{cycle} est noté par $C_{card(T_{cycle})}^i$ tel que:

$$C_{card(T_{cycle})}^i = \frac{card(T_{cycle})!}{i!(card(T_{cycle})-i)!}$$

Théorème 4.2 :

Pour un RdP k -borné, le nombre possible de combinaisons des transitions non ordonnées, c.à.d. de séquences de tir des transitions, correspond mathématiquement à une combinaison avec remise: $S = \sum_{i=0}^{card(T_{cycle})} \Gamma_{card(T_{cycle})}^i$, tel que $\Gamma_{card(T_{cycle})}^i = \frac{(card(T_{cycle})+i-1)!}{i!(card(T_{cycle})-1)!}$.

Preuve:

Lorsque nous choisissons i transitions parmi T_{cycle} , nous obtenons un groupement non ordonné de T_{cycle} transitions éventuellement répétées: ce groupement n'est pas un ensemble, c.à.d. la définition en extension d'un ensemble empêchant la répétition des éléments, mais un multi-ensemble. Une S -combinaison avec répétition d'un ensemble fini E de cardinal T_{cycle} est une application f de E dans $\{0, 1 \dots S\}$ tel que $\sum_{x \in E} f(x) = S$ c.à.d. si $E = \{x_1, x_2 \dots x_{T_{cycle}}\}$ alors f vérifie :

$$f(x_1) + \dots + f(x_T) = S = \Gamma_{T_{cycle}}^0 + \Gamma_{T_{cycle}}^1 + \dots + \Gamma_{T_{cycle}}^{T_{cycle}} = \sum_{i=0}^{card(T_{cycle})} \Gamma_{card(T_{cycle})}^i.$$

Définition 4.3 : (séquence de franchissement σ_{MI_i} d'un marquage interdit)

Une séquence de franchissement σ_{MI_i} menant à un état interdit MI_i est un préfixe d'une séquence v d'un ensemble E d'évènements, s'il existe une autre séquence w de E telle que $v = \sigma_{MI_i}w$, c.à.d. v est la concaténation de σ_{MI_i} et w . Si v est une séquence physiquement possible dans le système, alors naturellement tout préfixe de v l'est également.

L'analyse combinatoire est fusionnée dans notre politique de contrôle pour déterminer les conditions d'atteignabilité et de séparation d'évènements. Nous nous sommes inspirés de nos travaux précédents [RARK15][RAR14] pour concevoir l'ensemble des états interdits MI à partir d'une $CGEM$. Les équations de cycles de la théorie des régions sont jusque-là identifiées, il nous reste donc à déterminer les conditions d'atteignabilité des états admissibles et les conditions de séparation d'évènements interdits.

L'idée clé de ce travail est de considérer l'ensemble des transitions existantes dans les équations de cycle de base T_{cycle} comme une urne à partir de laquelle on effectue différents tirages de transitions avec ou sans remise selon la nature du réseau de Petri. Chaque tirage de transitions correspond à une séquence de tir σ_i ou à un chemin reliant M_0 à un état M_i .

Une fois que l'ensemble des séquences (chemins) est déterminé, on le filtre grâce à l'équation fondamentale des RdP : $M = M_0 + C\vec{\sigma}$. En testant les différentes σ_i , le filtre comprend les séquences de franchissement suivantes : Les séquences σ_i ayant des préfixes σ_{MI_i} , les séquences σ_i nous donnant : des marquages nuls (marquage à composantes nulles), des marquages négatifs (état contenant au moins une composante négative).

Par conséquent, les séquences seront classées en deux catégories : les séquences utilisables σ_U et les séquences inutilisables σ_{UN} . Les séquences σ_U et les équations de cycle obtenues génèrent le système linéaire de la théorie des régions recherché.

L'algorithme 4.1 traduit la politique de contrôle.

Algorithme 4.1 : Synthèse de contrôle basée sur la théorie des régions sans graphe de marquage.

Soient un RdP donné avec son marquage initial M_0 , l'ensemble des états MI qui ne satisfont pas une $CGEM$ et S l'ensemble de séquences de franchissement des transitions.

- 1 Déterminer les équations de cycles de base en utilisant la méthode des invariants des transitions et soit T_{cycle} l'ensemble des transitions existantes dans les équations de cycles.
- 2 Si le RdP est sauf, alors déterminer S avec $|S| = \sum_{i=0}^{card(T_{cycle})} C_{card(T_{cycle})}^i$
Si le RdP est k -borné avec $k > 1$ alors déterminer S avec $|S| = \sum_{i=0}^{card(T_{cycle})} \Gamma_{card(T_{cycle})}^i$
- 3 Identifier à partir S l'ensemble des séquences de transitions interdites $\sigma_{MI} \mid M_0[\sigma_{MI} > M', M' \in MI$.
- 4 Filtrer l'ensemble des séquences S pour obtenir σ_U en éliminant: les séquences de franchissement menant à des marquages nuls et négatifs et les séquences ayant comme préfix σ_{MI} .
- 5 Tant que $\sigma_{MI} \neq \emptyset$ faire:
 - 5.1 Pour chaque séquence interdite $\sigma_i \in \sigma_{MI}$, écrire le système linéaire de la théorie des régions composé de:
 - La condition de séparation d'évènements de l'éventuel chemin σ_{MI_i}
 - Les conditions d'atteignabilité des séquences utilisables en éliminant les séquences interdites σ_U/σ_{MI} .
 - Les équations de cycles de base
 - 5.2 Résoudre le système linéaire SL_i et soit $(M_0(P_{ci}); C(P_{ci}, \cdot))$ la solution, si elle existe.
- 6 Eliminer les places de contrôle redondantes pour obtenir le modèle RdP contrôlé

Etape 1 : Calcul des équations de cycles de base :

Cette étape est très importante car elle nous permet de déterminer l'ensemble T_{cycle} des transitions qui existent dans les équations de cycle. Les séquences de transitions seront tirées à partir de T_{cycle} . Le tirage est fait exprès à partir de cet ensemble et non pas de la totalité des transitions du réseau de Petri, car on est sûr que les cycles existent dans le comportement désiré du système.

Etape 2 : Détermination de l'ensemble des séquences de franchissement des transitions

Pour faire face à notre problématique, nous effectuons des combinaisons des transitions appartenant à T_{cycle} selon la nature du RdP. Le problème devient un exercice mathématique où nous avons une urne contenant un nombre fini ($card(T_{cycle})$) de transitions étiquetées. Le tirage avec ou sans remise de j transitions avec $0 \leq j \leq card(T_{cycle})$, nous forme une combinaison de transitions qui correspond à un chemin reliant M_0 à un état M . L'ensemble de tous les tirages est noté S .

Etape 3 : Identification des séquences interdites σ_{MI}

Puisque nous connaissons l'ensemble des états interdits MI , nous pouvons identifier l'ensemble de toutes les séquences interdites σ_{MI} . En outre, la détermination de cet ensemble peut se faire par deux méthodes :

- Méthode algébrique : En utilisant l'équation d'état $MI_i = M_0 + C \cdot \sigma_{MI_i}$ on peut identifier la séquence σ_{MI_i} menant à un état interdit. $\sigma_{MI} = (MI - M_0) \cdot C^{-1}$. En effet, cette méthode est valide si $\det(C) \neq 0$.
- Méthode canonique [RAR15] : En effectuant des chainages arrière CH_b et des chainages avant CH_f à partir de l'état interdit MI_i , on peut identifier l'état initial M_0 et obtenir la séquence qui relie M_0 à MI_i .

Etape 4 : Filtrage des séquences de franchissement des transitions

L'ensemble des séquences S est filtré pour obtenir les séquences utilisables σ_U . Les séquences inutilisables σ_{UN} à rejeter sont :

- Les séquences qui ont σ_{MI_i} comme préfixe
- Les séquences menant à des marquages nuls (toutes les composantes sont nulles) en utilisant l'équation fondamentale des RdP.
- Les séquences menant à des marquages négatifs (il existe au moins une composante négative) en utilisant l'équation fondamentale des RdP.

Etape 5 : Application de la théorie des régions

Sachant que $\sigma_{MI} \subset \sigma_U$, les séquences utilisables et les équations de cycle de base forment le système linéaire de la théorie des régions. En effet, pour chaque séquence interdite σ_{MI_i} qui représente la condition de séparation d'évènement, nous ajoutons les équations de cycle de base et les conditions d'atteignabilité à partir des chemins σ_U/σ_{MI} (l'ensemble des séquences utilisables moins les séquences interdites). Ainsi, la résolution de chaque système linéaire SL_i génère la place de contrôle correspondante P_{ci} .

Etape 6 : Elimination des places de contrôle redondantes

En appliquant la théorie des régions pour chaque instance de séparation d'évènement, c.à.d. pour chaque séquence interdite σ_{MI_i} , on obtient des places de contrôle redondantes. Leur élimination nous donne le modèle RdP contrôlé.

4.2.2 Exemple illustratif

Considérons le modèle RdP donné dans la figure 4.1. Le réseau modélise deux machines qui partagent deux ressources R_1 et R_2 . Pour traiter le problème de synchronisation et d'organisation entre les deux processus qui fonctionnent en parallèle, nous voulons implémenter la spécification de contrôle suivante (CGEM) : $M(P_1) + M(P_3) \leq 1$.

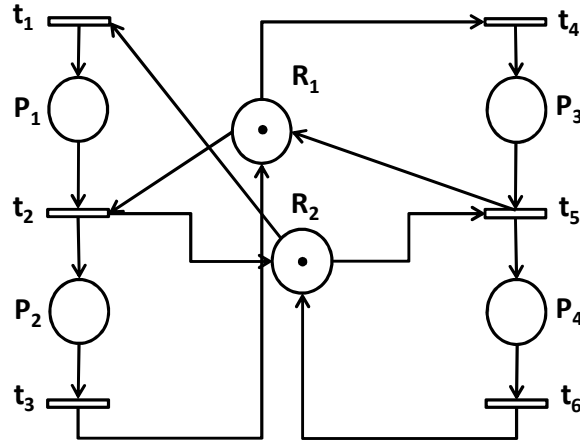


Figure 4.1- Exemple Illustratif

D'après l'approche proposée, nous devons d'abord déterminer les équations de cycles de base en utilisant la méthode des invariants de transitions. La matrice d'incidence du RdP est comme suit :

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

En outre, le système $C \cdot X_t = 0$ avec $X_t^T = [X_t(t_1)X_t(t_2)X_t(t_3)X_t(t_4)X_t(t_5)X_t(t_6)]^T$ est :

$$\Rightarrow \begin{cases} X_t(t_1) - X_t(t_2) = 0 \\ X_t(t_2) - X_t(t_3) = 0 \\ X_t(t_4) - X_t(t_5) = 0 \\ X_t(t_5) - X_t(t_6) = 0 \\ -X_t(t_2) + X_t(t_3) - X_t(t_4) + X_t(t_5) = 0 \\ -X_t(t_1) + X_t(t_2) - X_t(t_5) + X_t(t_6) = 0 \end{cases}$$

De ce fait, deux solutions du système sont trouvées :

$$\begin{aligned} (X_t^1)^T &= [0 \ 0 \ 0 \ 1 \ 1 \ 1]^T \\ (X_t^2)^T &= [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T \end{aligned}$$

Ces deux vecteurs définissent deux composantes répétitives stationnaires positives. En effet, à partir du marquage initial M_0 , qui est un marquage accessible, on peut franchir deux

séquences s_1 et s_2 qui sont deux invariants de transitions. Les équations de cycles de base sont obtenues comme suit :

$$\begin{cases} C(P_c, t_4) + C(P_c, t_5) + C(P_c, t_6) = 0 \\ C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) = 0 \end{cases}$$

Ensuite, l'ensemble des séquences de tir des transitions peut être déterminé comme convenu dans l'étape 2 de l'algorithme 4.1. Cependant, $T_{cycle} = \{t_1; t_2; t_3; t_4; t_5; t_6\}$, $card(T_{cycle}) = 6$. Le RdP est sauf, alors $|S| = \sum_{i=0}^6 C_6^i = C_6^0 + C_6^1 + C_6^2 + C_6^3 + C_6^4 + C_6^5 + C_6^6 = 64$. Ce qui correspond au tirage sans remise de : zéro transitions parmi six (qui se réfère au marquage initial M_0), une transition tirée à partir de T_{cycle} , deux transitions choisies parmi T_{cycle} , trois transitions de l'ensemble T_{cycle} , quatre transitions parmi six, cinq transitions tirées de T_{cycle} , et finalement le tirage de six transitions de T_{cycle} .

Comme illustrer à la figure 4.2, une fois les marquages interdits MI identifiés à partir de la $CGEM \Rightarrow MI = P_1P_3$, la méthode canonique est utilisé pour déterminer la séquence de franchissement interdite σ_{MI} .

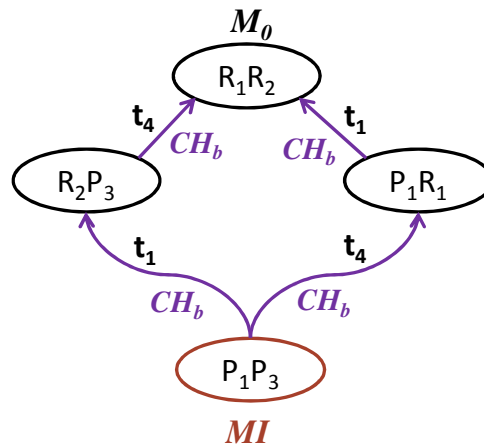


Figure 4.2- Méthode canonique

La séquence de franchissement de transitions interdite est $\sigma_{MI} = [1\ 0\ 0\ 1\ 0\ 0]^T$. Le filtrage de l'ensemble S des séquences obtenues constitue une étape majeure de notre méthode pour déterminer les séquences utilisables σ_U . Les informations détaillées des séquences σ_U et leurs marquages synthétisés sont répertoriés dans le tableau 4.1 suivant :

Tableau 4.1- Les séquences utilisables et leurs marquages

Séquences Utilisables	Séquence $\sigma_i = (t_1 t_2 t_3 t_4 t_5 t_6)$	Marquage synthétisé $(P_1 P_2 P_3 R_1 R_2 P_4 P_5 P_6)$
s_0	(0 0 0 0 0 0)	$M_0 = (0\ 0\ 0\ 1\ 1\ 0\ 0\ 0)$
s_1	(1 0 0 0 0 0)	$M_1 = (1\ 0\ 0\ 1\ 0\ 0\ 0\ 0)$
s_2	(0 0 0 1 0 0)	$M_2 = (0\ 0\ 1\ 0\ 1\ 0\ 0\ 0)$
s_3	(1 1 0 0 0 0)	$M_3 = (0\ 1\ 0\ 0\ 1\ 0\ 0\ 0)$
s_4	(0 0 0 1 1 0)	$M_4 = (0\ 0\ 0\ 1\ 0\ 1\ 0\ 0)$
$s_5 \in \sigma_{MI}$	(1 0 0 1 0 0)	$MI = (1\ 0\ 1\ 0\ 0\ 0\ 0\ 0)$
s_6	(0 0 0 2 1 0)	$M_6 = (0\ 0\ 0\ 1\ 0\ 1\ 0\ 0)$

s_7	$(2\ 1\ 0\ 0\ 0\ 0)$	$M_7 = (1\ 1\ 0\ 0\ 0\ 0\ 0\ 0)$
-------	----------------------	----------------------------------

Une fois les séquences utilisables et les équations de cycle déterminées, on applique la théorie des régions via son système linéaire composé de :

- Condition de séparation d'évènements : à partir de $\sigma_{MI} = [1\ 0\ 0\ 1\ 0\ 0]^T$, on peut écrire $M_0(P_c) + C(P_c, t_1) + C(P_c, t_4) < 0$.
- Les équations de cycle de base : $C(P_c, t_4) + C(P_c, t_5) + C(P_c, t_6) = 0$ et $C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) = 0$
- Les conditions d'atteignabilité : à partir de σ_U/σ_{MI} on a:

$$M_0(P_c) \geq 0$$

$$M_0(P_c) + C(P_c, t_1) \geq 0$$

$$M_0(P_c) + C(P_c, t_4) \geq 0$$

$$M_0(P_c) + C(P_c, t_1) + C(P_c, t_2) \geq 0$$

$$M_0(P_c) + C(P_c, t_4) + C(P_c, t_5) \geq 0$$

$$M_0(P_c) + 2C(P_c, t_4) + C(P_c, t_5) \geq 0$$

$$M_0(P_c) + 2C(P_c, t_1) + C(P_c, t_2) \geq 0$$

La résolution de ce système linéaire de la théorie des régions nous donne une seule place de contrôle P_c caractérisée par : $M_0(P_c) = 3$; $C(P_c, \cdot) = (-2, 1, 1, -2, 1, 1)$. Le modèle RdP contrôlé est donné dans la figure 4.3 suivante :

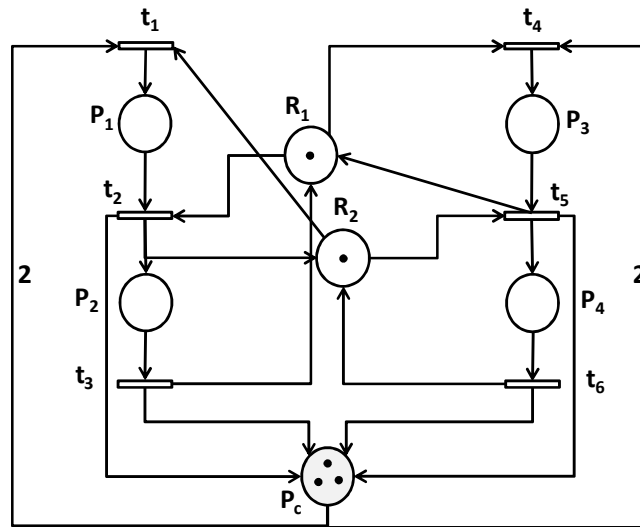


Figure 4.3- Modèle RdP contrôlé

4.2.3 Application sur un système de production flexible

Un système de production flexible contenant une station de contrôle qualité et son modèle RdP est présentés dans la figure 4.4. Cette station constitue la 4ème étape du processus de production de pièces en verre. Une fois que la pièce est gravée et assemblée avec des socles de différentes couleurs dans la place P_1 , elle sera testée et vérifiée dans la station de contrôle

de qualité au niveau de la place P_3 . Si les mesures de ses dimensions sont conformes à la norme, la pièce passe à la station d'emballage via la place P_5 , sinon elle sera rejetée dans le convoyeur de déchets modélisé par la place P_6 . La capacité de la station est modélisée par la place P_4 , soit une capacité d'une seule place. En outre, la place P_7 représente la priorité d'une pièce dans le convoyeur annexe (P_3) contre une pièce dans le convoyeur central (P_2).

Il faut noter que les transitions t_1 et t_5 sont incontrôlables, tandis que les transitions (t_2, t_3, t_4, t_6, t_7) sont contrôlables. Une description du rôle de chaque transition et de chaque place est détaillée dans le tableau 4.2 ci-après. La spécification de contrôle à satisfaire est exprimée sous forme de *CGEM*. En effet, cette contrainte consiste à maintenir le marquage de P_5 plus le double du marquage de P_6 moins de trois jetons.

La *CGEM* est : $M(P_5) + 2M(P_6) \leq 3$.

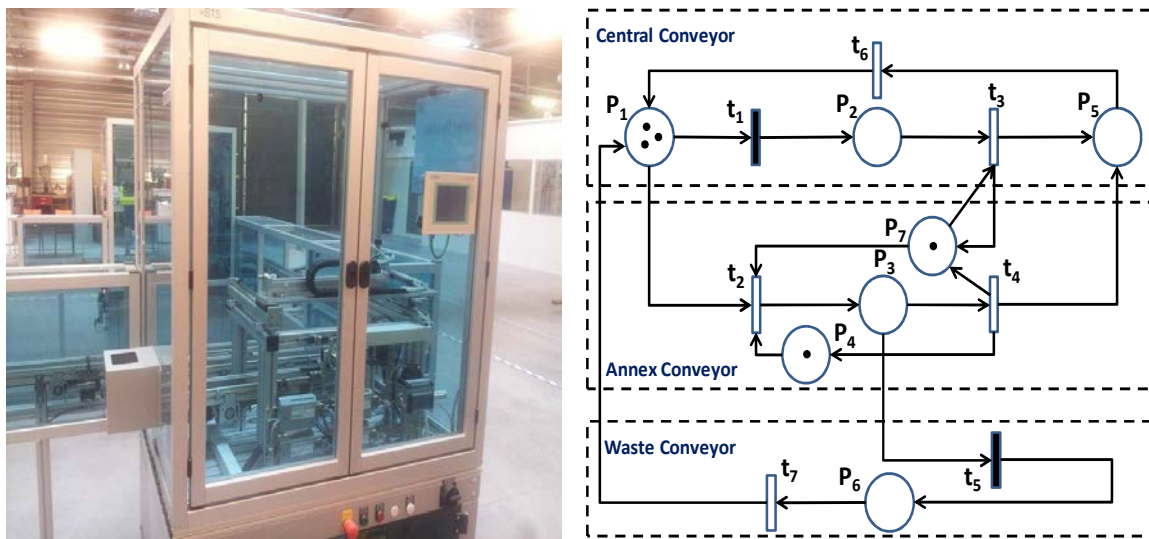


Figure 4.4- Station de contrôle qualité

Tableau 4.2- Description du modèle RdP

Place-Transition	Modélisation
P_1	Stock de production
P_2	Convoyeur central
P_3	Contrôle qualité
P_4	Capacité de contrôle
P_5	Produit contrôlé
P_6	Convoyeur de déchets
P_7	Priorité d'une pièce testée
$t_1; t_3$	Flux des palettes dans le convoyeur central
t_2	Entrée des palettes dans la station
t_4	Sortie des palettes
t_5	Produit défectueux
t_6	Passage à la station suivante
t_7	Produit recyclé

Conformément à la CGEM, les combinaisons admissibles des marquages sont listées dans le tableau 4.3 suivant :

Tableau 4.3- Combinaisons admissibles de marquages

$M(P_5)$	$M(P_6)$
0	1
1	0
1	1
2	0
3	0

Ainsi, la combinaison interdite de marquages est donnée dans le tableau 4.4 suivant :

Tableau 4.4- Combinaison interdite de marquages

$M(P_5)$	$M(P_6)$
2	1

1- Les équations de cycles de base : $C.X_t = [0 \ 0 \ \dots \ 0]^T$

$$C.X_t = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{1t} \\ X_{2t} \\ X_{3t} \\ X_{4t} \\ X_{5t} \\ X_{6t} \\ X_{7t} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{cases} -X_{1t} - X_{2t} + X_{6t} + X_{7t} = 0 \\ X_{1t} - X_{3t} = 0 \\ X_{2t} - X_{4t} - X_{5t} = 0 \\ -X_{2t} + X_{4t} = 0 \\ X_{3t} + X_{4t} - X_{6t} = 0 \\ X_{5t} - X_{7t} = 0 \\ -X_{2t} + X_{4t} = 0 \end{cases} \implies X_t^1 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}; X_t^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Les équations de cycles sont:

$$\begin{cases} C(P_c, t_1) + C(P_c, t_3) + C(P_c, t_6) = 0 \\ C(P_c, t_2) + C(P_c, t_4) + C(P_c, t_6) = 0 \end{cases} \implies \text{card}(T_{\text{cycle}}) = 5$$

Le réseau de Petri est k -borné, avec $k > 1$ alors:

$$|S| = \sum_{i=0}^{\text{card}(T_{\text{cycle}})} \Gamma_{\text{card}(T_{\text{cycle}})}^i = \sum_{i=0}^5 \Gamma_5^i = \Gamma_5^0 + \Gamma_5^1 + \Gamma_5^2 + \Gamma_5^3 + \Gamma_5^4 + \Gamma_5^5 = 252$$

S correspond au tirage avec remise de : zéro transitions (M_0), une transition de T_{cycle} , deux transitions quelconques de T_{cycle} , toutes les combinaisons de trois transitions choisies parmi T_{cycle} , quatre transitions retenues de l'ensemble T_{cycle} et enfin le tirage de toutes les combinaisons possibles de cinq transitions désignées parmi l'ensemble T_{cycle} .

Le marquage interdit est $MI = 2P_5P_3$. D'après nos travaux précédents [RAR15], on peut tester si la transition à interdire menant à MI est contrôlable ou pas. Si elle est incontrôlable, on effectue des chainages arrière CH_b jusqu'à ce que la transition interdite soit contrôlable (voir figure 4.5).

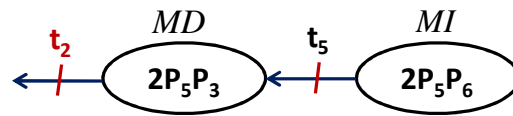


Figure 4.5- Transition incontrôlable

La transition interdite t_5 qui mène à $MI = 2P_5P_3$ est incontrôlable, on applique alors un chainage arrière pour déterminer l'état dangereux $MD = 2P_5P_3$. Ainsi, la transition t_2 devient interdite $\Rightarrow \sigma_{MI} = (2 \ 1 \ 2 \ 0 \ 0 \ 0 \ 0)$. Les séquences utilisables et inutilisables respectivement σ_U et σ_{UN} sont maintenant à déterminer. Nous listons dans le tableau 4.5 ci-dessous les séquences de transitions utilisables.

Tableau 4.5- Séquences utilisables

Séquences utilisables σ_U	Séquence $\sigma_i = (t_1 t_2 t_3 t_4 t_5 t_6 t_7)$	Marquage synthétisé $M_i = (P_1 P_2 P_3 P_4 P_5 P_6 P_7)$
s_0	(0 0 0 0 0 0 0)	$M_0 = (3 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1)$
s_1	(0 1 0 0 0 0 0)	$M_1 = (2 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0)$
s_2	(1 0 0 0 0 0 0)	$M_2 = (2 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1)$
s_3	(0 1 0 0 1 0 0)	$M_3 = (2 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0)$
s_4	(1 0 1 0 0 0 0)	$M_4 = (2 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1)$
s_5	(1 1 0 0 0 0 0)	$M_5 = (1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0)$
s_6	(2 0 0 0 0 0 0)	$M_6 = (1 \ 2 \ 0 \ 1 \ 0 \ 0 \ 1)$
s_7	(0 1 0 0 1 0 1)	$M_7 = (3 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$
s_8	(1 1 0 0 1 0 0)	$M_8 = (1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0)$
s_9	(1 1 1 0 0 0 0)	$M_9 = (1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0)$
s_{10}	(2 0 1 0 0 0 0)	$M_{10} = (1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1)$
s_{11}	(2 1 0 0 0 0 0)	$M_{11} = (0 \ 2 \ 1 \ 0 \ 0 \ 0 \ 0)$
s_{12}	(3 0 0 0 0 0 0)	$M_{12} = (0 \ 3 \ 0 \ 1 \ 0 \ 0 \ 1)$
s_{13}	(1 1 0 0 1 0 1)	$M_{13} = (2 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0)$
s_{14}	(2 1 0 0 1 0 0)	$M_{14} = (0 \ 2 \ 0 \ 0 \ 0 \ 1 \ 0)$
s_{15}	(1 1 1 0 1 0 0)	$M_{15} = (1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0)$
s_{16}	(2 0 2 0 0 0 0)	$M_{16} = (1 \ 0 \ 0 \ 1 \ 2 \ 1 \ 0)$
s_{17}	(2 1 1 0 0 0 0)	$M_{17} = (0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0)$

s_{18}	(3 0 1 0 0 0 0)	$M_{18} = (0 2 0 1 1 0 1)$
s_{19}	(2 1 0 0 1 0 1)	$M_{19} = (1 2 0 0 0 0 0)$
s_{20}	(1 1 1 0 1 0 1)	$M_{20} = (2 0 0 0 1 0 0)$
s_{21}	(2 1 1 0 1 0 0)	$M_{21} = (0 1 0 0 1 1 0)$
$s_{22} \in \sigma_{MI}$	(2 1 2 0 0 0 0)	$MD = (0 0 1 0 2 0 0)$
s_{23}	(2 1 1 1 0 0 0)	$M_{23} = (0 1 0 1 2 0 1)$
s_{24}	(3 1 0 0 1 0 1)	$M_{24} = (0 3 0 0 0 0 0)$
s_{25}	(2 1 1 0 1 0 1)	$M_{25} = (1 1 0 0 1 0 0)$
s_{26}	(2 1 2 1 0 0 0)	$M_{26} = (0 0 0 1 3 0 1)$

2- Condition de séparation d'évènements :

La séquence interdite est $\sigma_{MI} = (2 1 2 0 0 0 0)$, ainsi la condition de séparation d'évènement est traduite comme suit : $M_{22}(P_c) = M_0(P_c) + 2C(P_c, t_1) + 2C(P_c, t_3) + C(P_c, t_2) < 0$.

3- Conditions d'atteignabilité des états :

Les conditions d'atteignabilité peuvent être obtenues à partir de σ_U/σ_{MI} . Le nombre d'équations est de 26 équations déduites du tableau 4.5. Une fois le système linéaire de la théorie des régions obtenu à partir des conditions d'atteignabilité, des équations de cycles et de la condition de séparation d'évènement, on peut calculer le contrôleur RdP correspondant :

$$P_c: M_0(P_c) = 3; C(P_c, \cdot) = (0 - 2 - 1 1 0 1 0)$$

Le calcul de ce superviseur est effectué par l'intermédiaire de la bibliothèque *CPLEX* du programme C. Finalement, la place de contrôle est additionnée au modèle RdP initial (figure 4.4) pour avoir le modèle contrôlé montré dans la figure 4.6 suivante :

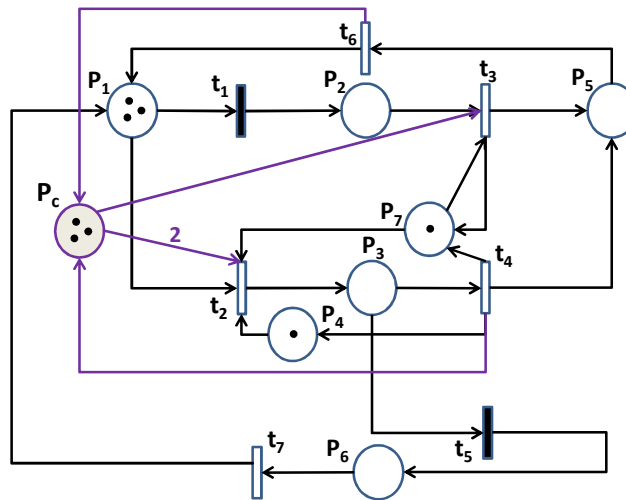


Figure 4.6- Modèle RdP contrôlé

4.3 Approche 2 : Synthèse de contrôle basée sur les propriétés des RdP

Dans cette partie, nous développons une deuxième approche pouvant faire face aux limites que présentent la théorie des régions, à savoir la complexité de calcul des contrôleurs RdP quand il existe une explosion combinatoire des états dans le graphe de marquage. L'objectif de ce chapitre vise à améliorer dans certaines situations la théorie des régions pour la synthèse de commande.

4.3.1 Détermination du système linéaire de la théorie des régions

On propose dans cette section une deuxième approche qui permet de déterminer les équations de cycles, les conditions d'atteignabilité et les conditions de séparation d'évènements sans utiliser le graphe de marquage $G(N, M_0)$.

Définition 4.4 : (Marquage primitif)

Un marquage primitif M_p est un vecteur présenté avec les coordonnées suivantes $M_p^t = (x_1 x_2 \dots x_n)^t$, n est le nombre de places qui existent dans le réseau de Petri.

Définition 4.5 : (Système linéaire des chemins SL_p)

Soit un RdP donné avec $M_0^t = (y_1 y_2 \dots y_n)^t$ et C sa matrice d'incidence. Un système linéaire des chemins SL_p est présenté comme suit : $C \cdot \sigma = M_p - M_0$.

$$C \cdot \sigma = \begin{pmatrix} x_1 - y_1 \\ x_2 - y_2 \\ \cdot \\ \cdot \\ x_n - y_n \end{pmatrix} \geq \begin{pmatrix} -y_1 \\ -y_2 \\ \cdot \\ \cdot \\ -y_n \end{pmatrix}$$

Cette deuxième approche se base aussi sur une méthode analytique. En effet, les invariants de transitions et de places sont introduits dans notre politique de contrôle. Dans ce qui suit, on présente l'algorithme 4.2 qui traduit notre nouvelle méthodologie permettant de synthétiser des contrôleurs RdP.

Algorithme 4.2 : Synthèse des contrôleurs RdP en se basant sur la théorie des régions sans utiliser le graphe de marquage.

Soient un RdP donné avec son marquage initial M_0 , C sa matrice d'incidence et S l'ensemble des chemins reliant M_0 à l'ensemble des états M .

- 1 Identifier les équations de cycles en déterminant les invariants de transitions
- 2 Fixer les marquages primitifs selon la nature du RdP
- 3 Déterminer le système linéaire des chemins SL_p
- 4 Résoudre SL_p et soit $S = \sum_i \sigma_i$ l'ensemble de tous les chemins possibles qui existent entre M_0 et tout marquage $M \in M$.
- 5 Eliminer les séquences redondantes
- 6 Déterminer les invariants de places s'ils existent
- 7 Générer les marquages finis M_x en utilisant l'équation fondamentale des RdP ou les invariants de places.

- 8 Assigner à chaque marquage fini M_{xi} son type.
- 9 Corriger le type de chaque marquage M_{xi} s'il existe des transitions incontrôlables dans le réseau de Petri.
- 10 Pour chaque instance de séparation d'évènements, dresser le système linéaire de la théorie des régions (SL).
- 11 Résoudre SL et soit $(M_0(P_{ci}); C(P_{ci}, \cdot))$ la solution si elle existe.
- 12 Eliminer les places de contrôle redondantes afin d'obtenir le modèle RdP contrôlé.

Etape 1 : Calcul des équations de cycles de base :

En calculant les t-invariants, on peut déterminer les équations de cycles qui sont utiles pour former le système linéaire de la théorie des régions nous permettant de calculer les places de contrôles.

Etape 2 : Calcul des marquages primitifs :

Selon la nature du réseau de Petri, on peut déterminer le nombre (n) de places qui existent. Cependant le marquage primitif sera un vecteur à n inconnus.

Exemple : $M_{pi}^t = (x_1 x_2 \dots x_n)^t$.

Etape 3-4 : Déterminer le système linéaire des chemins SL_p :

En résolvant le système linéaire des chemins, on peut déterminer l'ensemble des séquences $S = \sum_i \sigma_i$ qui nous donne tous les chemins possibles reliant l'état initial M_0 et tous les autres états $M \in M$. Le SL_p est défini comme suit : $C \cdot \sigma = M_p - M_0$

Etape 5 : Elimination des séquences redondantes:

Après avoir déterminé l'ensemble des séquences S en résolvant le SL_p , des solutions existantes peuvent contenir d'éventuels cycles. Il faut donc éliminer cette redondance.

Etape 6 : Calcul des invariants des places :

En calculant les p-invariants, on peut déterminer directement les marquages finis M_x . En effet, si toutes les places existent dans les p-invariants, tous les états peuvent être identifiés.

Etape 7 : Détermination des marquages finis M_x :

Les marquages finis peuvent être déterminés en utilisant l'équation fondamentale des RdP : $M_x = M_0 + C \cdot \sigma$, ou en déterminant les p-invariants.

Etape 8 : Détermination du type des marquages finis M_x :

Selon la contrainte ($CGEM$), on assigne à chaque marquage fini son type, à savoir : marquage interdit, marquage admissible ou marquage dangereux.

Etape 9 : Correction du type des marquages finis M_x :

Si la transition interdite est incontrôlable, alors il faut corriger le type du marquage admissible en un marquage dangereux, qui doit être à son tour un état interdit. En effet, en effectuant la méthode du chaînage arrière [RAR15], on peut identifier les marquages dangereux.

Etape 10-11 : Détermination du système linéaire de la théorie des régions :

Une fois que les invariants de transitions sont calculés, on peut déterminer les équations de cycles de base. De plus, les marquages finis et les séquences S nous donnent les conditions d'atteignabilité. Par conséquent, pour chaque instance de séparation d'évènements, on forme le système linéaire de la théorie des régions SL_i . Sa résolution nous génère une place de contrôle P_{ci} .

Etape 12 : Elimination de la redondance des contrôleurs RdP :

En résolvant tous les systèmes linéaires de la théorie des régions, il est fort possible d'avoir des places de contrôle redondantes. Cependant, leur élimination nous permet de filtrer l'ensemble des solutions. L'ajout de ces contrôleurs RdP nous permet d'obtenir enfin le modèle RdP contrôlé.

Remarque 4.2 :

La détermination des invariants de places présente un facteur important dans notre méthode. En effet, si toutes les places du RdP existent dans les p-invariants déterminés, alors les marquages finis M_x peuvent être directement déterminés à partir des invariants de places calculés.

4.3.2 Exemples

Dans cette section, on présente quelques exemples illustratifs qui permettent de mieux comprendre la méthode de contrôle proposée.

4.3.2.1 Exemple 1

Pour cet exemple de réseau de Petri, toutes les transitions sont contrôlables, sauf la transition t_3 qui est incontrôlable.

On considère la CGEM suivante pour ce RdP : $M(P_4) + M(P_6) \leq 3$.

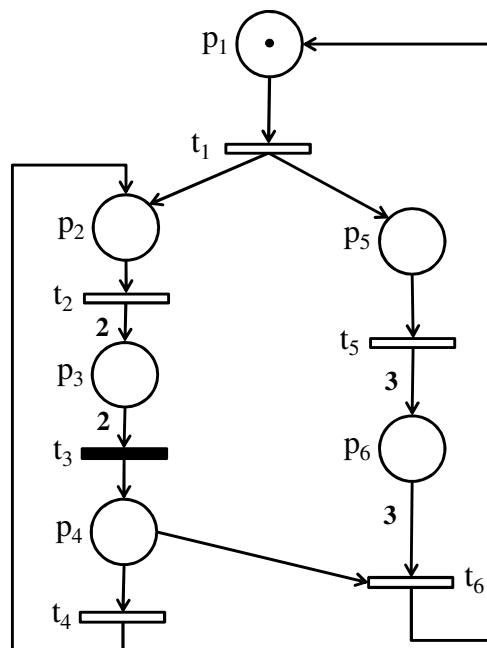


Figure 4.7- Exemple RdP 1

Dans un premier temps, on applique la méthode des invariants de transitions pour déterminer les équations de cycles.

Le système $C.X_t = 0$ avec $X_t^T = [t'_1 t'_2 t'_3 t'_4 t'_5 t'_6]^T$ est :

$$\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} t'_1 \\ t'_2 \\ t'_3 \\ t'_4 \\ t'_5 \\ t'_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{cases} -t'_1 + t'_6 = 0 \\ t'_1 - t'_2 + t'_4 = 0 \\ 2t'_2 - 2t'_3 = 0 \\ t'_3 - t'_4 - t'_6 = 0 \\ t'_1 - t'_5 = 0 \\ 3t'_5 - 3t'_6 = 0 \end{cases} \implies \begin{cases} t_1 + t_2 + t_3 + t_5 + t_6 = 0 \\ t_2 + t_3 + t_4 = 0 \end{cases}$$

Le réseau de Petri contient 6 places, donc $n = 6$. Le marquage primitif à déterminer est le suivant : $M_{pi}^t = (x_1 x_2 x_3 x_4 x_5 x_6)^t$. Le système linéaire des chemins SL_p est le suivant : $C.\sigma = M_p - M_0$:

$$M_p - M_0 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \geq -1 \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ \geq 0 \end{pmatrix}$$

$$SL_p: \begin{cases} -t'_1 + t'_6 \geq -1 \\ t'_1 - t'_2 + t'_4 \geq 0 \\ 2t'_2 - 2t'_3 \geq 0 \\ t'_3 - t'_4 - t'_6 \geq 0 \\ t'_1 - t'_5 \geq 0 \\ 3t'_5 - 3t'_6 \geq 0 \end{cases}$$

$$\Rightarrow \sigma_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}; \sigma_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}; \sigma_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}; \sigma_4 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \sigma_5 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \sigma_6 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

Ces séquences sont déterminées après un filtre qui s'est fait sur les solutions du système linéaire des chemins SL_p . En effet, par la soustraction de chaque solution redondante des vecteurs de cycles, on trouve une des six solutions déterminées σ_i . Une fois qu'on s'aperçoit que les solutions générées sont déjà trouvées, on arrête la recherche de séquences. On peut conclure qu'à ce stade, toutes les solutions ont été trouvées.

Une fois que les séquences sont calculées, il ne nous reste qu'à déterminer les marquages finis M_x en utilisant l'équation fondamentale des RdP : $M_{xi} = M_0 + C \cdot \sigma_i$.

$$M_{x1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$M_{x2} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$M_{x3} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$M_{x4} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 3 \end{pmatrix}$$

$$M_{x5} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 3 \end{pmatrix}$$

$$M_{x6} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 3 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 3 \end{pmatrix}$$

La CGEM à satisfaire est la suivante : $M(P_4) + M(P_6) \leq 3$. La transition t_3 est incontrôlable. D'après la CGEM, le type du marquage fini M_{x6} est : *Interdit*, tandis que tous les autres états sont : *Atteignables*. La séquence qui mène au marquage fini M_{x6} est : $t_1 + t_2 + t_3 + t_5$. Ce chemin contient une transition incontrôlable (t_3). $\Rightarrow \sigma_6 - t_3 = t_1 + t_2 + t_5 = \sigma_5$. Donc le type du marquage M_{x5} est : *dangereux*. Ainsi, le système linéaire de la théorie des régions est formé comme suit :

- Equations de cycles :

$$C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) + C(P_c, t_5) + C(P_c, t_6) = 0$$

$$C(P_c, t_2) + C(P_c, t_3) + C(P_c, t_4) = 0$$

- Conditions d'atteignabilité :

$$M_0(P_c) \geq 0$$

$$M_1(P_c) = M_0(P_c) + C(P_c, t_1) \geq 0$$

$$M_2(P_c) = M_0(P_c) + C(P_c, t_1) + C(P_c, t_2) \geq 0$$

$$M_3(P_c) = M_0(P_c) + C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) \geq 0$$

$$M_4(P_c) = M_0(P_c) + C(P_c, t_1) + C(P_c, t_5) \geq 0$$

- Conditions de séparation d'évènements :

$$M_5(P_c) = M_0(P_c) + C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_5) < 0$$

$$M_6(P_c) = M_0(P_c) + C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) + C(P_c, t_5) < 0$$

Pour chaque condition de séparation, on résout le système linéaire de la théorie des régions correspondant en utilisant la bibliothèque CPLEX du programme C.

Les contrôleurs RdP obtenus sont :

$$P_{c1}: M_0(P_{c1}) = 1; C(P_{c1}, \cdot) = (1, -1, 1, 0, -2, 1)$$

$$P_{c2}: M_0(P_{c2}) = 1; C(P_{c2}, \cdot) = (1, 1, -2, 1, -2, 0)$$

4.3.2.2 Exemple 2

Dans cet exemple, la transition t_3 est aussi incontrôlable. La contrainte à satisfaire est la suivante : $M(P_4) + M(P_6) \leq 1$.

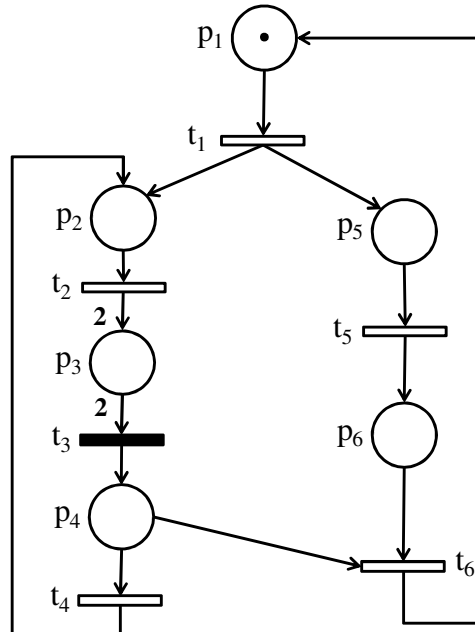


Figure 4.8- Exemple RdP 2

L'application de la méthode des t-invariants nous donne les mêmes équations de cycles de l'exemple 1 qui sont :

$$\begin{cases} t_1 + t_2 + t_3 + t_5 + t_6 = 0 \\ t_2 + t_3 + t_4 = 0 \end{cases}$$

On calcule dans cet exemple les invariants des places :

$$\begin{pmatrix} P_1' & P_2' & P_3' & P_4' & P_5' & P_6' \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{cases} -P_1' + P_2' + P_5' = 0 \\ -P_2' + 2P_3' = 0 \\ -2P_3' + P_4' = 0 \\ P_2' - P_4' = 0 \\ -P_5' + P_6' = 0 \\ P_1' - P_4' - P_6' = 0 \end{cases} \Rightarrow P_1' + P_5' + P_6' = 1$$

Les places P_1 , P_5 et P_6 sont comprises dans les invariants de places, tandis que les trois autres places ne le sont pas. Par conséquent, les marquages primitifs sont :

$$M_{p1}^t = (1 \ x_2 \ x_3 \ x_4 \ 0 \ 0)^t ; M_{p2}^t = (0 \ x_2 \ x_3 \ x_4 \ 1 \ 0)^t \text{ et } M_{p3}^t = (0 \ x_2 \ x_3 \ x_4 \ 0 \ 1)^t.$$

Systemes linéaires des chemins :

Pour chaque marquage primitif, on détermine le système linéaire des chemins correspondant :

$$(1)- C.\sigma_i = M_{p_i} - M_0 : \begin{pmatrix} 1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ 0 \\ 0 \end{pmatrix}$$

$$C.\sigma_1 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{pmatrix} = \begin{pmatrix} 0 \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} -t_1 + t_6 = 0 \\ t_1 - t_2 + t_4 \geq 0 \\ 2t_2 - 2t_3 \geq 0 \\ t_3 - t_4 - t_6 \geq 0 \\ t_1 - t_5 = 0 \\ t_5 - t_6 = 0 \end{cases} \Rightarrow \sigma_1 = (000000)^t$$

$$(2)- C.\sigma_2 = M_{p_2} - M_0 : \begin{pmatrix} 0 \\ x_2 \\ x_3 \\ x_4 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} -t_1 + t_6 = -1 \\ t_1 - t_2 + t_4 \geq 0 \\ 2t_2 - 2t_3 \geq 0 \\ t_3 - t_4 - t_6 \geq 0 \\ t_1 - t_5 = 1 \\ t_5 - t_6 = 0 \end{cases} \Rightarrow \sigma_2 = (100000)^t; \sigma_3 = (110000)^t; \sigma_4 = (111000)^t$$

$$(3)- C.\sigma_2 = M_{p_3} - M_0 : \begin{pmatrix} 0 \\ x_2 \\ x_3 \\ x_4 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ \geq 0 \\ \geq 0 \\ \geq 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\Leftrightarrow \begin{cases} -t_1 + t_6 = -1 \\ t_1 - t_2 + t_4 \geq 0 \\ 2t_2 - 2t_3 \geq 0 \\ t_3 - t_4 - t_6 \geq 0 \\ t_1 - t_5 = 0 \\ t_5 - t_6 = 1 \end{cases} \Rightarrow \sigma_5 = (1\ 0\ 0\ 0\ 1\ 0)^t; \sigma_6 = (1\ 1\ 0\ 0\ 1\ 0)^t; \sigma_7 = (1\ 1\ 1\ 0\ 1\ 0)^t$$

Une fois que les séquences sont déterminées, on peut calculer les marquages finis $M_{p_{xi}}$ en utilisant l'équation fondamentale des RdP : $M_{p_{xi}} = M_0 + C.\sigma_i$.

$$M_{p_{x1}} = M_0 + C.\sigma_1 = (1\ 0\ 0\ 0\ 0\ 0)^t \Rightarrow \text{Marquage initial}$$

$$M_{p_{x2}} = M_0 + C.\sigma_2 = (0\ 1\ 0\ 0\ 1\ 0)^t$$

$$M_{p_{x3}} = M_0 + C.\sigma_3 = (0\ 0\ 2\ 0\ 1\ 0)^t$$

$$M_{p_{x4}} = M_0 + C.\sigma_4 = (0\ 0\ 0\ 1\ 1\ 0)^t$$

$$M_{p_{x5}} = M_0 + C.\sigma_5 = (0\ 1\ 0\ 0\ 0\ 1)^t$$

$$M_{p_{x6}} = M_0 + C.\sigma_6 = (0\ 0\ 2\ 0\ 0\ 1)^t$$

$$M_{p_{x7}} = M_0 + C.\sigma_7 = (0\ 0\ 0\ 1\ 0\ 1)^t$$

Correction des types :

$$CGEM : M(P_4) + M(P_6) \leq 1.$$

D'après la CGEM, le marquage $M_{p_{x7}}$ est de type : *Interdit*. Les restes des états sont des marquages *atteignables*.

Le chemin menant à $M_{p_{x7}}$ est : $t_1 + t_2 + t_3 + t_5$. Ce chemin contient une transition incontrôlable (t_3).

$(t_1 + t_2 + t_3 + t_5) - (t_3) = (t_1 + t_2 + t_5) = \sigma_6$. Ce qui nous amène à $M_{p_{x6}}$. Cet état un état *dangereux*.

Ainsi, le système linéaire de la théorie des régions est formé comme suit :

- Equations des cycles :

$$\begin{aligned} C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) + C(P_c, t_5) + C(P_c, t_6) &= 0 \\ C(P_c, t_2) + C(P_c, t_3) + C(P_c, t_4) &= 0 \end{aligned}$$

- Conditions d'atteignabilité :

$$M_1(P_c) \geq 0$$

$$M_2(P_c) = M_0(P_c) + C(P_c, t_1) \geq 0$$

$$M_3(P_c) = M_1(P_c) + C(P_c, t_1) + C(P_c, t_2) \geq 0$$

$$M_4(P_c) = M_1(P_c) + C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) \geq 0$$

$$M_5(P_c) = M_1(P_c) + C(P_c, t_1) + C(P_c, t_5) \geq 0$$

- Conditions de séparation d'évènements :

$$M_6(P_c) = M_1(P_c) + C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_5) < 0$$

$$M_7(P_c) = M_1(P_c) + C(P_c, t_1) + C(P_c, t_2) + C(P_c, t_3) + C(P_c, t_5) < 0$$

Pour chaque condition de séparation, on résout le système linéaire de la théorie des régions correspondant en utilisant la bibliothèque *CPLEX* du programme *C*.

4.4 Implémentation des contrôleurs RdP sur un automate programmable (PLC)

Dans cette dernière section, on présente une partie expérimentale dans laquelle on implémente des contrôleurs RdP sur un automate programmable (PLC : Programmable Logic Controller) d'un système de production flexible. Ce système est un atelier flexible présent sur la plateforme du laboratoire LGIPM. Il est constitué de huit stations placées autour d'un convoyeur central. Le terme flexible désigne le fait que cette chaîne de production peut être rapidement reconfigurable en fonction des contraintes de production.

Cet atelier produit des pièces en verre gravées, assemblées et emballées. Les stations 3 et 4 ont la même fonction et représentent toutes les deux la troisième étape dans le processus de production, à savoir l'assemblage des pièces gravées avec des socles de différentes couleurs (voir figure 4.9). Cette partie a été développée dans le cadre du projet de fin d'études d'un élève ingénieur.

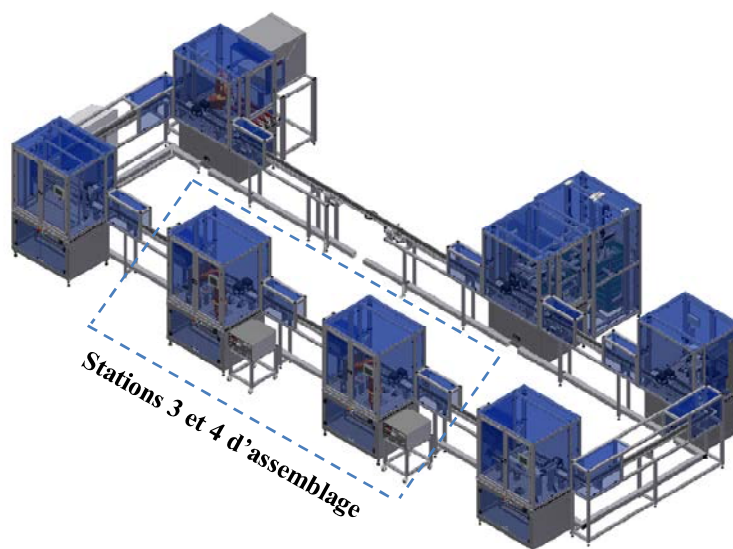


Figure 4.9- Stations d'assemblage de l'atelier flexible

4.4.1 Modélisation des stations d'assemblage par RdP

Ces deux stations ont été modélisées par réseau de Petri [RRKA14] et simulées par le logiciel CPN Tools. Ce dernier permet de simuler, d'éditer et d'analyser les RdP. Il dispose d'un ensemble d'outils aidant à la construction de l'espace d'état du modèle RdP, (voir figure 4.10).

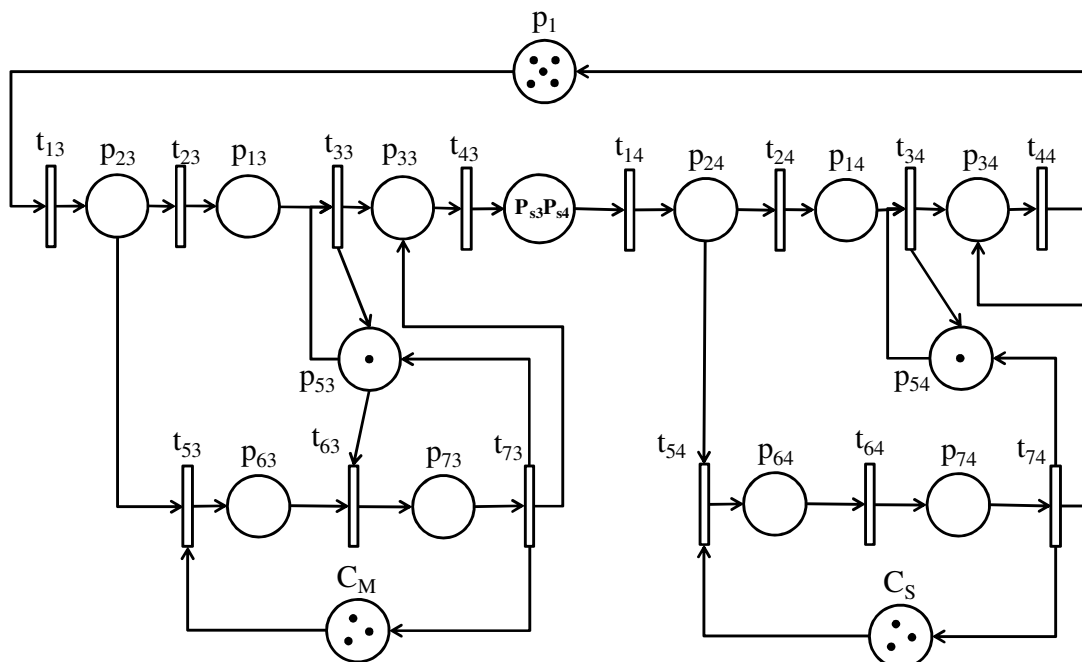


Figure 4.10- Modèle RdP des stations 3 et 4

Les capteurs, les effecteurs et les actionneurs des deux stations sont décrits dans les tableaux 4.6 et 4.7 suivants:

Tableau 4.6- Identification des capteurs

Désignation	Rôle
11B ₁	Détecte la présence d'une palette au niveau du bloqueur à l'entrée de la station d'assemblage
11B ₂	Indique la position haute du bloqueur B ₁
11B ₃	Détecte la présence d'une palette à l'entrée de la station
12B ₁	Indique la position initiale de l'aiguilleur de déviation
12B ₂	Indique la position tournée de l'aiguilleur de déviation
12B ₃	Marque la présence de trois palettes sur le convoyeur principal
12B ₄	Marque la présence de trois palettes sur le convoyeur annexe
13B ₁	Détecte la présence d'une palette au niveau du bloqueur présent sur le convoyeur annexe
13B ₂	Indique la position haute du bloqueur B ₂
13B ₃	Détecte la présence d'une palette sur le convoyeur annexe
14B ₁	Détecte la présence d'une palette au niveau du bloqueur à la sortie de la station d'assemblage
14B ₂	Indique la position haute du bloqueur B ₃
14B ₃	Détecte la sortie d'une palette

Tableau 4.7- Identification des effecteurs

Désignation	Rôle
Bloqueur B_1	Bloquer le passage des palettes en cas d'encombrement dans les deux convoyeurs
Bloqueur B_2	Bloquer le passage de la palette lors de l'opération d'assemblage
Bloqueur B_3	Bloquer le passage des palettes dans le cas où il y a une palette qui sort du convoyeur annexe. De cette façon, les collisions sont évitées.
Aiguilleur de déviation	Permet de dévier la palette vers le convoyeur annexe afin que la pièce en verre subisse l'opération d'assemblage. Il est actionné pneumatiquement.

Les références de chaque transition et de chaque place du modèle RdP sont regroupées dans le tableau 4.8 suivant :

Tableau 4.8- Description des transitions/places

Evènement/Place	Description
t_{1i}	Entrée d'une palette à la station i
t_{2i}	Poursuite du parcours de la palette sur le convoyeur principal
t_{3i}	Blocage de la palette jusqu'à la sortie de la palette provenant du convoyeur annexe
t_{4i}	Sortie de la palette de la station i
t_{5i}	Entrée de la palette au convoyeur annexe
t_{6i}	Début de l'opération d'assemblage
t_{7i}	Fin d'assemblage et sortie de la palette du convoyeur annexe
p_{1i}	Palette au niveau du bloqueur B_3
p_{2i}	Palette au niveau du bloqueur B_1
p_{3i}	Palette sur le convoyeur principal à la sortie de la station
p_{5i}	Priorité de la palette sortante du convoyeur annexe sur la palette venant dans le convoyeur principal
p_{6i}	Palette au niveau du bloqueur B_2
p_{7i}	Opération d'assemblage
C_M	Capacité du convoyeur annexe de la station 3
C_S	Capacité du convoyeur annexe de la station 4
$P_{S_3S_4}$	Palette sur le convoyeur reliant les deux stations d'assemblage

4.4.2 Programmation de l'automate programmable

La liaison entre le PC et l'automate programmable est réalisée à travers un câble *Ethernet* branché au niveau du commutateur industriel SCALANCE XB008, comme le montre la figure 4.11 suivante.



Figure 4.11- Branchement entre le PC et l'automate

Une fois que la communication entre le PC et l'API est établie et l'identification des entrées/sorties faite, on passe à la programmation sur STEP 7. Ce progiciel d'ingénierie complet permet d'exécuter toutes les opérations d'ingénierie d'un projet.

La CPU dont on dispose est la CPU 315F-2PN/DP qu'il faut configurer. Les entrées/sorties de la station 3 sont regroupées dans la table des mnémoniques (sur STEP7). Une adresse physique et un nom symbolique leurs ont été attribués.

4.4.3 Programmation des stations d'assemblage 3 et 4

La spécification de contrôle à satisfaire est exprimée sous forme de CCA (Contraintes de Chemins Admissibles) [RRKA14] : $2\sigma[t_{54}] - \sigma[t_{53}] \leq 0$. En effet, le scénario suivant est adopté : Afin d'absorber les pannes de la station 4, la première station (station 3) doit opérer deux fois plus que la deuxième (station 4) dans sa tâche d'assemblage des pièces gravées.

L'application de la théorie des régions génère le contrôleur RdP suivant [RRKA14] : $P_{c1} \Rightarrow M_0(P_{c1}) = 0; C(P_{c1}, \cdot) = (0,0,0,0, -1,0,0,0,0,0,2,0,0)$.

Il existe une liaison directe entre les deux stations d'assemblage ; de ce fait, la configuration des deux réseaux Ethernet se fait séparément. On attribue donc pour chaque station, une adresse IP différente. L'organigramme de programmation est donné dans la figure 4.12 suivante :

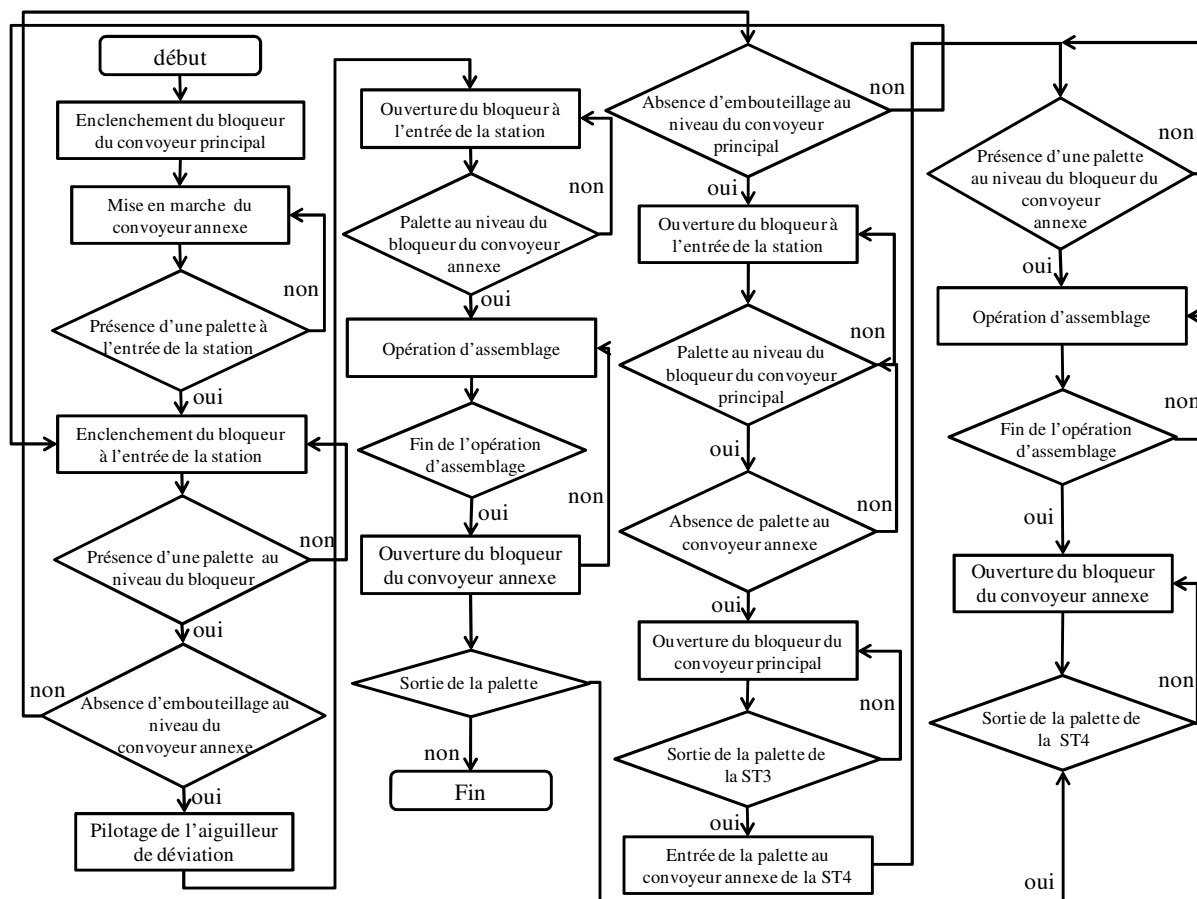


Figure 4.12- Organigramme du programme

La mise en marche des stations s'enclenche par une détection des palettes (produits) à l'entrée de la station 3 au niveau du capteur **11B3** ; par conséquent, une série d'actions se déroule :

- Le bloqueur de déviation se soulève empêchant ainsi le passage du chariot
- Le convoyeur annexe se met en marche
- Le bloqueur du convoyeur principal se met à l'état haut bloquant systématiquement le passage de chaque palette.

La programmation se fait par partie : la première partie consiste dans le pilotage de l'aiguilleur de déviation. Ce dernier se tourne vers la station de travail lors de l'activation du capteur **11B1**. L'aiguilleur maintient la position tournée tant qu'il n'y a pas d'embouteillage au niveau du convoyeur annexe et tant que des palettes sont présentées à l'entrée de la station. L'aiguilleur retourne à sa position initiale au bout de quelques secondes après que les palettes soient entrées.

La deuxième partie se réfère au mouvement du bloqueur de déviation ; il se bloque systématiquement dès la détection de la palette et se débloque au bout de quelques secondes permettant ainsi le passage de la palette vers la station de travail. Le bloqueur se maintient fermé jusqu'à ce que la palette sorte de la station. De cette façon, le chariot à l'entrée sera traité par la station 3.

La troisième partie traite le fonctionnement du bloqueur de convoyeur. Ce dernier cède le passage à la palette se trouvant au niveau du capteur **14B1** quand il n'y a aucune palette sortant de la station de travail. Dans le cas contraire, il se bloque jusqu'à ce que la palette soit sortie de la station 3.

La quatrième partie fait référence à l'opération d'assemblage. La palette se trouvant alors au niveau du capteur **13B1** va attendre jusqu'à ce que l'assemblage de la pièce en verre avec le socle soit terminé. La durée de traitement est de 15 secondes ; par la suite, le bloqueur de la position de travail s'abaisse pour que la palette sorte et que celle qui la suit se bloque.

Le principe de fonctionnement de la quatrième station (station 4) s'inverse. Alors que pour la station 3, la détection des palettes au niveau du bloqueur de déviation (activation du capteur **11B1**) enclenche le mouvement de l'aiguilleur, dans la station 4, ce n'est pas le cas. En effet, les trois premières palettes poursuivent leur parcours jusqu'au bloqueur du convoyeur principal situé à la sortie de la station. C'est alors, en marquant la présence d'embouteillage au niveau du bloqueur du convoyeur principal (détekté par le capteur **12B3**), que l'aiguilleur intervient pour permettre aux palettes suivantes d'entrer poursuivre l'opération d'assemblage. Lorsque la première palette entrant dans le convoyeur principal est acheminée au niveau du capteur **13B1**, les palettes se trouvant au niveau du bloqueur du convoyeur principal sont libérées.

4.5 Conclusion

Dans ce chapitre, nous avons présenté deux nouvelles approches de synthèse de commande utilisant la théorie des régions pour synthétiser des contrôleurs RdP qui ont été implémentés sur les automates programmables d'un système de production flexible lors de la dernière partie de ce chapitre.

En effet, l'objectif principal de ce travail est de réduire au maximum la complexité de calcul du superviseur RdP. Grâce à nos deux approches, nous avons pu éviter la génération du graphe d'états. Les résultats de ce chapitre démontrent l'enjeu important de nos méthodes qui révèlent un grand avantage dans le domaine de synthèse de contrôle.

En effet, dans la première approche, nous nous sommes basés sur les propriétés des réseaux de Petri et la théorie des régions sans tenir compte du graphe de marquage. L'exploration du graphe et l'explosion combinatoire des états alourdissent nettement la tâche de synthèse du contrôle. Cette approche ne tient pas compte du graphe et respecte toutes les propriétés des réseaux de Petri. Notre politique de contrôle a été appliquée sur un système de production flexible pour prouver l'efficacité des résultats de notre approche.

De plus, une deuxième approche a été développée dans ce chapitre. Elle démontre d'une manière simple comment utiliser des propriétés mathématiques et algébriques pour trouver un moyen efficace nous permettant de ne pas générer le graphe de marquage $G(N, M_0)$ et synthétiser au final les places de contrôle. Cette méthode se base sur les fondements de la théorie des régions. Ces résultats ont été appliqués sur des exemples, qui illustrent la robustesse de notre deuxième approche.

Dans la dernière section de ce chapitre, nous avons présenté une partie expérimentale dans laquelle des contrôleurs RdP ont été implémentés sur des automates programmables présents sur la plateforme de notre laboratoire. Dans cette partie, nous avons pu commander des stations d'assemblage et modifier leurs cadences de travail. Grâce à l'identification des entrées/sorties du système, nous avons modélisé les stations par RdP en respectant une spécification de contrôle ; les contrôleurs synthétisés par l'application de la théorie des régions ont été implémentés sur les machines en utilisant le progiciel STEP7.

Conclusion

L'objectif principal du travail de cette thèse est d'étudier différentes méthodes de synthèse de contrôle pour les SED modélisés avec les RdP. Les méthodes développées visent à réduire la complexité de calcul des contrôleurs RdP en utilisant la théorie des régions. Notre étude a abouti à la conception de plusieurs approches distinctes.

La première méthode permet de construire un contrôleur RdP sous forme de places de contrôle en utilisant les coupes minimales et la théorie des régions. La théorie des régions a été conçue pour synthétiser des réseaux de Petri à partir d'automates, mais elle a été réinterprétée dans la synthèse de places de contrôle. Nos deux premières approches s'appuient sur un nouveau concept de coupes minimales dans un graphe de marquage et diffèrent par la façon de choisir une coupe. En effet, l'application de la théorie des régions a été restreinte à des zones du graphe avec un nombre minimal de nœuds, appelées coupes minimales et non pas sur la totalité du graphe de marquage. Chaque contrôleur synthétisé aura la tâche de contrôler sa zone. Ainsi, les places de contrôle conçues sont pures et le contrôle déterminé est maximum permissif, c'est-à-dire que le système contrôlé a exactement le comportement souhaité. Grâce à cette première méthode, nous avons pu réduire, d'une part, le système linéaire de la théorie des régions et, d'autre part, le temps de calcul des places de contrôle. Une fonction Booléenne de contrôle a été définie afin de synchroniser et de switcher entre les contrôleurs RdP, du fait que chaque place contrôle sa coupe minimale correspondante.

La méthode de synthèse de contrôleurs RdP utilisant les coupes minimales nous a inspiré pour développer une nouvelle approche basée sur des marquages dits canoniques. Les marquages canoniques sont des pseudo-états qui forment une coupe minimale canonique CMC_i sur laquelle on applique la théorie des régions. L'intérêt de cette approche est la conception d'une CMC_i et donc d'une place de contrôle avant même de construire le graphe de marquage. Un avantage très important en découle. En effet, nous avons pu dans un premier temps synthétiser un contrôleur P_{ci} sans utiliser le graphe de marquage. Néanmoins, l'identification de la CMC_i et des marquages canoniques dans le graphe reste jusqu'à présent nécessaire pour valider l'efficacité de notre méthode.

L'inconvénient majeur de la synthèse de contrôle en utilisant la théorie des régions est qu'elle nécessite l'énumération de tous les marquages du comportement admissible. Nous avons continué à réfléchir sur une manière de soutenir l'objectif principal de ce travail, à savoir la réduction de la complexité de calcul de la théorie des régions pour la synthèse de contrôle. Nous en proposons deux approches qui utilisent la théorie des régions en se dépossédant du graphe de marquage pour la synthèse de contrôleurs RdP.

Une première idée consiste à transformer la problématique de contrôle en un exercice mathématique qui se résout en se basant sur les propriétés des réseaux de Petri. Le système linéaire de la théorie des régions est déterminé sans générer le graphe de marquage. En effet, à partir du modèle RdP, on arrive à identifier les équations de cycles en utilisant la méthode

d'invariants de transitions. Ensuite, les conditions d'atteignabilité et de séparation d'évènements seront déterminées en utilisant des combinaisons avec ou sans remise selon la nature du RdP. Finalement, la résolution du système linéaire obtenu conduit à l'obtention des contrôleurs RdP.

Nous proposons également une autre méthode de synthèse de contrôleurs RdP sans utiliser le graphe de marquage. En effet, cette approche se base sur une méthode analytique. Les invariants de places et de transitions sont introduits dans notre politique de contrôle. Ainsi, grâce aux propriétés des réseaux de Petri, nous avons pu déterminer le système linéaire de la théorie des régions et synthétiser des places de contrôle sans utiliser le graphe de marquage.

En outre, une nouvelle contribution a été ajoutée à ce travail, qui se situe au niveau de l'extension des types de spécification de contrôle. En effet, les Contraintes Généralisées d'Exclusion Mutuelles (CGME) n'arrivent à modéliser toutes les spécifications de contrôle. Ainsi, nous avons pu introduire un nouveau type de modélisation basé sur des combinaisons de séquences de transitions et non pas sur des combinaisons de marquages des places.

En résumé, nous avons développé plusieurs approches de synthèse de contrôleurs RdP capables de traiter une large classe de problèmes pour tout type de modèle RdP en présence d'évènements non contrôlables. Un tel contrôle est très avantageux dans de nombreux systèmes réels comme les systèmes de production par exemple. Cependant, dans ce mémoire, nous avons adopté quelques hypothèses. Premièrement, nous supposons que toutes les transitions sont observables. Or, dans la réalité, tous les évènements dans un SED ne sont pas forcément observables. Ensuite, l'activation et la désactivation des contrôleurs RdP dans les deux premières approches peuvent forcer certaines règles des réseaux de Petri.

L'aboutissement de cette thèse offre la perspective intéressante de développer les deux approches proposées dans le dernier chapitre, à savoir, la synthèse de contrôleurs sous forme de places RdP, sans utilisation du graphe de marquage, en étudiant la prévention de blocage, grâce aux différentes techniques développées et aux propriétés des RdP.

Bibliographie

- [BBD95] Badouel E., Bernardinello L., et Darondeau P., « Polynomial algorithms for the synthesis of bounded nets », in *TAPSOFT '95: Theory and Practice of Software Development*, P. D. Mosses, M. Nielsen, et M. I. Schwartzbach, Éd. Springer Berlin Heidelberg, 1995, p. 364-378.
- [BBD97] Badouel E., Bernardinello L., et Darondeau P., « The synthesis problem for elementary net systems is NP-complete », *Theor. Comput. Sci.*, vol. 186, n° 1–2, p. 107-134, oct. 1997.
- [BBN95] Boel, R. K. et Ben-Naoum L., et Breusegem V. V., « On forbidden state problems for a class of controlled Petri nets », *IEEE Trans. Autom. Control*, vol. 40, n° 10, p. 1717-1731, oct. 1995.
- [BD98] Badouel E. et Darondeau P., « Theory of regions », in *Lectures on Petri Nets I: Basic Models*, W. Reisig et G. Rozenberg, Éd. Springer Berlin Heidelberg, 1998, p. 529-586.
- [CL08] Cassandras C. G. et Lafortune S., *Introduction to Discrete Event Systems*. Springer Science & Business Media, 2009.
- [DA10] David R. et Alla H., *Discrete, Continuous, and Hybrid Petri Nets*. Springer Science & Business Media, 2010.
- [DES] Design//CPN. – *groupe CPN*. – Université de Aarhus, Danemark, [http : //www.daimi.au.dk/designCPN/](http://www.daimi.au.dk/designCPN/).
- [DX03] Darondeau P. et Xie X., « Linear control of live marked graphs », *Automatica*, vol. 39, n° 3, p. 429-440, mars 2003.
- [ECM95] Ezpeleta J., Colom J. M., et Martinez J., « A Petri net based deadlock prevention policy for flexible manufacturing systems », *IEEE Trans. Robot. Autom.*, vol. 11, n° 2, p. 173-184, avr. 1995.
- [ER90a] Ehrenfeucht (A.) et Rosenberg (G.) – Partial (set) 2- structures; part1: Basic notions and the representation problem. *Acta Informatica*, vol. 27, 1990.
- [ER90b] Ehrenfeucht (A.) et Rosenberg (G.) – Partial (set) 2- structures; part2: State spaces of concurrent systems. *Acta Informatica*, vol. 27, 1990.
- [GDS92] Giua A., DiCesare F., et Silva M., « Generalized mutual exclusion constraints on nets with uncontrollable transitions », in *IEEE International Conference on Systems, Man and Cybernetics, 1992*, 1992, p. 974-979 vol.2.
- [GDS93] Giua (A.), DiCesare (F.) et Silva (M.). – Petri net supervisors for generalized mutual exclusion constraints. In: *Proceeding of 12th IFAC World Congress*, pp. 267-270. Sidney, Australia, 1993.
- [Giu 96] Giua (A.) – Petri net techniques for supervisory control of discrete event systems. In: *Proceeding of 1st Int. Workshop on Manufacturing and Petri nets*, pp. 1-30. – Osaka, Japan, 1996.
- [GRX02a] Ghaffari A., Rezg N., et Xie X., « Algebraic and geometric characterization of Petri net controllers using the theory of regions », in *Sixth International Workshop on Discrete Event Systems, 2002. Proceedings*, 2002, p. 219-224.

- [GRX02b] Ghaffari A., Rezg N., et Xie X., « Live and Maximally Permissive Controller Synthesis Using Theory of Regions - Springer ». http://link.springer.com/chapter/10.1007/978-1-4757-6656-1_10.2002.
- [GRX02c] Ghaffari A., Rezg N., et Xie X., « Maximally permissive and non-blocking control of Petri nets using theory of regions », in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, 2002, vol. 2, p. 1895-1900 vol.2.
- [GRX03a] Ghaffari A., Rezg N., et Xie X., « Design of a live and maximally permissive Petri net controller using the theory of regions », *IEEE Trans. Robot. Autom.*, vol. 19, n° 1, p. 137-141, févr. 2003.
- [GRX03b] Ghaffari A., Rezg N., et Xie X., « Feedback control logic for forbidden-state problems of marked graphs: application to a real manufacturing system », *IEEE Trans. Autom. Control*, vol. 48, n° 1, p. 18-29, janv. 2003.
- [GW00] Gohari P. et Wonham W. M., « On the complexity of supervisory control design in the RW framework », *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 30, n° 5, p. 643-652, oct. 2000.
- [HMU07] Hopcroft, J., Motwani, R. & Ullman, J. *Introduction to Automata Theory, Languages, and Computation, 3rd Edition*, Prentice Hall. 2007.
- [HP11] Pan Y.-L. et Huang Y.-S., « An improved maximally permissive deadlock prevention policy based on the theory of regions and reduction approach », *IET Control Theory Appl.*, vol. 5, n° 9, p. 1069-1078, juin 2011.
- [IA07] Iordache M. et Antsaklis P. J., *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*. Springer Science & Business Media, 2007.
- [IWZA13] Iordache, M., Wu, P., Zhu, F., and Antsaklis. Efficient design of Petri-net supervisors with disjunctive specifications. In *Proceedings Of IEEE International Conference on Automation Science and Engineering*. Madison, WI: IEEE. 2013
- [Kro87] Krogh B. H., "Controlled Petri nets and maximally permissive feedback logic", *25th Annual Allerton Conference*, University of Illinois, Urbana, pp. 317-326, 1987.
- [KSTW10] Kroening D., Sharygina N., Tsitovich A., et Wintersteiger C. M., « Termination Analysis with Compositional Transition Invariants », in *Computer Aided Verification*, T. Touili, B. Cook, et P. Jackson, Éd. Springer Berlin Heidelberg, 2010, p. 89-103.
- [LS87] Leveson N. G. and Stolzy J. L., "Safety analysis using Petri nets", *IEEE Trans. on Software. Eng.*, vol, 13, no, 3, nar, 1987.
- [LW93] Li Y., Wonham W. M., "control of vector discrete-event systems-part I: The base model," *IEEE Transactions on Automatic control*. vol. 3g. pp. I 21 5-1 227, Aottt 1993.
- [LW94] Li Y., Wonham W. M., "control of vector discrete event systems II - Controller synthesis" *IEEE Trans. on Automatic control*, vol. 39, no 3, pp. 512-531, 1994.
- [LZJ08] Li, Z.W., Zhou, M.C., & Jeng, M.D. A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions. *IEEE Transactions on Automation Science and Engineering*, 5(1), 182–

188. 2008
- [PCF08] Piroddi, L., Cordone, R., & Fumagalli, I. Selective siphon control for deadlock prevention in Petri Nets. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(6), 1337–1348. 2008
- [Pet81] Peterson (J. L.). – *Petri net theory end the modeling of systems*. – Prentice - Hall, 1981.
- [PR00] Park, J., & Reveliotis, S.A. Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems. *IEEE Transactions on Robotics and Automation*, 16(2), 190–195. 2000.
- [Pro01] Projet PARAGRAPHIE. – Rapport d’activité, INRIA. – 2001. Accessible sous [http : //www.inria.fr/rapportsactivite/RA2001/paragraphe_tf.html](http://www.inria.fr/rapportsactivite/RA2001/paragraphe_tf.html).
- [PX96] Proth (J.M.) et Xie (X.L). – *Petri nets : A tool for design and management of manufacturing systems*. – John Wiley, 1996.
- [QLZKM] Qin, M., Li, Z., Zhou, M.C., Khalgui, M., & Mosbahi, O. Deadlock prevention for a class of Petri nets with uncontrollable and unobservable transitions. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 42(3), 727–738. 2012
- [RAR14] Rezig S., Achour Z., et Rezg N., « Control synthesis based on reachability graph with minimal cuts: Application to a flexible manufacturing system », in *2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, p. 1-6.
- [RARK14] Rezig, S., Achour, Z., Rezg, N., & Kammoun, M-A, Optimal control synthesis for a flexible manufacturing system based on minimal cuts. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (254–258). Bandar Sunway: IEEE. 2014
- [RAR15] Rezig S., Achour Z., et Rezg N., « Optimal Control Synthesis for Flexible Manufacturing System Based on Canonic Markings», in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.
- [RARK15] Rezig S., Achour Z., Rezg N., et Kammoun M.-A., « Supervisory control based on minimal cuts and Petri net sub-controllers coordination », *Int. J. Syst. Sci.*, vol. 0, n° 0, p. 1-11, août 2015
- [RAR16] Rezig S., Achour Z., et Rezg N., « Control Synthesis Based On Theory of Regions with Minimal Reachability Graph Knowledge», in *8th IFAC Conference on Manufacturing Modelling, Management and Control (MIM)*, 2016.
- [RCGH12] Ru, Y., Cabasino, M. P., Giua, A., & Hadjicostis, C.N. Supervisor synthesis for discrete event systems under partial observation and arbitrary forbidden. 2012
- [Rio12] Riordan, J. Introduction to combinatorial analysis. *Courier Corporation*. 2012
- [RRKA14] Rezg, N., Rezig, S., Kammoun, M.-A., & Achour, Z. Control synthesis for admissible paths based on theory of regions: Application to a flexible manufacturing system. In *2014 IEEE Conference on Control Applications (CCA)* (pp. 1807–1811). Juan Les Antibes: IEEE, 2014
- [RW87] Ramadge P. et Wonham W., « Supervisory Control of a Class of Discrete

- Event Processes », *SIAM J. Control Optim.*, vol. 25, n° 1, p. 206-230, janv. 1987.
- [UGT15] Uzam M., Gelen G., et Saleh T. L., « Think-globally-act-locally approach with weighted arcs to the synthesis of a liveness-enforcing supervisor for generalized Petri nets modeling FMSs », *Inf. Sci.*
- [Uzam09] Uzam M., « On suboptimal supervisory control of Petri nets in the presence of uncontrollable transitions via monitor places », *Int. J. Adv. Manuf. Technol.*, vol. 47, n° 5-8, p. 567-579, août 2009.
- [UZ07] Uzam, M., & Zhou, M. An iterative synthesis approach to Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(3), 362–371. 2007
- [RW89] Ramadge P. et Wonham W., « The control of discrete event systems », *IEEE Proceedings*, vol. 27, n° 1, pp. 81-98, janv. 1989.
- [WR87] Wonham W. et Ramadge P., « On the Supremal Controllable Sublanguage of a Given Language », *SIAM J. Control Optim.*, vol. 25, n° 3, p. 637-659, mai 1987.
- [YMLA96] Yamalidou K., Moody J., Lemmon M., et Antsaklis P., « Feedback control of petri nets based on place invariants », *Automatica*, vol. 32, n° 1, p. 15-28, janv. 1996.

Approches Canoniques pour la Synthèse des Contrôleurs Réseaux de Petri

Résumé

Dans ce mémoire, nous présentons différentes approches de synthèse de contrôleurs réseaux de Petri en se basant sur la théorie des régions. Cette théorie présente quelques limites dans la synthèse de contrôle. En effet, la synthèse du contrôleur RdP, s'il existe, n'est pas du tout une tâche facile vue sa complexité de calcul et l'explosion combinatoire des états dans le graphe de marquage. De plus, le système linéaire de la théorie des régions à résoudre peut contenir des combinaisons convexes entre ces équations ce qui rend la théorie insoluble pour calculer les superviseurs RdP. Ce travail vise à simplifier la complexité de calcul de la théorie des régions en réduisant le nombre d'équations du système linéaire de la théorie des régions d'une part, et d'autre part en minimisant le temps de calcul des contrôleurs RdP. De nouveaux concepts de coupes minimales et de marquages canoniques ont été introduits afin d'appliquer la théorie des régions sur des zones précises du graphe et non pas sur la totalité du graphe de marquage. Finalement, deux autres nouvelles approches ont été développées pour synthétiser des contrôleurs RdP sans générer le graphe de marquage.

Abstract

In this work, we present different control synthesis approaches based on Petri nets and the theory of regions. This theory has some limitations in supervisory control. Indeed, the design on the PN controller, if it exists, is not an easy task due to the resolution complexity and the combinatorial explosion of states in the generated reachability graph. In addition, the linear system of the theory of regions may contain convex combinations of its equations making the theory insoluble. This work aims to simplify the computational complexity of the theory of regions by reducing the number of equations of the linear system and decreasing the computation time of PN controllers. Consequently, new concepts of minimal cuts and canonic markings are introduced in order to apply the theory of regions on specific zones of the graph and not on the whole reachability graph. Finally, two new approaches are developed to synthesize PN controllers without generating the reachability graph.
