

THÈSE

Présentée pour obtenir le grade de

Docteur de l'Université de Reims Champagne Ardenne

Spécialité

Génie Informatique, Automatique et Traitement de signal

Identification des systèmes à événements discrets par réseaux de Petri

Par

Rebiha BERGHOUTE BEKRAR

Soutenue publiquement le :31 mars 2009

devant le jury composé de :

Dimitri LEFEBVRE	Professeur à l'Université du Havre	Rapporteurs
Abdellah EL-MOUDNI	Professeur à UTBM, Belfort	
Jean Jacques LOISEAU	DR CNRS à l'IRCCyN, Nantes	Examineurs
Farouk YALAOUI	Professeur à l'UTT, Troyes	
Nadhir MESSAI	Maître de Conférences à l'URCA, Reims	
Najib ESSOUNBOULI	Maître de Conférences à l'IUT de Troyes	
Abdelaziz HAMZAOUI	Professeur à IUT de Troyes	Directeur de thèse
Bernard RIERA	Professeur à l'URCA, Reims	Co-directeur de thèse

Remerciements

Ce travail a été réalisé à l'Institut Universitaire de Technologie de Troyes (IUT) au sein du laboratoire CReSTIC (Centre de Recherche en Sciences et Techniques de l'Information et de la Communication). J'adresse mes plus vifs remerciements à Monsieur Janan ZAYTOON, directeur du CReSTIC, de m'avoir accueilli.

Je tiens à exprimer particulièrement ma profonde gratitude à Monsieur Abdelaziz HAMZAOU, directeur et Professeur à l'IUT de Troyes, qui a dirigé ma thèse et à Monsieur Bernard RIERA, professeur à l'Université de Reims Champagne Ardenne, d'avoir codirigé ce travail.

Je suis également très reconnaissante envers Mrs. Nadhir MESSAI et Najib ESSOUNBOULI, Maîtres de conférences à l'Université de Reims Champagne-Ardenne, les encadrants de cette thèse, pour leurs disponibilités, leur patience, leurs nombreux et précieux conseils ainsi que leur soutien tout au long de cette thèse.

J'adresse mes profonds remerciements à Monsieur Dimitri LEFEBVRE, Professeur à l'Université du Havre, et à Monsieur Abdellah ELMOUDNI, Professeur à l'Université de Technologie Belfort-Montbéliard, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de ce travail. Je leur suis très reconnaissante pour les remarques et les suggestions qu'ils m'ont apportées.

Je tiens à remercier également Monsieur Jean Jacques LOISEAU, Professeur à l'Ecole Centrale de Nantes, et Monsieur Farouk YALAOU, Professeur à Université de Technologie de Troyes, d'avoir accepté d'examiner ma thèse et de faire partie du jury de soutenance.

J'exprime mes reconnaissances à tous les membres de l'équipe Automatique de CReSTIC et à tous mes collègues à l'IUT de Troyes qui m'ont aidé et soutenu au quotidien.

Un grand merci à mon mari Abdelghani qui m'a aidé, soutenu et supporté tout au long de la période de la thèse. Tout ce que je dis ne suffit pas pour lui rendre une partie de ce qu'il a fait pour moi.

Enfin, je ne saurais terminer ces remerciements sans oublier à adresser mes remerciements à toute ma famille et ma belle famille et tout particulièrement mes parents qui m'ont toujours fait confiance et soutenu.

Table des matières

Table des matières	1
Introduction	5
1 Identification des systèmes à événements discrets : État de l'art	9
1.1 Introduction	10
1.2 Les outils d'identification d'un SED	10
1.2.1 Les automates à états finis	11
1.2.2 Les réseaux de Petri	14
1.3 Classification des méthodes d'identification	21
1.4 Approches d'identification d'un SED	22
1.4.1 Approches à base d'automates	22
1.4.2 Approches à base de RdP	28
1.5 Conclusion	42
2 Identification d'une classe des SED par réseaux de Petri	45
2.1 Introduction	46
2.2 Introduction à l'optimisation discrète	47
2.3 Classe de systèmes à identifier	51
2.4 Proposition d'une approche d'identification à base de signaux de sortie	54
2.4.1 Définition du problème	54
2.4.2 Détermination de l'ensemble des transitions du système	55
2.4.3 Etude de l'unicité de la solution	56
2.4.4 Cas général	61
2.4.5 Analyse de l'approche proposée	70
2.5 Proposition d'une approche d'identification à base de signaux d'E/S mesurables	71
2.5.1 Définition du problème	72
2.5.2 Résolution du problème d'identification	72
2.5.3 Analyse de l'approche proposée	79
2.6 Conclusion	81

3	Synthèse d'un contrôleur RdP pour résoudre le problème d'états interdits	83
3.1	Introduction	84
3.2	Réseaux de Petri et contrôle	85
3.2.1	Problème de transitions d'état interdites	85
3.2.2	Problème d'états interdits	86
3.3	Approches de synthèse d'un contrôleur RdP	87
3.3.1	Approches utilisant les contraintes linéaires	88
3.3.2	Approches utilisant la théorie des régions	92
3.3.3	Discussion	97
3.4	Synthèse d'un contrôleur RdP à base de chemins d'influence	98
3.4.1	Identification des marquages interdits	100
3.4.2	Détermination de l'ensemble des transitions d'état interdites	101
3.4.3	Synthèse d'un contrôleur RdP : Algorithme 3	102
3.4.4	Exemple illustratif 1	104
3.5	Synthèse d'un contrôleur RdP à base de séquences de transitions interdites	107
3.5.1	Détermination de l'ensemble des séquences de transitions interdites	107
3.5.2	Synthèse d'un contrôleur RdP : Algorithme 4	109
3.5.3	Exemple illustratif 2	111
3.6	Conclusion	113
4	Application : La machine "PRODUCTIS"	115
4.1	Introduction	116
4.2	Présentation de la machine "PRODUCTIS"	116
4.2.1	Poste 1 : Distribution de comprimés blancs par comptage	117
4.2.2	Poste 2 : Fermeture des gros flacons par bouchonnage	118
4.2.3	Poste 3 : distribution de comprimés verts par comptage	119
4.2.4	Poste 4 : fermeture des petits flacons par bouchonnage	120
4.2.5	Poste 5 : chargement des palettes	120
4.3	Description et modélisation du système à identifier : approche boîte blanche	122
4.4	Application de la première méthode d'identification	123
4.5	Application de la deuxième méthode d'identification	131
4.5.1	Application du premier algorithme de synthèse d'un contrôleur RdP	138
4.5.2	Application du deuxième algorithme de synthèse d'un contrôleur RdP	140
4.6	Conclusion	143
	Conclusion générale	145

Annexe	149
4.7 Les données d'identification	149
4.8 Les marquages atteignables par le RdP identifié et son graphe de marquages	152
Table des matières	164
Table des figures	164
Bibliographie	176

Introduction générale

Les systèmes à événements discrets (SED) sont des systèmes dynamiques dont l'état évolue avec l'occurrence d'événements. En effet, au lieu de s'intéresser au déroulement continu des phénomènes, on ne prend en considération que leurs débuts et leurs fins ainsi leur enchaînement dynamique. Le temps dans les SED n'est codé que par la suite des événements, il est donc discrétisé. On peut citer, comme exemples de SED, les réseaux de communications, les réseaux de transport et les systèmes de production manufacturière.

L'étude des SED peut être menée avec différents outils tels que : les langages de programmation parallèle/temps réel, les modèles dynamiques algébriques, comme l'algèbre Max Plus, les automates à états finis et les réseaux de Petri (RdP). Ces derniers sont choisis pour mener le travail présenté dans cette thèse. Caractérisés par un ensemble de propriétés très importantes, les RdP constituent un outil de modélisation particulièrement bien adapté aux systèmes qui comportent du parallélisme et de la synchronisation. Grâce à leur aspect mathématique, les RdP peuvent être utilisés pour vérifier des propriétés du système comme la vivacité et le blocage.

Généralement la modélisation d'un système est basée sur les lois physiques régissant son comportement et/ou sur des signaux d'entrée et de sortie. Ainsi, on distingue deux grandes classes de modèles : les modèles de connaissance et les modèles de représentation. En effet, les processus décrits par des équations faisant intervenir les lois physiques qui régissent leurs comportements sont appelés modèles de connaissance. Ces modèles, aussi appelés modèles boîte blanche, possèdent l'avantage d'avoir des paramètres directement liés aux éléments physiques du système. Ainsi, dans le cas où la structure du modèle peut être construite avec des lois physiques, mais un où plusieurs paramètres doivent être estimés à partir des observations, les modèles sont appelés modèles boîte grise. Enfin, les systèmes complexes ou mal connus sont souvent équipés d'un grand nombre de capteurs, qui génèrent des signaux pouvant être utilisés pour l'identification d'un modèle à partir de données expérimentales. On appelle un modèle ainsi obtenu modèle de représentation ou modèle boîte noire.

La modélisation présente une étape importante dans l'étude des systèmes indus-

triels. En effet, l'étude de n'importe quel type de systèmes nécessite un modèle. Pour ce faire, la modélisation est à envisager et parmi ses techniques l'identification. Cette dernière est une méthodologie de modélisation mathématique qui s'appuie sur les observations et les connaissances disponibles ([HG06]). Dans ce cadre, les modèles issus de l'identification sont appelés *modèles boîte noire* lorsqu'ils ne sont pas inspirés de la connaissance physique sur le procédé considéré. Ils sont appelés *modèles boîte grise*, lorsque leurs structures peuvent être construites avec des lois physiques, mais plusieurs paramètres doivent être estimés à partir des observations. Finalement, ils sont appelés *modèles boîte blanche*, lorsque leurs paramètres sont directement liés aux éléments constituant le système.

L'identification est une technique largement utilisée en automatique des systèmes continus ([Lju87]). Cependant, elle est loin d'être entièrement explorée dans le cadre des SED. Généralement, les méthodes d'identification des SED décrivent le système comme un langage régulier et cherchent à identifier la structure de l'automate ou du RdP qui accepte ce langage [Vee81], [Gou02]. Cependant, les travaux existants sont peu nombreux et le domaine est loin d'être entièrement exploré, ce qui nous motive à l'aborder.

Notre objectif à travers cette thèse est d'explorer des propriétés graphiques et mathématiques des RdP pour développer des approches d'identification d'une classe de SED. Bien évidemment, cette dernière représente un grand nombre de systèmes industriels.

Dans ce travail, les SED considérés sont particulièrement des systèmes manufacturiers composés de deux parties opérative (PO) et commande (PC) fonctionnant en boucle fermée. La partie opérative est supposée composée d'un ensemble de capteurs et d'actionneurs discrets. Les signaux d'entrée et de sortie utilisés pour identifier le système sont récupérés à l'aide des capteurs et des actionneurs équipant le système. Par ailleurs, nous considérons que l'on ne dispose pas de la description physique du SED à identifier et que ce dernier est caractérisé par des événements détectables.

Le manuscrit a la structure suivante. Le premier chapitre présente les principaux travaux existant se rapportant à l'identification d'un SED. Nous commençons, tout d'abord, par une présentation générale des SED. Ensuite, nous posons le problème d'identification et nous exposons les deux outils les plus usuels dans le champ de l'identification des SED, à savoir les automates à états finis et les RdP. Nous mettons l'accent sur les RdP dans la mesure où c'est l'outil utilisé dans cette thèse. Puis, nous décrivons quelques approches d'identification d'un SED à base d'automates à états finis. Nous présentons en détail les différentes méthodes proposées pour résoudre ce problème en utilisant les RdP.

Dans le second chapitre, nous proposons d'abord une première approche d'identification d'un SED par RdP. Cette approche est basée sur l'utilisation des signaux de sortie du système considéré. Nous commençons par l'élaboration des conditions nécessaires et suffisantes pour l'unicité de la solution. Puis, nous introduisons nos algorithmes, qui permettent de fournir les solutions faisables et de sélectionner la solution optimale dans le cas le plus général. Dans la seconde partie de ce chapitre, nous présentons une deuxième approche basée sur l'analyse des signaux, d'entrée et de sortie, mesurables du système considéré, nous définissons le problème d'identification sous la forme d'un problème d'optimisation et nous utilisons la programmation linéaire binaire pour sélectionner la solution optimale. Nous montrons enfin que, bien que cette approche soit pertinente, les modèles issus de la phase d'identification peuvent, dans certains cas, générer des comportements indésirables.

Le troisième chapitre vise à inhiber les comportements indésirables que l'on peut avoir à la fin de la phase d'identification. Pour ce faire, nous proposons de synthétiser des contrôleurs garantissant un comportement désiré. Ainsi, le problème de synthèse d'un contrôleur à base de RdP est présenté dans un premier temps. Ensuite, les principaux travaux qui se rapportent à la synthèse d'un contrôleur RdP sont exposés. Nous terminons ce chapitre par la proposition de deux approches de synthèse de contrôleurs RdP.

Le quatrième chapitre présente les résultats de validation des algorithmes proposés sur un système réel : la machine "PRODUCTIS" installée à l'IUT de Reims. Ce système est un dispositif destiné à exploiter, gérer, régler et piloter un système intégré de conditionnement de comprimés. Tout d'abord, nous présentons la machine avec ses différents postes où nous décrivons comment on récupère les données de l'identification. Ensuite, nous analysons les données récupérées et nous appliquons nos algorithmes proposés dans les chapitre 2 et 3 afin d'élaborer le modèle RdP de ce système. Nous finirons, par une analyse des résultats obtenus pour montrer leur efficacité en termes de calcul et de la nature de la solution trouvée. Enfin, nous terminerons le manuscrit par des conclusions et perspectives.

Chapitre 1

Identification des systèmes à événements discrets : État de l'art

Résumé : Dans ce chapitre, nous présentons des notions de base sur les systèmes à événements discrets (SED). Nous commençons par une introduction où nous exposons d'une façon générale le problème d'identification des SED. Puis, nous présentons brièvement les deux outils généralement utilisés dans le cadre des SED : les automates à états finis et les Réseaux de Petri(RdP). Nous mettons particulièrement l'accent sur les RdP dans la mesure où c'est l'outil utilisé dans cette thèse. Nous introduisons, ensuite, quelques approches d'identification d'un SED à base d'automates à états finis. Nous exposons en détail les différentes méthodes proposées pour résoudre ce problème en utilisant les réseaux de Petri (RdP). Nous finirons par des conclusions sur les directions de recherche que nous exploiterons dans le prochain chapitre.

1.1 Introduction

L'évolution des systèmes physiques se modélise de trois façons différentes : continue, discrète et hybride. Contrairement aux systèmes discrets qui prennent en compte les informations que dans des moments précis, les systèmes continus mettent en jeu les signaux continus. Quand à eux, les systèmes hybrides considèrent à la fois des signaux continus et des signaux discrets. Généralement, la modélisation d'un système est basée sur les lois physiques régissant son comportement et/ou sur des signaux d'entrée et de sortie. On peut distinguer trois types de modèles de systèmes : *modèles boîte blanche*, *modèles boîte grise* et *modèle boîte noire*. Les *modèles boîte blanche* possèdent l'avantage d'avoir des paramètres directement liés aux éléments constituant le système. Ils sont complètement inspirés de la connaissance physique du système. Dans le cas où la structure du modèle peut être construite avec des lois physiques, mais plusieurs paramètres doivent être estimés à partir des observations, les modèles sont appelés *modèles boîte grise*. Dans le cas où la connaissance du fonctionnement du système est très restreinte, les paramètres de son modèle sont déterminés parmi une classe de modèles assez générale, les modèles sont dans ce cas appelés *modèles boîte noire*.

Pour les SED, les techniques de modélisation appliquées reposent souvent sur une analyse physique du système. L'objectif de cette thèse est d'utiliser la technique d'identification pour obtenir le modèle du système considéré. Cette technique vise à élaborer un modèle comportemental à partir d'observations expérimentales des signaux de sortie qu'il produit en réponse à une variation de ses signaux d'entrée. En effet, la maîtrise d'une telle technique permet d'élaborer un modèle interne d'un système à partir de son modèle externe. Notons que, le modèle interne d'un SED est un modèle formel décrivant l'évolution des sorties du système en fonction de l'évolution des entrées. Ce modèle prend généralement la forme d'un automate à états finis ou d'un réseau de Petri et représente une image du modèle externe. Ce dernier est composé des informations mesurables du système et il est, généralement représenté par une (ou plusieurs) séquence(s) comportant une succession d'évolution des signaux d'entrée et de sortie.

Dans cette thèse nous nous intéressons au problème d'identification d'un SED par RdP. Pour cette raison et dans ce qui suit, nous essayons de faire une synthèse des principaux travaux qui portent sur ce problème et nous mettons l'accent sur les approches d'identification à base de RdP.

1.2 Les outils d'identification d'un SED

Généralement, les travaux qui portent sur l'identification des SED utilisent, comme outil, les automates à états finis et les RdP. Dans cette section, des notions de base

relatives à ces deux outils sont présentées.

1.2.1 Les automates à états finis

La technique la plus ancienne et la plus classique pour représenter les comportements dynamiques d'un SED est d'utiliser un automate à états finis. Un automate est une machine à états constituée d'états et de transitions associées à des événements. Son comportement dépend du mot fourni en entrée : l'automate passe d'état en état, suivant les transitions, à la lecture de chaque symbole de l'entrée. L'automate est dit "fini" car il possède un nombre fini d'états distincts.

Définition 1 : Un automate fini A est un quintuplet $A = (Q, \Sigma, \delta, q_0, Q_m)$ où :

Q est l'ensemble fini des états.

Σ est un ensemble fini de symboles d'entrée (l'ensemble des événements), appelé alphabet d'entrée.

δ est la fonction de transition d'états $\delta : Q \times \Sigma \rightarrow Q$ qui associe un état d'arrivée à un état de départ et un symbole d'entrée.

$q_0 \in Q$ est l'état initial.

$Q_m \subseteq Q$ est l'ensemble des états marqués qui définissent les états finaux.

Un état $q \in Q$ est dit accessible s'il existe une chaîne $s \in \Sigma^*$ telle que $q = \delta(q_0, s)$, cela signifie 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 fini A est donné par :

$$L(A) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}.$$

C'est un langage régulier qui représente l'ensemble de toutes les chaînes permettant de rejoindre un état quelconque de l'automate à partir de son état initial.

Exemple 1 [Zay01] : Considérons l'exemple classique d'une souris qui se déplace d'une manière spontanée à l'intérieur d'un labyrinthe comme montré sur la figure 1.1. Les salles S_i communiquent par des portes unidirectionnelles P_1 et P_3 et bidirectionnelle P_2 . On note " p_i " l'événement : "la souris passe par la porte P_i ". Soit Σ l'ensemble des événements : $\Sigma = \{p_1, p_2, p_3\}$. Les situations possibles sont :

- la souris est dans la salle S_0 .
- la souris est dans la salle S_1 .
- la souris est dans la salle S_2 .

Si l'on considère que la souris commence par se déplacer depuis la salle S_0 et revient finalement à cette salle, ce système peut être représenté par l'automate à états finis décrit dans la figure 1.2. Cet automate est constitué de trois états $\{q_0, q_1, q_2\}$ qui représentent les différentes situations visitées par la souris. L'état initial q_0 , indique la

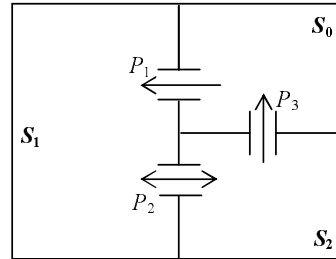


FIGURE 1.1 – Un exemple d'un SED

position initiale de la souris depuis laquelle elle peut effectuer les différents parcours. On considère, dans cet exemple, que l'état final et l'état initial sont identiques. L'ensemble de symboles d'entrée est $\Sigma = \{p_1, p_2, p_3\}$, indique le passage de la souris entre les portes et sa fonction de transitions est définie par : $\delta = \{\delta(q_0, p_1) = q_1, \delta(q_1, p_2) = q_2, \delta(q_2, p_2) = q_1, \delta(q_2, p_3) = q_0\}$.

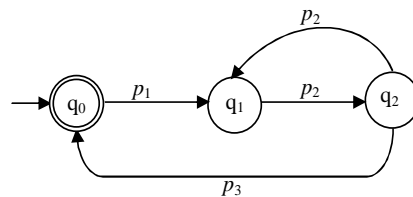


FIGURE 1.2 – Automate à états finis

Remarque : Un automate 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. On remarque que, l'automate de la figure 1.2 satisfait cette condition donc, il est *déterministe*. Contrairement à un automate déterministe, un automate *non déterministe* peut avoir plusieurs états initiaux, et plusieurs transitions de sortie associées au(x) même(s) symbole(s) et conduisent à des états différents.

Les automates à états finis, qu'ils soient déterministes ou non, permettent de décrire le comportement d'un SED au moyen d'états et d'événements. Parmi les types d'automates à états finis utilisés comme outil pour l'identification d'un SED, nous trouvons les automates de Moore [Sud96], [CL99], les automates de Mealy [HMu01], [KR03], [Pin06] et les automates Autonomes non Déterministes avec fonction de Sortie (AANDS) [KLL05], [Kle05].

Un automate de Moore est un automate fini pour lequel les valeurs des variables de sortie ne peuvent dépendre que des variables d'état. On appelle les systèmes modélisés par ce type d'automate des systèmes strictement synchrones car le changement des

sorties ne se fait qu'avec le changement d'état. Quant aux machines de Mealy, les sorties dépendent à la fois de l'état courant et des variables d'entrées. Enfin, un automate autonome non déterministe avec fonction de sortie est un automate non déterministe, comme décrit dans [CL99] où chaque état est associé à un symbole de sortie. Ce type d'automates représente le comportement non déterministe d'un système autonome. Ce type d'automates est défini comme suit :

Définition 2 [Kle05] : Un Automate Autonome Non Déterministe avec fonction de Sortie (AANDS) est un quintuplet $AANDS = (Q, \Omega, f_{nd}, \lambda, q_0)$ tel que :

$Q = \{q_0, q_1, \dots, q_{|Q|-1}\}$ est un ensemble fini d'états.

$\Omega = \{\omega_1, \omega_2, \dots, \omega_{|\Omega|}\}$ est un ensemble fini de symboles de sortie.

$f_{nd} : Q \mapsto 2^Q$ est une relation de transition autonome non déterministe.

$\lambda : Q \mapsto \Omega$ est une fonction de sortie qui associe à chaque état un symbole de sortie.

$q_0 \in Q$ représente l'état initial.

L'AANDS est un graphe biparti dont les sommets représentent les états de cet automate et ses arcs sont représentés par la fonction de transfert. Chaque sommet est associé à une sortie $\lambda(q_i)$ de l'état correspondant. L'automate peut accéder à un état q_j à partir d'un état q_i si et seulement si $q_j \in f_{nd}(q_i)$, c.à.d, s'il y a un arc orienté de q_i vers q_j . Ce type d'automates avec sortie accepte un langage défini par : $L_{q_0}(A) = \{s \in E^* : f(q_0, s) \text{ est définie}\}$ où :

E est un ensemble fini d'événements associés aux transitions de l'automate.

$f : Q \times E \rightarrow Q$ est la fonction de transfert.

E^* est l'ensemble fini des mots composés d'éléments de E , y compris le mot vide ϵ .

Exemple 2 : La figure 1.3 représente un AANDS défini par : $Q = \{q_0, q_1, q_2, q_3\}$ l'ensemble d'états de l'automate avec q_0 est l'état initial, un ensemble fini de symboles de sortie $\Omega = \{0000, 1001, 1010, 1111\}$, une fonction de sortie : $\lambda(q_0) = 0000$, $\lambda(q_1) = 1001$, $\lambda(q_2) = 1010$, $\lambda(q_3) = 1111$, une relation de transfert $f_{nd} : f_{nd}(q_0) = \{q_1, q_2\}$, $f_{nd}(q_1) = \{q_0, q_3\}$, $f_{nd}(q_2) = \{q_0, q_3\}$, $f_{nd}(q_3) = \{q_1, q_2\}$.

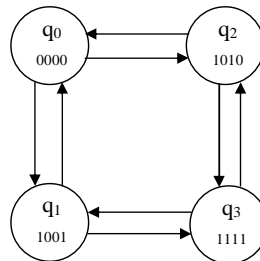


FIGURE 1.3 – Automate autonome non déterministe avec fonction de sortie

1.2.2 Les réseaux de Petri

Dès leur création en 1962 par Carl Adam Petri, les réseaux de Petri représentent un outil graphique et mathématique puissant pour la modélisation des SED. Les modèles obtenus, outre leur représentation graphique de la structure du système, permettent une analyse des propriétés et par conséquent une validation de leurs spécifications. Les réseaux de Petri ont, par rapport aux automates, l'avantage d'être un modèle beaucoup plus général, bénéficiant de structures beaucoup plus riches, s'adaptant parfaitement à la description de différents types de SED [Mur89], [DA99], [GS07]. Dans la suite, des notions de base de RdP et des définitions qui seront utilisées dans ce travail seront présentées. Des propriétés nécessaires pour la présentation de notre contribution seront aussi introduites.

1.2.2.1 Définitions générales

Plusieurs types de RdP ont été utilisés dans la littérature comme outil pour identifier les SED. Parmi les classes de RdP les plus utilisées nous pouvons citer les RdP *free labelled*, les RdP *λ -free labelled*, les RdP *marqués* et les RdP *interprétés*. Dans ce qui suit, nous introduisons les définitions de ces types de RdP qui seront utilisées dans la section (1.4.2) en présentant les approches d'identification d'un SED à base de RdP. Avant de présenter ces définitions, nous introduisons la définition générale d'un RdP.

Un Réseau de Petri (RdP) est un graphe orienté constitué de :

- $P = \{p_1, p_2, \dots, p_m\}$: un ensemble fini de places symbolisées par des cercles et représentant des conditions : une ressource d'un système (ex : une machine, un stock,...), l'état d'une ressource du système (machine libre, stock vide,...).
- $T = \{t_1, t_2, \dots, t_n\}$: un ensemble fini de transitions symbolisées par des traits et représentant l'ensemble d'événements dont l'occurrence provoque un changement de l'état du système.
- Un ensemble fini d'arcs orientés qui assurent la liaison entre les places et les transitions.

De manière plus formelle, un RdP peut être défini comme suit :

Définition 3 : un RdP est un quatre-uplet $(P, T, Post, Pre)$ où [Mur89], [DA99] :

P et T sont respectivement l'ensemble fini et non vide de places et de transitions.

$Post$ est l'application d'incidence arrière définie par :

$Post : P \times T \rightarrow \mathbb{N}$ avec $Post(p_i, t_j)$ le poids de l'arc entre la place p_i et la transition t_j .

Pre est l'application d'incidence avant définie par :

$Pre : P \times T \rightarrow \mathbb{N}$ avec $Pre(p_i, t_j)$ le poids de l'arc entre la transition t_j et la place p_i .

Dans ce mémoire, les notations suivantes sont adoptées :

- *t_j : l'ensemble des places en amont de la transition t_j .
- t_j^* : l'ensemble des places en aval de la transition t_j .
- *p_i : l'ensemble des transitions en amont de la place p_i .
- p_i^* : l'ensemble des transitions en aval de la place p_i .

Les places dans un RdP peuvent contenir des marques (jetons) qui permettent de modéliser la dynamique du système considéré. Le RdP dans ce cas est dit marqué et la distribution des jetons dans les places est appelée "marquage du RdP". Ce dernier est noté $M^{|P|}$ où $M^{|P|}$ est un vecteur dont la i ème composante représente le marquage de la place p_i . Notons, enfin, que le marquage initial d'un RdP est noté M_0 .

Pour illustrer cette définition, nous considérons le RdP de la figure 1.4 qui modélise le système décrit dans l'exemple 1. Ce RdP comporte trois places notées : p_1 , p_2 ,

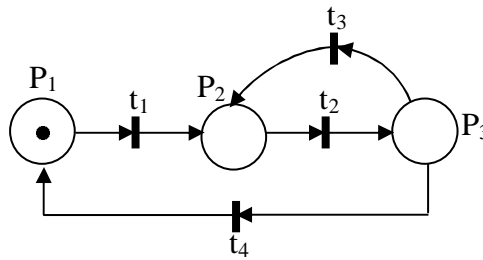


FIGURE 1.4 – Un RdP marqué modélisant le système de l'exemple 1

et p_3 et quatre transitions notées : t_1 , t_2 , t_3 , et t_4 . Les places p_1 , p_2 et p_3 représentent les salles telles que, la place p_1 représente la salle S_0 , p_2 la salle S_1 et p_3 la salle S_2 . Les transitions t_1 , t_2 , t_3 , et t_4 sont associées aux événements indiquant le passage de la souris entre les salles telles que, la transition t_1 est associée au passage de la souris de la salle S_0 à la salle S_1 . La transition t_2 est associée à l'événement indiquant le passage de la souris de S_1 à S_2 et t_3 est associée à celui indiquant le passage dans le sens inverse. Finalement, t_4 est associée à l'événement qui indique le passage de la souris de S_2 à S_0 . Notons que, la marque présente dans la place p_1 correspond à la présence de la souris initialement dans la salle S_0 . Le marquage initial de ce réseau est $M_0 = (1, 0, 0)$. Nous remarquons que, les poids associés aux arcs de ce RdP sont égaux à 1, ce qui implique que ce RdP est ordinaire.

Définition 4 [CGS07] : Un RdP *free labelled* est un triplet $\mathcal{G}(N, M_0, \psi)$ où : $N = (P, T, Post, Pre)$ est un RdP, M_0 est son marquage initial et $\psi : T \rightarrow E \setminus \{\epsilon\}$ est la fonction qui associe à chaque transition un symbole de l'ensemble de symboles E différent de ϵ (mot vide).

Définition 5 [Hir92] : Un RdP marqué est un triplet $\mathfrak{M} = (N, M_0, M_F)$ où : $N = (P, T, Post, Pre)$ est un RdP, M_0 est son marquage initial et M_F est l'ensemble de marquages finaux.

Définition 6 [MRM98] : Un RdP Interprété est un quintuplet RdPI = $(N, \Sigma, \Phi, \lambda, \varphi)$ avec :

N est un RdP de marquage initial M_0 .

$\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_r\}$: un ensemble fini de signaux d'entrée.

$\Phi = \{\phi_1, \phi_2, \dots, \phi_p\}$: un ensemble fini de signaux de sortie.

$\lambda : T \rightarrow \Sigma \cup \{\epsilon\}$: une fonction qui associe à chaque transition un symbole d'entrée.

$\varphi : R(N) \rightarrow \{\Phi \cup \{\epsilon\}\}^q$: fonction de sortie.

Où $R(N)$ est l'ensemble de marquages accessibles depuis le marquage initial et q est la dimension du vecteur de sortie.

1.2.2.2 Règles d'évolution d'un RdP

L'évolution d'un RdP dépend du franchissement de ses transitions. Une transition t_j est validée si et seulement si :

$$M_i \geq Pre(p_i, t_j) \tag{1.1}$$

avec $p_i \in {}^*t_j$ et *t_j représente l'ensemble de places en amont de t_j . Le franchissement d'une transition validée modifie le marquage des places en amont et en aval de cette transition. Le franchissement d'une transition a pour conséquences le retrait d'un nombre de marques égale aux poids des arcs pour les places en amont de cette transition et l'ajout d'un nombre de marques égale aux poids des arcs pour les places en aval. Si le RdP considéré est ordinaire alors, le nombre de marques à ajouter et à retirer est égal à 1. La figure 1.5 illustre le franchissement d'une transition.

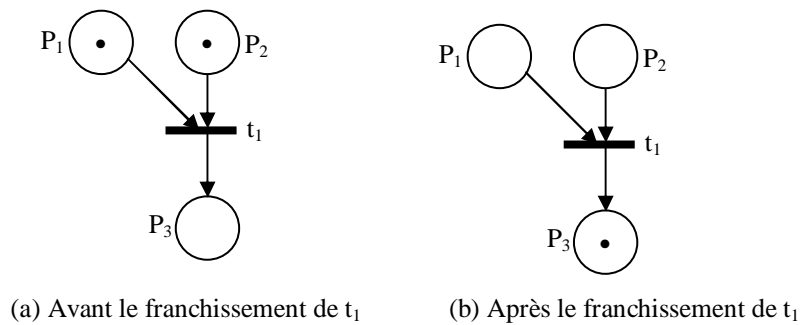


FIGURE 1.5 – Franchissement d'une transition

Nous appelons une séquence de franchissement une suite de transitions qui sont franchissables successivement. A chaque séquence de franchissement, on associe un vecteur caractéristique noté $\bar{\sigma}$. C'est un vecteur de dimension $|n|$ (le nombre de transitions) où, la composante numéro j correspond au nombre de franchissements de la transition t_j dans la séquence σ . Si la séquence de franchissement σ est réalisable à partir de M_i , le marquage obtenu M_k est donné par l'équation fondamentale :

$$M_k = M_i + W \cdot \bar{\sigma} \quad (1.2)$$

où W est la matrice d'incidence du RdP définie par :

$$W = Post - Pre \quad (1.3)$$

où $Post = [Post_{ij}]_{i=1,\dots,m;j=1,\dots,n} \in \mathbb{N}^{m \times n}$ la matrice d'incidence arrière avec $Post_{ij} = Post(p_i, t_j)$ et $Pre = [Pre_{ij}]_{i=1,\dots,m;j=1,\dots,n} \in \mathbb{N}^{m \times n}$ la matrice d'incidence avant avec $Pre_{ij} = Pre(p_i, t_j)$.

Notons que, l'équation fondamentale ne garantit pas que σ soit franchissable mais nous permet simplement de trouver le nouveau marquage lorsqu'on connaît le marquage M_i et la séquence franchissable σ .

Exemple 3 : Soit le RdP de la figure 1.4 qui modélise le système de l'exemple 1. Depuis le marquage initial de ce RdP, on peut avoir les séquences de transitions suivantes : $\sigma_1 = t_1 t_2 t_3$, $\sigma_2 = t_1 t_2 t_4$, $\sigma_3 = t_1 t_2 t_3 t_2 t_4$, $\sigma_4 = t_1 t_2$. Chacune de ces séquences décrit un comportement de la souris auquel on associe un vecteur caractéristique : le vecteur de franchissement. Le vecteur caractéristique de σ_1 est $\bar{\sigma}_1 = [1 \ 1 \ 1 \ 0]^T$, celui de σ_2 est $\bar{\sigma}_2 = [1 \ 1 \ 0 \ 1]^T$, celui de σ_3 est $\bar{\sigma}_3 = [1 \ 2 \ 1 \ 1]^T$ et celui de $\bar{\sigma}_4 = [1 \ 1 \ 0 \ 0]^T$.

La matrice d'incidence arrière de ce RdP est : $Post = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, celle d'incidence avant est : $Pre = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$, et la matrice d'incidence est : $W = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}$.

Le franchissement de la séquence de transitions σ_4 depuis le marquage initial permet

d'atteindre le marquage $M_1 = (1, 0, 0) + \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \cdot [1 \ 1 \ 0 \ 0]^T = (0, 0, 1)$.

1.2.2.3 Quelques propriétés des RdP

Quelques propriétés des RdP qui seront utiles dans la suite sont présentées ci-après [PX95], [DA99].

Accessibilité : Le problème d'accessibilité dans un RdP consiste à vérifier si un marquage M_i est atteignable à partir du marquage initial M_0 . Soit (N, M_0) un RdP, on dit qu'un marquage M_i est accessible (ou atteignable) depuis M_0 s'il existe une séquence de franchissements σ telle que : $M_0[\sigma > M_i$. L'ensemble de marquages atteignables depuis M_0 est noté $R(N, M_0)$. L'ensemble de marquages atteignables depuis le marquage initial M_0 et les passages entre ces marquages sont représentés par un graphe appelé graphe de marquages G (graphe de marquages accessibles). Ce dernier est composé de noeuds correspondant aux marquages accessibles, et d'arcs correspondant aux franchissements de transitions faisant passer d'un marquage à un autre. Notant que, le graphe de marquages d'un RdP est un automate sans états marqués.

Vivacité et blocage : Une transition t_j est dite vivante si elle peut être franchie quelque soit le marquage atteint : $\forall M \in R(N, M_0), \exists M_i \in R(N, M_0)$ telle que t_j soit franchissable pour M_i . De façon plus générale, un RdP (N, M_0) est vivant si chacune de ses transitions est vivante. Ainsi, un RdP est vivant pour un marquage initial M_0 si toutes ses transitions sont vivantes pour M_0 .

Par ailleurs, un marquage M atteignable par un RdP est un blocage si aucune transition n'est franchissable depuis ce marquage. D'autre part, un RdP est sans blocage si aucun de ses marquages atteignables n'est un blocage.

Bornitude : Un RdP (N, M_0) est borné si, quelque soit la place p_i et quelque soit le marquage accessible M_i depuis M_0 , le nombre de jetons dans cette place p_i est borné :

$$\forall M_i \in R(N, M_0), \forall p \in P, \exists k \in \mathbb{N} \text{ tel que } M_i(p) \leq k$$

La réversibilité : Un RdP est dit réversible si, à partir de n'importe quel marquage atteignable M_i , il existe une séquence de transitions franchissables qui permet de revenir au marquage initial M_0 .

Réinitialiabilité et état d'accueil : Un RdP a un état d'accueil M_a pour un marquage initial M_0 si pour tout marquage M_i accessible depuis M_0 , il existe une séquence de franchissement σ tel que M_a est accessible depuis M_i en franchissant σ . Le RdP est dit réinitialisable pour un marquage initial M_0 si M_0 est un état d'accueil.

La connexité et la forte connexité : Un RdP est dit connexe si et seulement si il existe un chemin (non forcément orienté) depuis n'importe quel noeud vers un autre noeud.

Un noeud peut être une place ou une transition. Un RdP est dit fortement connexe si et seulement si il existe un chemin orienté entre chaque paire de noeuds.

Les P-invariants et les T-invariants : La matrice d'incidence nous permet de définir les notions importantes d'invariants :

Un P-invariant : est un vecteur X de dimension m , dont ses éléments sont des entiers, vérifiant :

$$X^T . W = 0 \quad (1.4)$$

Soient X un P-invariant et M_0 un marquage initial d'un RdP quelconque. Pour tout marquage M_i atteignable depuis M_0 la relation suivante est vérifiée :

$$X^T . M_0 = X^T . M_i \quad (1.5)$$

Cette relation déduite à partir de l'équation fondamentale, est une relation de conservation du nombre de marques. Elle exprime qu'une certaine combinaison linéaire entre les jetons figurant dans les places d'un P-invariant pour tous les marquages accessibles depuis M_0 est constante.

Un T-invariant : est un vecteur Y d'entiers positifs et de dimension n vérifiant :

$$W . Y = 0 \quad (1.6)$$

Soit σ une séquence de franchissements et $\bar{\sigma}$ son vecteur caractéristique. Si $\bar{\sigma} = Y$ est un T-invariant et si M_0 est le marquage initial, le marquage M_i obtenu en franchissant la séquence σ est tel que :

$$M_i = M_0 \quad (1.7)$$

Autrement dit que σ est une séquence cyclique.

Deux définitions découlent des définitions précédentes. La définition d'une composante conservative et celle d'une composante répétitive. Un ensemble de places est une composante conservative si et seulement si ses places forment un P-invariant. Un ensemble de transitions forme une composante répétitive si et seulement si elles forment un T-invariant.

1.2.2.4 Les langages générés par les RdP

Dans la littérature, les langages générés par les RdP ont été utilisés pour traiter des problèmes d'identification [Hir92] [GS05], de supervision [IA05][BCG06] et de diagnostic [Lef05][ELL06]. Ci-après, une définition de ces langages est introduite. Cette

définition sera utilisée dans la section (1.4.2) en exposant les approches d'identification des SED à base de RdP qui utilisent ce langage.

Définition 7 : Un générateur RdP est un quadruplet $\mathbf{G} = (N, l, M_0, M_F)$ où N un RdP, $l : T \rightarrow \Sigma$ une fonction qui associe à chaque transition un symbole de l'ensemble de symboles Σ , M_0 le marquage initial et M_F l'ensemble de marquages finaux. La fonction l peut être étendue de façon immédiate à la fonction : $l : T^* \rightarrow \Sigma^*$ avec Σ^* est l'ensemble de tous les mots finis sur l'ensemble des symboles Σ , y compris le mot vide ϵ . Le langage clos d'un générateur RdP est donné par :

$$L(\mathbf{G}) = \{l(\sigma) \in \Sigma^* \mid \sigma \in T^*, M_0[\sigma >]\}$$

et le langage marqué de \mathbf{G} est décrit par :

$$L_l(\mathbf{G}) = \{l(\sigma) \in \Sigma^* \mid \sigma \in T^*, M_0[\sigma > M, M \in M_F]\}$$

Les générateurs RdP représentent une plus large classe de systèmes que les automates à états finis. Un générateur RdP est dit déterministe si la connaissance de la séquence d'événements générée suffit pour déterminer le seul marquage atteignable après le franchissement de cette séquence. Ainsi, un générateur RdP est dit non bloquant si chaque état atteignable par N est aussi co-atteignable. Un état co-atteignable est l'état à partir duquel on peut atteindre un état marqué.

1.2.2.5 Mesurabilité, contrôlabilité et détectabilité dans un RdP

Mesurabilité : Dans un RdP (N, M_0) , un marquage M de l'ensemble de marquages atteignables par ce RdP est dit *mesurable* si et seulement si pour toute place dans l'ensemble de places de ce RdP, le marquage de cette place est connu (i.e., $\forall p \in P : M(p)$ est connu). Par ailleurs, le marquage peut être partiellement mesurable [DFM06] ou complètement non mesurable.

Contrôlabilité : Cette propriété est liée aux transitions du RdP. Une transition t_i est dite contrôlable si le contrôleur peut empêcher son franchissement. Autrement, elle est non contrôlable [Ior03], [Giu96], [GS01].

Détectabilité : Généralement, le changement de l'état d'un SED est dû à l'occurrence des événements. Ainsi, l'occurrence d'un événement peut être détectée à l'aide des symboles de sorties [AIE03], [ABT02]. En terme de RdP, le franchissement d'une transition t_i associée à un événement ne peut pas être détecté du fait qu'il n'y a aucun changement sur le marquage, cette transition est dite non détectable. Un RdP est dit à événements détectables si et seulement si le franchissement de n'importe quelle paire de transitions t_i et t_j est distingué l'une de l'autre par l'observation de marquages en

amont et en aval de ces transitions.

Après avoir présenté les deux outils utilisés pour l'identification d'un SED, une classification des méthodes d'identification d'un SED est introduite dans ce qui suit.

1.3 Classification des méthodes d'identification

La manière d'élaborer un modèle comportemental d'un système dépend : (i) de l'algorithme d'identification, qui peut utiliser soit les signaux d'entrée/sortie observés lors du fonctionnement de système (système en boucle fermé), soit une séquence particulière d'excitation (ii) du niveau de connaissance du modèle à identifier, (iii) du type des signaux d'entrée/sortie récupérées et de la façon dont on analyse ces signaux.

Lorsque le processus d'identification se fait durant le fonctionnement du système, l'identification est dite adaptative. Dans ce cas, le modèle externe ne peut être élaboré qu'en relevant, en ligne, les valeurs des entrées/sorties (E/S) du système. En revanche, lorsque le processus d'identification se fait hors ligne l'identification est dite passive et les séquences d'E/S récupérées sont également passives.

Par ailleurs, une identification, que ce soit en ligne ou hors ligne, peut s'effectuer en exploitant quelques connaissances sur le comportement du système et/ou le modèle à identifier. Elle peut également s'effectuer sans aucune connaissance sur le système et/ou le modèle à identifier. Ainsi, si aucune information a priori n'est valable l'identification est dite absolue sinon, elle est dite relative.

Aussi, une identification est dite simple si les signaux d'E/S sont représentés sous la forme d'une séquence d'E/S unique. Elle est dite multiple si les signaux d'E/S sont représentés sous la forme de plusieurs séquences commencent et se terminent par le même vecteur d'E/S (i.e., le système est dans ce cas supposés être réinitialisable).

L'identification est dite constructive si chacun des vecteurs composant la séquences d'E/S est utilisé afin d'affiner le modèle obtenu à la phase précédant (i.e., la séquence d'E/S est analysé d'une façon séquentielle). Enfin, l'identification est dite énumérative si plusieurs modèles sont construits en analysant les séquences d'E/S d'une façon globale.

L'organigramme de la figure 1.6 représente une classification des différentes méthodes d'identification d'un SED.

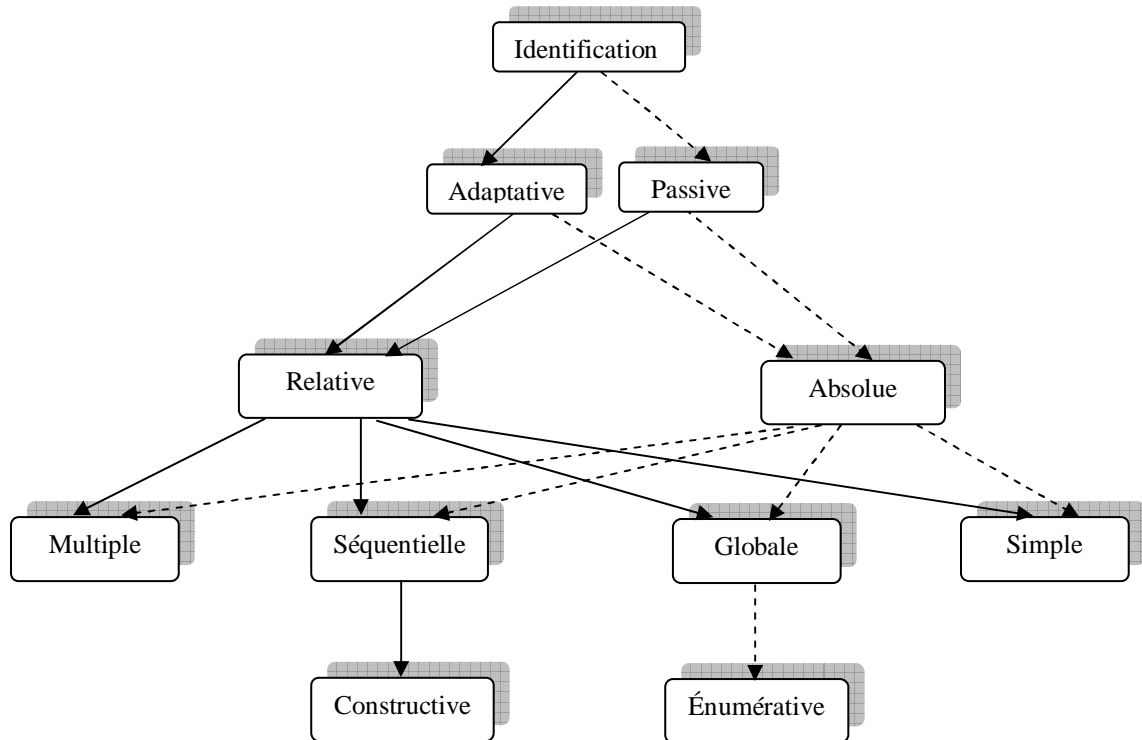


FIGURE 1.6 – Classification des méthodes d'identification d'un SED

1.4 Approches d'identification d'un SED

Les travaux proposés pour résoudre le problème d'identification d'un SED peuvent être classés en deux catégories : approches à base d'automates et d'autres à base de RdP. Quelques approches seront exposées ci-après.

1.4.1 Approches à base d'automates

Les automates à états finis ont été largement utilisés pour résoudre le problème d'identification d'un SED. Généralement, les travaux qui portent sur ce problème décrivent le système comme un langage régulier et cherchent à identifier la structure de l'automate (ou des automates) qui accepte ce langage. Différentes méthodes provenant du domaine de l'informatique ont été proposées. Ces méthodes cherchent à établir les automates de Moore ou de Mealy en analysant le langage généré par le système à identifier, à savoir : l'approche de Kella, l'approche de Biermann et Feldman, l'approche de Booth, l'approche de Veelenturf et celle de Klein *et al.*

1.4.1.1 Approche de Kella

Dans cette approche [Kel71], l'auteur a proposé une méthode d'identification globale et énumérative. Cette méthode ne nécessite pas des connaissances *a priori* sur le système à identifier, donc elle est absolue. Ainsi, une seule séquence observée est disponible ce qui rend la méthode simple. Cette méthode permet d'établir des automates de Mealy complètement spécifiés en utilisant une seule séquence d'E/S σ de longueur n et consiste à :

1. Construire une machine de Mealy originale de $n + 1$ états en exploitant σ .
2. Calculer l'ensemble des états incompatibles.
3. Fusionner les états qui peuvent être fusionnés pour avoir des machines de Mealy d'une taille minimale.

Cependant, le fusionnement des états pour avoir des machines à états minimales risque de causer une perte des informations importantes sur le système.

1.4.1.2 Approche de Biermann et Feldman

Dans cette approche [BF72], les auteurs ont proposé une méthode d'identification globale et énumérative. Cette méthode permet d'établir des machines de Moore non déterministes en analysant un ensemble de séquences d'E/S, qui commencent par le même état initial. Ceci implique que la machine est supposée être réinitialisable et la méthode est dite multiple. L'algorithme d'identification est proposé dans le but de validation et consiste à :

1. Représenter les séquences d'E/S sous la forme d'un arbre d'états.
2. Déduire les différentes machines de Moore minimales en fusionnant les états équivalents d'ordre k . Ce dernier est un paramètre, qui permet d'assurer l'exactitude du modèle identifié et son choix peut influencer sur les comportements générés par l'automate résultant. En effet, si la valeur de ce paramètre est inférieure à la longueur de la plus grande séquence, l'automate établi ne peut pas générer toutes les séquences.

1.4.1.3 Approche de Booth

L'objectif de cette approche est d'identifier des automates de Moore ou de Mealy complètement spécifiés où la réponse de la machine pour chaque entrée doit être définie pour chaque état [Boo67]. Cette approche a l'avantage d'être séquentielle. L'algorithme d'identification proposé est relatif parce qu'il exige la connaissance du nombre maximum d'états dans l'automate à identifier. À partir de l'état initial du système dont la séquence d'entrée choisie arbitrairement est injectée et la séquence de sortie correspondante est utilisée pour construire les automates qui ont au plus n états. Une autre séquence d'entrée est appliquée à partir de l'état initial et la séquence de sortie correspondante est utilisée soit pour mettre à jour ou bien pour éliminer les machines d'états

identifiées précédemment. Notons que, l'élimination des automates partiellement identifiés se fait en respectant le type de l'automate (complètement spécifié). Cette étape est répétée jusqu'à l'obtention des automates à états complètement spécifiés.

Par ailleurs, dans le cas où il est impossible d'établir des automates complètement spécifiés avec un nombre d'états fixé, l'algorithme est exécuté en l'incrémentant. Avec ce processus d'identification itératif, il est possible d'établir des automates fortement connexes, de plus petites tailles et qui représentent exactement les séquences d'E/S. Cependant, pour un système avec un nombre considérable de signaux d'entrée, la génération de séquences d'entrée est plus compliquée et prend un temps plus grand.

1.4.1.4 Approche de Veelenturf

Dans cette approche [Vee81], l'auteur a proposé un algorithme d'identification constructive qui permet d'établir une machine unique de Moore. Cet algorithme consiste à utiliser des séquences d'E/S commençant par le même état initial, ce qui rend le processus d'identification multiple. Il s'exécute en trois étapes principales : (i) l'élaboration des hypothèses du modèle à identifier, (ii) la vérification de l'existence des contradictions entre la sortie prédite par le modèle est la sortie observée, (iii) l'élimination des contradictions détectées. En effet, dans cet algorithme, le calcul des différents états constituant l'automate est basé sur la découverte des contradictions entre les séquences d'E/S. Cependant, cette méthode peut prendre un temps de calcul considérable.

Après avoir cité quelques travaux qui s'intéressent à l'identification d'un automate (des automates) à états finis en analysant une séquence (des séquences) d'un langage régulier, nous présentons, ci-après, une approche utilisant un type spécifique d'automates à états finis.

1.4.1.5 Approche de Klein *et al.*

Dans cette approche [KLL05], l'auteur cherche à établir un modèle comportemental d'une classe particulière de SED sous la forme d'un Automate Autonome Non déterministe avec Fonction de Sortie (AANDS) pour le diagnostic. Le choix de ce type d'automates est motivé par le comportement autonome non déterministe du système considéré. L'algorithme d'identification proposé consiste à définir les différents éléments qui constituent cet automate en utilisant Σ l'ensemble des séquences d'E/S. Le système à identifier est composé de deux parties : opérative et commande fonctionnant en boucle fermée. Il est observé durant ses différents cycles de fonctionnement afin de récupérer ses différents signaux d'entrée et de sortie. Ces derniers sont regroupés dans des vecteurs d'E/S notés $u_i(t) = [i_1(t) \dots i_n(t) \ o_1(t) \dots o_m(t)]^T$ où n et m représentent le nombre de signaux d'entrée et de sortie respectivement. Par ailleurs,

chaque cycle de fonctionnement du système considéré est représenté par une succession ordonnée des vecteurs d'E/S à différents instants t . Cette succession, notée par σ_i , est définie par $\sigma_i = (u_i(1), u_i(2), \dots, u_i(|\sigma_i|))$ avec $|\sigma_i|$ est la longueur de la séquence considérée. Le fait que le système à identifier est cyclique implique que les séquences d'E/S doivent avoir des vecteurs initiaux et finaux identiques.

Initialement, l'automate à identifier est composé d'un seul état q_0 , sans symbole de sortie associé à cet état et sans relation de transition. La première étape de cette approche consiste à construire un AANDS initial en exploitant la première séquence d'E/S observée σ_1 . Si la condition de réinitialiable de cette séquence est vérifiée alors, le premier vecteur $u_1(1)$ de σ_1 est ajouté à Ω l'ensemble de symboles de sortie de cet automate qui était initialement vide. Ainsi, $u_1(1)$ est associé à q_0 l'état initial comme symbole de sortie. Puis, l'algorithme considère le vecteur suivant $u_1(t)$ dans la même séquence σ_1 . Si $u_1(t)$ est déjà dans l'ensemble Ω , l'algorithme cherche l'état auquel il était associé et poursuit le processus, sinon, il est ajouté à Ω . Un nouvel état q_i avec un symbole de sortie égal à $u_1(t)$ est considéré. Si le vecteur de sortie associé à un état q_j est le successeur du vecteur de sortie associé à un état q_i alors la relation de transition de q_i est définie ($f_{nd}(q_i) = q_j$) et l'état q_i est relié à l'état q_j . L'opération d'analyse de vecteurs de la séquence σ_1 est répétée jusqu'au dernier vecteur dans cette séquence. A la fin de cette étape, un automate initial est établi avec : un état initial, un nombre d'états égal au nombre de vecteurs distincts dans la séquence σ_1 et un ensemble de symboles de sortie Ω , où chaque symbole est associé à un état et une relation de transition entre les états de cet automate. Cette étape est résumée dans la procédure suivante :

Données : σ_1 , la séquence d'E/S initiale.

Objectif : construire l'AANDS initial.

Début

Initialisation

1. Créer un AANDS($Q, \Omega, f_{nd}, \lambda, q_0$) avec $Q = \{q_0\}$, $\Omega = \emptyset$, $\lambda(q_0) = \emptyset$ et $f_{nd}(q_0)$ est indéfinie.
2. Si $u_1(1) = u_1(|\sigma_1|)$ alors aller à l'étape suivante
Sinon, $\Sigma := \Sigma \setminus \{\sigma_1\}$.
3. $\Omega := \Omega \cup \{u_1(1)\}$.
4. Associer $u_1(1)$ à q_0 : $\lambda(q_0) = u_1(1)$.
5. Enregistrer l'état actuel : $Pr := q_0$

Fin initialisation

6. Prendre le vecteur suivant $u_1(t)$ de σ_1 .
7. Si $u_1(t) \in \Omega$ alors chercher l'état q_i associé à $u_1(t)$
Sinon, $\Omega := \Omega \cup \{u_1(t)\}$.

- Créer un nouvel état $q_i \in Q$.
 - Associer $u_1(t)$ à q_i : $\lambda(q_i) = u_1(t)$.
 - $f_{nd}(q_i) = \emptyset$.
8. Si $q_i \notin f_{nd}(Pr)$ alors ajouter q_i à $f_{nd}(Pr)$.
 9. Enregistrer l'état actuel : $Pr := q_i$
 10. Tq $u_1(t)$ n'est pas le dernier élément dans σ_1 , aller à 6.
- Fin.
-

Exemple 4 : Etant donné un système avec un vecteur d'E/S composé de deux signaux d'entrée et un signal de sortie et caractérisé par l'ensemble de séquences d'E/S suivant :

$$\Sigma = \left\{ \sigma_1 = \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right), \left(\begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right), \left(\begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 1 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right), \sigma_2 = \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right), \right. \\ \left. \left(\begin{array}{c} 0 \\ 1 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right), \left(\begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 1 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right), \left(\begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right), \left(\begin{array}{c} 0 \\ 0 \\ 0 \end{array} \right) \right\}.$$

Au début, l'automate est composé d'un seul état initial q_0 comme indiqué dans la première phase de l'algorithme. Cet état n'a aucune relation de transition et possède un ensemble de symboles de sortie vide ($\Omega = \emptyset$). L'analyse de σ_1 , comme indiqué dans la deuxième phase de l'algorithme, montre que le vecteur d'E/S initial de cette séquence et le vecteur final sont égaux. Donc, cette séquence sera utilisée afin de construire l'AANDS initial de ce système. Selon, l'étape 3 et 4 de l'algorithme, le premier vecteur de cette séquence $u_1(1) = [0 \ 0 \ 0]^T$ est associé à l'état initial q_0 comme symbole de sortie et l'ensemble Ω est mis à jour tel que $\Omega := \Omega \cup \{u_1(1)\}$. Ensuite, selon l'étape 6, le vecteur suivant $u_1(2)$ dans σ_1 est pris. On remarque que $u_1(2)$ n'est pas dans Ω , selon l'étape 7 un nouvel état q_1 est créé et associé à $u_1(2)$ comme symbole de sortie. L'ensemble Ω est mis à jour tel que $\Omega := \Omega \cup \{u_1(2)\}$ et selon l'étape 8 la fonction de transition est définie pour q_0 ($f_{nd}(q_0) = q_1$) où q_0 est relié à q_1 . La procédure est poursuivie jusqu'au dernier vecteur dans σ_1 . À la fin, un AANDS initial composé de cinq états q_0, q_1, q_2, q_3 et q_4 est obtenu comme indiqué dans la figure 1.7.

La deuxième étape de cette approche consiste à mettre à jour l'automate initial, établi dans l'étape précédente, en utilisant les séquences d'E/S restantes. Tout d'abord, l'algorithme vérifie, pour chaque séquence σ_i dans Σ , si son vecteur initial est égal à son vecteur final. Si la condition n'est pas vérifiée, la séquence est éliminée de Σ . Sinon, le vecteur qui suit le vecteur initial est considéré et on vérifie s'il appartient à l'ensemble Ω . Si, la relation d'appartenance est satisfaite alors, on cherche l'état auquel ce vecteur est associé et on continue le processus. Dans le cas inverse, ce vecteur est ajouté à Ω , un nouvel état q_j est construit et associé par ce vecteur comme symbole de sortie et la relation de transfert entre ce vecteur et celui qui le précède est définie si

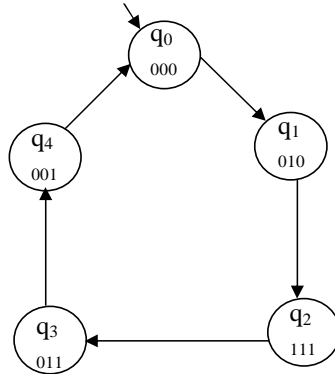


FIGURE 1.7 – Automate initial

elle ne l'était pas. L'opération est refaite avec le reste des vecteurs de cette séquence et les séquences restantes.

Nous reprenons notre exemple, dans cette étape l'automate initial de la figure 1.7 est mis à jour en utilisant la séquence σ_2 . Nous remarquons que cette dernière commence et se termine par le même vecteur que celui de la séquence précédente, donc nous allons l'utiliser pour construire l'automate final. À partir de cette séquence, nous remarquons qu'il n'y a pas de nouvelles valeurs du vecteur d'E/S qui sont enregistrées alors le nombre d'états dans l'automate final reste le même que celui de l'automate initial. Cependant, de nouvelles transitions entre les états seront ajoutées comme le montre la figure 1.8.

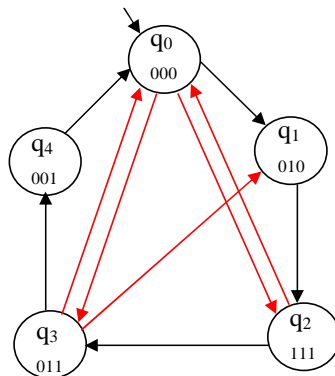


FIGURE 1.8 – L'AANDS final

1.4.2 Approches à base de RdP

Généralement, les approches d'identification à base d'automates à états finis imposent des contraintes soit sur le modèle à identifier ou bien sur les données d'identification. Ces approches nécessitent un temps de calcul élevé et peuvent conduire à une explosion combinatoire. Pour remédier à ces limites, d'autres travaux qui exploitent le formalisme RdP ont été proposés. Par rapport aux automates à états finis, les RdP offrent une modélisation plus puissante des systèmes comportant de la synchronisation ou du parallélisme. De plus, leurs principes mathématiques ainsi que leurs multiples extensions permettent de penser qu'ils peuvent servir pour développer une méthodologie d'identification efficace. Ainsi, leurs propriétés structurelles et comportementales jouent un rôle important dans la définition et la résolution de ce problème par les techniques d'optimisation combinatoire. Dans la suite de cette section, quelques approches proposées afin de résoudre le problème d'identification d'un SED par RdP sont présentées.

1.4.2.1 Approche de Hiraichi

L'approche de Hiraichi est l'une des premières approches développées pour résoudre le problème d'identification d'un SED par RdP. Elle concerne les RdP marqués binaires purs et à choix libre [Hir92]. Dans son approche, l'auteur a exploité les séquences d'événements générés par le système et qui représentent son comportement positif. Notons que, le comportement positif d'un système est représenté par les séquences d'événements qui appartiennent au langage généré par ce système.

Cette approche s'exécute en deux étapes principales. La première consiste à identifier un automate à états finis qui permet de générer les séquences d'événements. Le langage marqué de cet automate est un sous ensemble de langage généré par le RdP à établir. Pour élaborer cet automate, l'algorithme d'identification permet d'abord de construire l'arbre de préfixes pour chaque séquence d'événements dans l'ensemble de séquences d'événements et les regrouper dans un seul arbre de préfixes. Ensuite, cet arbre est représenté sous la forme d'un automate à états finis et le fusionnement des états pouvant être fusionnés est effectué en utilisant les partitions d'états et les invariants de langage. Ces derniers sont des vecteurs associés aux séquences d'événements qui représentent le nombre d'apparitions de chaque symbole dans la séquence concernée selon un ordre bien déterminé.

La deuxième étape de cette approche consiste à trouver un RdP marqué binaire pur et à choix libre équivalent à l'automate établi dans l'étape précédente. En effet, les différents états de cet automate doivent être équivalents aux différents marquages atteignables par le RdP à établir. Ainsi, les états finaux de l'automate doivent être équivalents aux marquages finaux du RdP à élaborer. Notons que, la fonction de transfert

entre un marquage M et une transition t est définie si t est franchissable depuis M . Afin d'établir la structure de ce RdP, l'auteur s'est basé sur l'utilisation des relations de dépendance entre les différents symboles du langage généré par l'automate établi dans la première étape. Elles sont associées aux relations de dépendance entre les transitions du RdP à établir et elles permettent d'insérer les places de ce RdP. Ces relations de dépendance sont les suivantes :

R_{d1} : deux symboles x et y d'un automate ont une relation de dépendance notée R_{d1} si et seulement si $x^* \cap^* y \neq \emptyset$. En terme de RdP marqué : $R_{d1} = \{(x, y) \in T \times T \mid x^* \cap^* y \neq \emptyset\}$.

R_{d2} : deux symboles x et y d'un automate ont une relation de dépendance notée R_{d2} si et seulement si $^*x \cap^* y \neq \emptyset$ et $x \neq y$. En terme de RdP marqué : $R_{d2} = \{(x, y) \in T \times T \mid ^*x \cap^* y \neq \emptyset \wedge x \neq y\}$.

R_{d3} : deux symboles x et y d'un automate ont une relation de dépendance notée R_{d3} si et seulement si $x^* \cap y^* \neq \emptyset$ et $x \neq y$. En terme de RdP marqué : $R_{d3} = \{(x, y) \in T \times T \mid x^* \cap y^* \neq \emptyset \wedge x \neq y\}$.

L'auteur a prouvé qu'en vérifiant ces relations de dépendances, il est possible de déterminer une structure unique d'un RdP marqué binaire pur qui peut générer les séquences d'événements du système considéré. Malheureusement, cette méthode est restreinte à une sous-classe spéciale de RdP.

Exemple 5 : Soit Q un ensemble des séquences d'événements générés par un système à identifier tel que $Q = \{\epsilon, abdecfij, abdcefij, adbce fgh, adebc fgh\}$, ϵ est le mot vide. Selon la première étape de cet algorithme, l'arbre de préfixes final de toutes les séquences d'événements est donné par la figure 1.9.

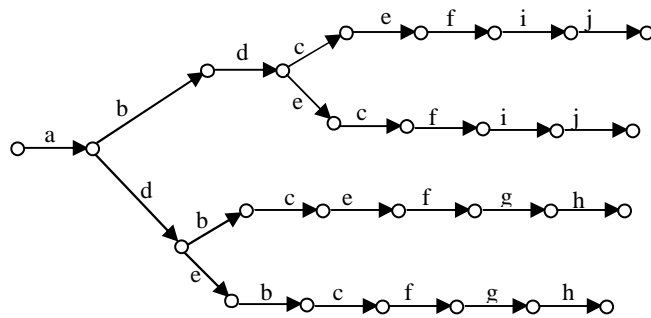


FIGURE 1.9 – L'arbre de préfixes final de toutes les séquences d'événements

Les deux invariants de langage dans cet exemple sont $[1, 1, 1, 1, 1, 1, 0, 0, 1, 1]$ et $[1, 1, 1, 1, 1, 1, 1, 1, 0, 0]$. Ils sont équivalents au nombre d'apparition de chaque symbole dans les séquences d'événements où l'ordre est pris alphabétique. L'automate à

états finis initial qui permet de générer ces séquences d'événements est donné par la figure 1.10.

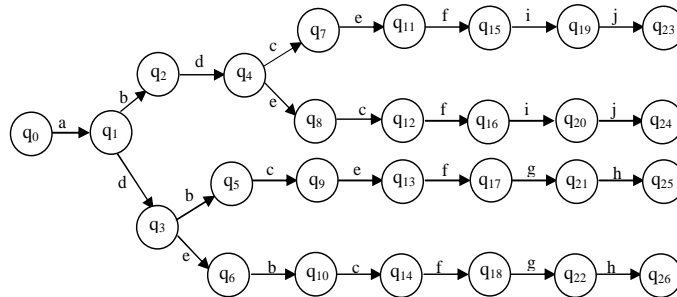


FIGURE 1.10 – L'automate initial qui génère l'ensemble des séquences d'événements Q

Il est composé de 27 états. Dans cet exemple, l'état initial et l'état final sont considérés les mêmes. Donc, les états q_0 , q_{23} , q_{24} , q_{25} , et q_{26} seront fusionnés dans le même état q_0 . L'application du principe de minimisation d'un automate en utilisant les relations d'équivalence entre ses états permet de minimiser progressivement le nombre d'états de cet automate. Par exemple, les états q_{19} et q_{20} sont équivalents, ils sont fusionnés dans un seul état représenté dans l'automate final A_f par l'état q_{11} , pareil pour les états q_{21} et q_{22} qui sont représentés par q_{10} dans A_f . L'automate final obtenu est donné par la figure 1.11.

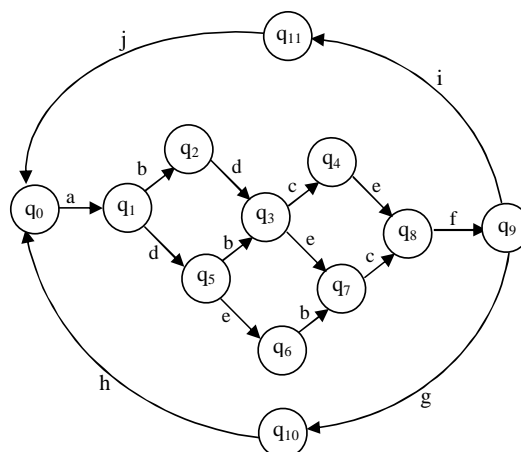


FIGURE 1.11 – L'automate final A_f qui génère l'ensemble des séquences d'événements Q

En appliquant la deuxième étape de l'algorithme, les relations de dépendances entre les symboles de l'automate décrit dans la figure 1.11 sont :

$$\text{dep}(L(A_f)) = \{(h, a), (j, a), (a, b), (a, d), (b, c), (d, c), (d, e), (c, f), (e, f), (f, i), (f, g), (i, j), (g, h)\}.$$

L'utilisation de ces relations permet d'établir le RdP qui peut générer les séquences d'événements considérées précédemment, comme donné sur la figure 1.12.

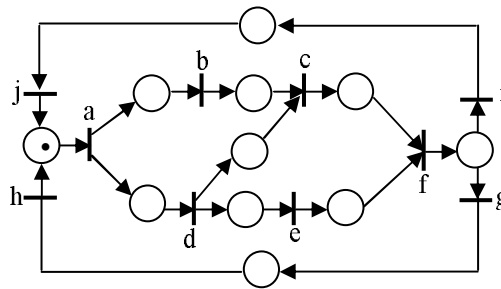


FIGURE 1.12 – Le RdP marqué qui génère les séquences d'événements de l'ensemble Q

1.4.2.2 Approche de T. Bourdeaud'huy et P. Yim

Cette approche fait partie de celles qui ont pour but la synthèse d'un RdP à partir des exigences [BY04]. Dans cette approche, la structure du RdP à identifier est contrainte par un ensemble de spécifications. Ces dernières sont de type structural et comportemental correspondant à l'expression formelle des contraintes à respecter. L'idée principale de cette approche est de définir un RdP partiel dont la structure n'est pas fixée mais composée de variables, i.e., les matrices *Pre* et *Post* sont composées uniquement de variables à valeur dans \mathbb{N} . Puis, les contraintes à respecter sont décrites par des formules logiques portant sur ces variables. Dans cette approche, les auteurs ont formulé deux types de contraintes : structurelles et comportementales. Ainsi, la programmation par contraintes a été utilisée pour trouver les différents RdP qui satisfassent ces contraintes.

Les résultats obtenus montrent que, certains RdP synthétisés peuvent présenter une sorte de "sur-comportement" : toutes les propriétés attendues sont satisfaites, mais certains "mauvais" comportements non spécifiés peuvent aussi apparaître. Ces mauvais comportements sont soit des états interdits ou bien des transitions entre états interdites. Pour inhiber l'atteignabilité de ces comportements, une opération d'affinement est introduite. Tout d'abord, la meilleure solution qui satisfait le maximum des exigences initiales est sélectionnée parmi les solutions qui existent. Puis, de nouvelles places sont ajoutées itérativement au RdP sélectionné. Notons que, cette approche permet de synthétiser plusieurs solutions satisfaisant la plupart des exigences de départ en

utilisant que le comportement positif du RdP à identifier. Cependant, elle permet de donner l'ensemble des solutions possibles sans appliquer aucune technique pour sélectionner la meilleure.

Exemple 6 : Soit le tableau 1.1 qui représente un ensemble de contraintes structurelles.

Exigence	Contrainte correspondante
RdP ordinaire	$\bigwedge_{(i,j) \in [1,m] \times [1,n]} (X_{i,j} \leq 1 \wedge Y_{i,j} \leq 1)$
RdP connexe	$\bigwedge_{i \in [1,m]} (\sum_k X_{i,k} \geq 1 \wedge \sum_k Y_{i,k} \geq 1)$ $\wedge \bigwedge_{j \in [1,n]} (\sum_k X_{k,j} \geq 1 \wedge \sum_k Y_{k,j} \geq 1)$
RdP pur	$\bigwedge_{(i,j) \in [1,m] \times [1,n]} (X_{i,j} + Y_{i,j} < 2)$
Composantes conservatives	$(1, 1, 1, 0, 0, 0)^t \cdot (X_{post} - X_{pre}) = 0$ $(0, 0, 0, 1, 1, 1)^t \cdot (X_{post} - X_{pre}) = 0$

TABLE 1.1 – Table de contraintes à respecter

L'objectif est d'établir un RdP qui vérifie ces contraintes. La première et la troisième contrainte de ce tableau limitent le type de RdP à identifier. La deuxième contrainte exige la connexité de ce RdP. Quant à la dernière contrainte, elle indique les deux T-invariants que possède le RdP à établir. Ce dernier doit vérifier les contraintes comportementales suivantes :

• $M_{p1}[t_1 > \underline{M}_{p2}$, • $M_{p4}[t_4 > \underline{M}_{p5}$, • $M_{p1p4}[t_1 > M_{C_x}$, • $M_{p1p4}[t_4[t_2 > M_{C_x}$
 avec $M_{C_x} = \underline{M}_{p3p6}$, • $M_{p3}[t_3 > ;$ • $\underline{M}_{p6}[t_6 > ,$ • $M_{C_x}[t_3[t_6 > \underline{M}_{p1p4}$. La notation soulignée est utilisée afin d'indiquer que les marquages correspondants sont partiellement connus. Par exemple $M_{p1} = (1, 0, 0, 0, 0, 0)$ et $\underline{M}_{p1p4} = (1, ?, ?, 1, ?, ?)$. Les RdP de la figure 1.13 sont obtenus en appliquant la programmation par contraintes.

Une analyse de ces réseaux permet de constater des problèmes au niveau de l'ordre de franchissement de quelques transitions pour les réseaux numérotés de 2 à 7. Par exemple, rien ne garantit que la transition t_4 doit être franchie après t_2 comme indiqué dans l'ensemble de contraintes. Parmi ces réseaux, le premier a été choisi pour l'affiner parce qu'il satisfait un nombre maximum de contraintes.

Notons que, ce RdP présente un comportement inattendu où la transition t_6 peut être tirée avant la transition t_3 . Afin de l'affiner, la technique de synthèse d'un superviseur a été appliquée pour ajouter une place de contrôle entre ces deux transitions dont le marquage initial égal à 1 comme l'indique la figure 1.14.

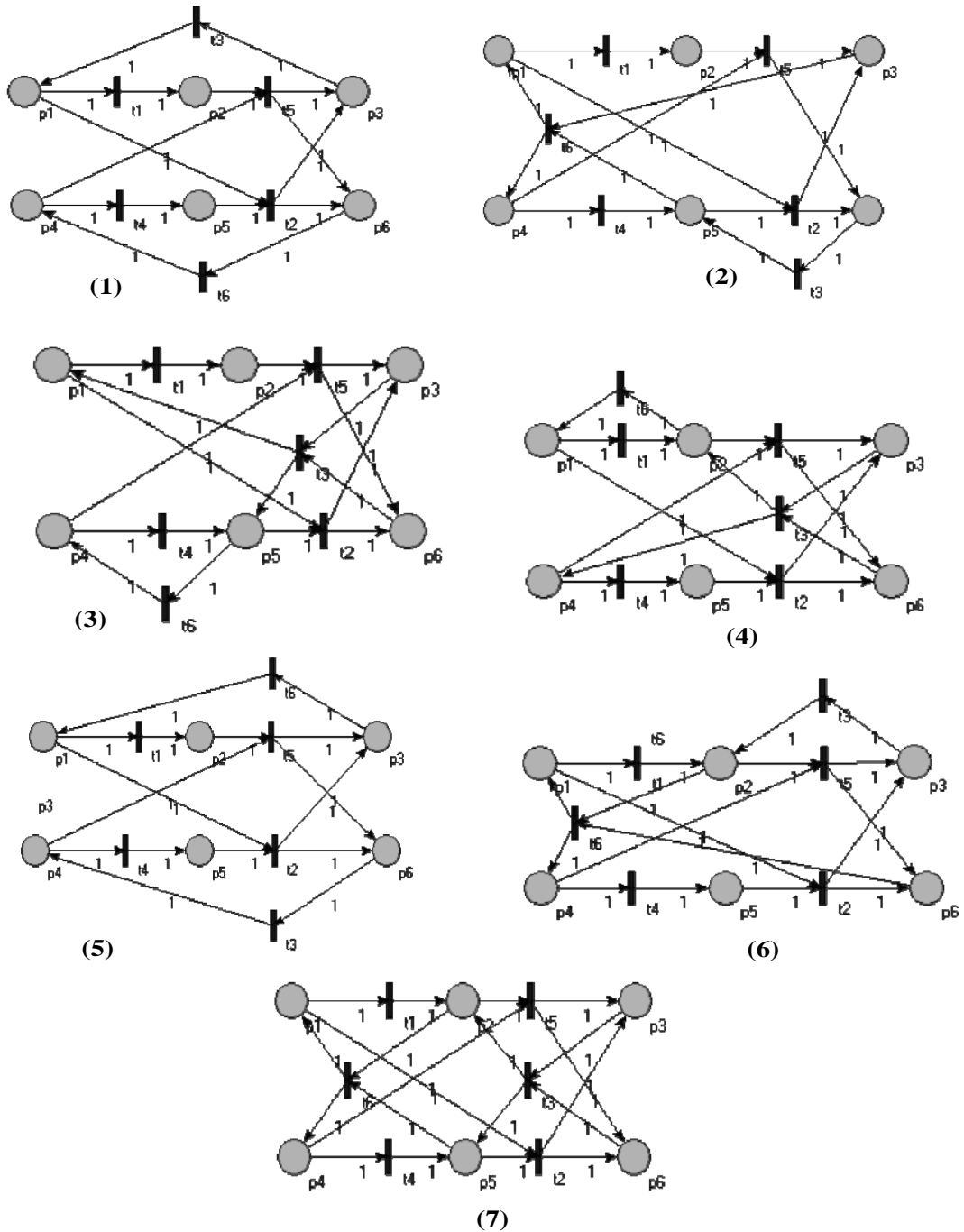


FIGURE 1.13 – Les réseaux de Petri synthétisés

1.4.2.3 Approche de Meda *et al.*

Cette approche fait partie de celles qui exploitent les signaux de sortie du système afin d'établir son modèle comportemental. Elle concerne les SED caractérisés par des

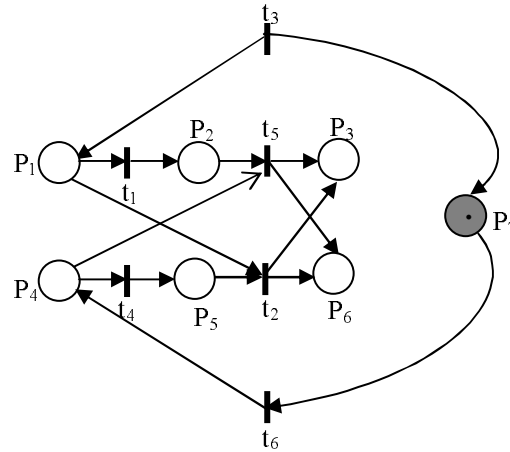


FIGURE 1.14 – Le réseau de Petri affiné

événements détectables et modélisés par au moins un RdP interprété (RdPI) [MTM00]. Le but de l'algorithme d'identification proposé est la validation des comportements du système, raison pour laquelle la description du système à identifier doit être connue. Les auteurs supposent que le modèle RdP à identifier est composé de places mesurables et non mesurables et de transitions contrôlables et non contrôlables. Les places mesurables sont associées aux signaux de sortie mesurables et les transitions contrôlables sont associées aux signaux d'entrée avec $\lambda(t_i) \neq \epsilon$. Les places non mesurables et les transitions non contrôlables sont détectées en utilisant les signaux de sortie. En effet, une transition non contrôlable est associée à un signal d'entrée avec $\lambda(t_i) = \epsilon$. Une place non mesurable est détectée si deux transitions consécutives t_i et t_j ne sont pas connectées par une place mesurable.

Les auteurs ont exploité la propriété de détectabilité d'événements pour déduire des séquences de transitions appelées m_q – mots depuis les séquences de symboles de sortie du système considéré. Puis, ils ont utilisé les relations de dépendance entre les transitions de la même séquence et entre les transitions de séquences différentes pour construire le modèle RdP. Notons que, dans un RdPI, deux transitions t_i et t_j forment une dépendance notée $D = [t_i, t_j]$ si et seulement si il existe une place p_k tel que t_i appartient à l'ensemble de transitions en amont de p_k et t_j à l'ensemble de transitions en aval de p_k . Si p_k est mesurable alors $D = [t_i, t_j]$ est une dépendance mesurable autrement, elle est non mesurable.

L'algorithme d'identification proposé s'exécute durant le fonctionnement du système à identifier. Il est valable avec les RdPI de dépendances non mesurables simples, i.e., chaque place non mesurable forme une seule dépendance non mesurable. A chaque

détection d'un comportement cyclique du système considéré, le $RdPI_{i-1}$ établi à l'étape précédente est mis à jour pour obtenir un nouveau modèle $RdPI_i$ qui acquiert plus d'informations du système que le modèle précédent. Ainsi, de nouvelles dépendances mesurables et non mesurables sont ajoutées et des dépendances non mesurables qui contredisent le nouveau comportement sont éliminées. On peut résumer l'algorithme d'identification par les étapes suivantes :

1. Calculer le modèle initial $RdPI_0$ avec un nombre de places mesurables égal au nombre de places mesurables dans le modèle original (modèle de spécification) et avec un nombre de transitions égal à 0. Puis initialiser q à 0 (q est la dimension du vecteur de sortie).
2. Lire les symboles de sortie jusqu'à la détection d'un $m_q - mot$.
3. Utiliser le $m_q - mot$ détecté pour calculer : la matrice d'incidence actuelle, les dépendances mesurables et non mesurables.
4. Mettre à jour l'ensemble de dépendances et incrémenter q .
5. Construire le modèle $RdPI_q$.
6. Aller à l'étape 2.

Cet algorithme a été adapté pour identifier les systèmes concurrents [MM05] et les systèmes modélisables par les RdPI de dépendances non mesurables complexes (i.e., une seule place non mesurable forme plusieurs dépendances non mesurables simples) [MM01; MM02]. Ainsi, les conditions nécessaires et suffisantes pour avoir un modèle complet en utilisant une seule séquence de transitions ont été étudiées [MM03]. Cependant, cet algorithme exige la connaissance de la description du système à identifier qui doit être décrit par au moins un RdPI. Ceci rend l'algorithme applicable beaucoup plus pour la vérification du modèle.

Exemple 7 [MRM98] : Soit le système de la figure 1.15(a) composé de deux chariots C1 et C2 et quatre capteurs R1, R2, L1 et L2. Le capteur R1 (respectivement L1) détecte le mouvement de C1 vers la droite (la gauche) et R2 (respectivement L2) détecte le mouvement de C2 vers la droite (la gauche). La spécification de fonctionnement de ce système est décrite comme suit. Les deux chariots sont initialement à la position à gauche et ils commencent un mouvement vers la droite suite à l'activation du signal de départ "s". Ainsi, dès que le chariot C1(C2) touche l'interrupteur "c" ("d") il bouge vers la gauche. Une fois que le chariots C1(C2) touche l'interrupteur "a" ("b") il s'arrête et il attend le signal "s" pour recommencer un autre cycle. Le modèle de la spécification de ce système est représenté par la figure 1.15(b).

Les séquences des symboles de sorties observées sont :

$$\delta_1 = \left(\left(\begin{pmatrix} 1 \\ 1 \\ \epsilon \\ \epsilon \end{pmatrix}, \begin{pmatrix} \epsilon \\ 1 \\ 1 \\ \epsilon \end{pmatrix}, \begin{pmatrix} \epsilon \\ 1 \\ \epsilon \\ \epsilon \end{pmatrix}, \begin{pmatrix} \epsilon \\ \epsilon \\ \epsilon \\ 1 \end{pmatrix}, \begin{pmatrix} \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \epsilon \\ \epsilon \end{pmatrix} \right),$$

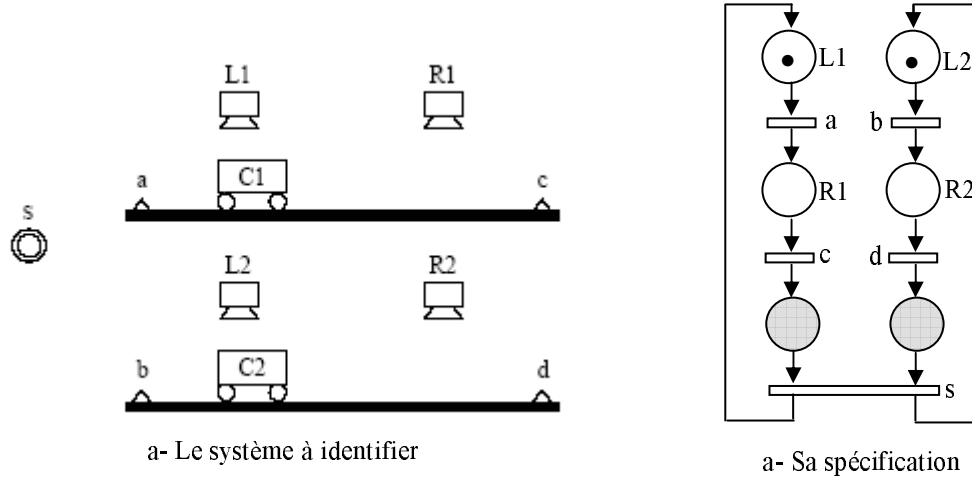
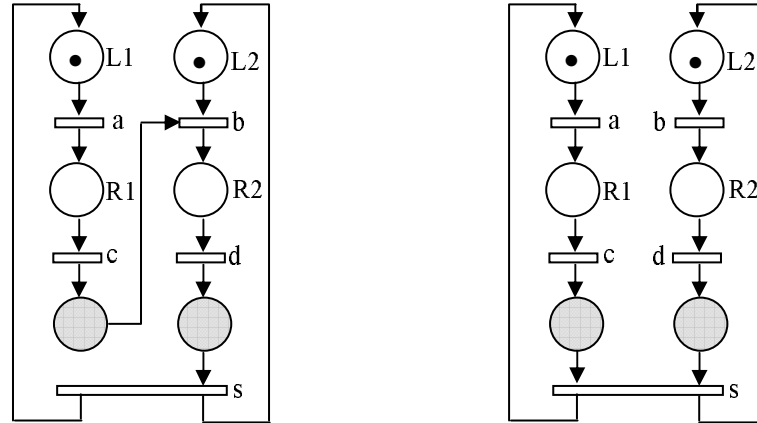


FIGURE 1.15 – Le système à identifier et sa spécification

et $\delta_2 = \left(\left(\begin{pmatrix} 1 \\ 1 \\ \epsilon \\ \epsilon \end{pmatrix}, \begin{pmatrix} 1 \\ \epsilon \\ \epsilon \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ \epsilon \\ \epsilon \\ \epsilon \end{pmatrix}, \begin{pmatrix} \epsilon \\ \epsilon \\ 1 \\ \epsilon \end{pmatrix}, \begin{pmatrix} \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \epsilon \\ \epsilon \end{pmatrix} \right), \text{ avec un vecteur de}$

sortie égal à $\begin{pmatrix} L1 \\ L2 \\ R1 \\ R2 \end{pmatrix}$. Chaque symbole de ces vecteurs de sortie indique le(s) cap-

teur(s) qui a (ont) été activé(s). Ces séquences de symboles de sortie permettent de détecter les séquences de transitions suivantes : $\sigma_1 = (a, c, b, d, s)$ et $\sigma_2 = (b, d, a, c, s)$. En appliquant l'algorithme d'identification, le $RdPI_0$ initial a quatre places et aucune transition. L'analyse de la première séquence de transitions σ_1 permet d'établir les dépendances mesurables $[a, c]$, $[b, d]$, $[s, a]$, $[s, b]$ et les dépendances non mesurables $[c, b]$, $[d, s]$. En effet, deux places non mesurables sont détectées et le RdPI identifié dans cette étape est représenté par la figure 1.16(a). L'analyse de σ_2 permet de détecter les dépendances mesurables : $[b, d]$, $[a, c]$, $[s, b]$, $[s, a]$ et les dépendances non mesurables : $[d, a]$, c, s . La mise à jour du modèle précédent permet d'éliminer la dépendance non mesurable $[c, b]$ parce qu'elle contredit le nouveau comportement et la dépendance non mesurable $[d, a]$ vu que chaque place non mesurable forme une seule dépendance non mesurable. Le modèle RdPI final représenté par la figure 1.16(b), est identique à celui de la spécification et le comportement du système considéré est validé.



a- Modèle identifié après la détection de δ_1 b- Modèle identifié après la détection de δ_1 et δ_2

FIGURE 1.16 – Le modèle identifié

1.4.2.4 Approche de Giua *et al.*

Dans cette approche les auteurs se sont intéressés au problème d'identification d'un RdP en analysant le langage \mathcal{L} qui le génère [GS05]. Ils supposent que ce langage est préfixe-clos et la longueur de la plus grande séquence de transitions de ce langage est connue. Ce langage décrit non seulement les exemples de comportement positif du système considéré (les séquences de transitions qui appartiennent à ce langage) mais aussi les contre exemples (les séquences de transitions qui n'appartiennent pas à ce langage). Dans cette approche, les auteurs considèrent les RdP free labelled où chaque transition est associée à un label unique [GS05]. Ceci implique que l'ensemble de transitions est connu et coïncide avec l'ensemble de labels. Ils supposent aussi que le nombre de places dans le RdP à identifier est connu. Le problème d'identification considéré dans ce cas consiste alors à trouver la structure d'un RdP $N(P, T, Pre, Post)$ et son marquage initial M_0 en donnant : un ensemble de places P de cardinal m , un langage \mathcal{L} et la longueur de la plus grande séquence dans \mathcal{L} tel que le langage généré par le RdP à identifier doit vérifier : $L(N, M_0) = \mathcal{L}$.

Pour résoudre ce problème, une caractérisation algébrique linéaire d'un RdP qui permet de générer un ensemble de séquences de transitions est proposée. Cette caractérisation constitue un ensemble de contraintes qui doivent être respectées par ce RdP. Celles-ci sont formulées en utilisant l'ensemble des exemples positifs et les contre exemples, en exploitant les conditions de franchissement et de désactivation d'une transition depuis une séquence de transitions. Cet ensemble de contraintes est défini comme suit :

$G(\varepsilon, \mathcal{D}) \triangleq$

$$\left\{ \begin{array}{ll} M_0 + Post.\vec{\sigma} - Pre.(\vec{\sigma} + \vec{\varepsilon}_j) \geq \vec{0}, \forall(\sigma, t_j) \in \varepsilon & (a) \\ -KS_{\sigma,j} + M_0 + Post.\vec{\sigma} - Pre.(\vec{\sigma} + \vec{\varepsilon}_j) \leq -\vec{1}_m, \forall(\sigma, t_j) \in \mathcal{D} & (b) \\ \vec{1}^T S_{\sigma,j} \leq m - 1, \forall(\sigma, t_j) \in \mathcal{D} & (c) \\ M_0 \in \mathbb{N}^m & (d) \\ Pre, Post \in \mathbb{N}^{m \times n} & (e) \\ S_{\sigma,j} \in \{0, 1\}^m & (f) \end{array} \right. \quad (1.8)$$

avec,

$$\varepsilon = \{(\sigma, t_j) | \sigma \in \mathcal{L}, |\sigma| < k, \sigma t_j \in \mathcal{L}\} \quad (1.9)$$

est l'ensemble d'exemples positifs

$$\text{et } \mathcal{D} = \{(\sigma, t_j) | \sigma \in \mathcal{L}, |\sigma| < k, \sigma t_j \notin \mathcal{L}\} \quad (1.10)$$

est l'ensemble de contre exemples où K est une constante dont la valeur est très grande. La contrainte (a) représente la condition de franchissement de la transition t_j depuis la séquence σ tel que $\sigma \in \mathcal{L}$. Quant aux contraintes (b) et (c), elles représentent les conditions de désactivation de la transition t_j depuis σ tels que : $\sigma \in \mathcal{L}$ et $\sigma t_j \notin \mathcal{L}$. Pour chaque paire $(\sigma, t_j) \in \mathcal{D}$, on cherche au moins une place qui désactive la transition t_j depuis σ . La contrainte (c) est inspirée de la contrainte (b) en exploitant les marquages des places en amont de la transition t_j et en supposant qu'au moins une place parmi celles-ci peut désactiver cette transition.

Généralement, plusieurs RdP peuvent satisfaire ces contraintes. Pour sélectionner le meilleur, un indice de performance est choisi. Dans ce cas, les auteurs cherchent un RdP avec un nombre minimum de poids sur les arcs et de jetons dans le marquage initial. La fonction objectif est formulée, en utilisant les éléments des matrices Pre et $Post$ et ceux du marquage initial, comme suit :

$$f(M_0, Pre, Post) = \vec{1}_{1 \times m} \cdot (Pre + Post) \cdot \vec{1}_{n \times 1} + \vec{1}_{1 \times m} \cdot M_0.$$

Le problème d'identification, considéré dans ce cas, est résolu en le définissant sous la forme d'un problème de programmation linéaire en nombres entiers :

$$\left\{ \begin{array}{l} \min f(M_0, Pre, Post) \\ \text{s.t. } G(\varepsilon, \mathcal{D}) \end{array} \right. \quad (1.11)$$

Cette approche a été adaptée pour les RdP λ -free labelled où plusieurs transitions peuvent partager le même label [CGS06b][CGS07]. L'étude de la complexité de cette approche [CGS06a] a montré que, le nombre des inconnus augmente au fur et à mesure que le nombre de contre exemples augmente aussi. Quand au temps de calcul, il

est négligeable pour avoir juste des solutions admissibles. Cependant, il devient excessif pour avoir une solution optimale dans le cas de langages décrits par un grand nombre de séquences de transitions de longueur finie et qui peut diminuer l'efficacité de l'approche.

Exemple 8 : Soit $\mathcal{L} = \{\epsilon, t_1, t_1t_1, t_1t_2, t_1t_1t_2, t_1t_2t_1\}$ le langage généré par le RdP à identifier et $m = 2$. La longueur de la plus grande séquence de transition $k = 3$. L'objectif est de déterminer un RdP (N, M_0) avec un nombre minimum d'arcs, un nombre minimum de marques dans le marquage initial et qui vérifie $L_3(N, M_0) = \mathcal{L}$. Ceci revient à résoudre le problème défini dans l'équation (1.11) avec :

$\varepsilon = \{(\epsilon, t_1), (t_1, t_1), (t_1, t_2), (t_1t_2, t_1), (t_1t_1, t_2)\}$ et $\mathcal{D} = \{(\epsilon, t_2), (t_1t_2, t_2), (t_1t_1, t_1)\}$. La solution de ce problème est le RdP introduit dans la figure 1.17 avec $Pre = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $Post = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ et $M_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$.

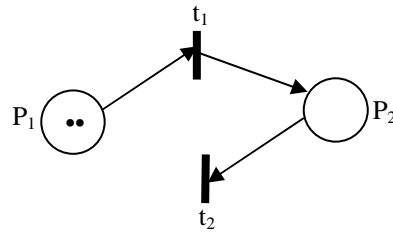


FIGURE 1.17 – Le RdP obtenu

1.4.2.5 Approche de Dotoli *et al.*

Cette approche sert à exploiter le langage généré par un RdP pour l'identifier. Cependant, par rapport à l'approche de Giua, elle consiste à utiliser un seul mot de ce langage [DFM06]. Ce travail concerne les RdP purs et free labelled. Dans cette approche, les auteurs supposent que l'ensemble de places P et l'ensemble de transitions T du RdP à établir sont connus. L'évolution des marquages de ce RdP correspondant à la séquence de transitions $w \in L(N, M_0)$ qui le génère avec $w = t_{\alpha_1}, t_{\alpha_2}, \dots, t_{\alpha_k}$ est décrite par : $M_0[t_{\alpha_1} > M_{\alpha_1}[t_{\alpha_2} > \dots[t_{\alpha_k} > M_{\alpha_k}$. Le problème d'identification consiste à trouver les éléments des matrices Pre et $Post$ et le marquage initial M_0 du RdP qui peut générer le mot w sous une connaissance complète ou partielle de l'évolution de ses marquages.

La résolution de ce problème est faite en deux étapes : dans la première, les auteurs ont étudié la possibilité de trouver un modèle RdP unique qui peut générer w . Par ailleurs, la résolution du problème, dans le cas où l'unicité de la solution n'est pas garantie, est

faite dans la deuxième étape.

Le premier cas : Les conditions nécessaires et suffisantes qui caractérisent w pour obtenir une solution unique sont synthétisées. Ces conditions sont décrites comme suit :
 Condition 1 : La séquence de transitions w doit être complète et autorise le franchissement de toutes les transitions de l'ensemble T au moins une fois.

Condition 2 : Un sous ensemble de transitions T^* doit être sélectionné à partir de l'ensemble de transitions T tels que les marquages en amont et en aval de chaque transition dans ce sous-ensemble sont complètement mesurables et le marquage M_{α_1} est aussi mesurable. Ainsi, pour chaque transition t_{α_i} dans l'ensemble T^* tel que $t_{\alpha_i} \neq t_{\alpha_1}$ et pour chaque transition t_j dans l'ensemble T une seule transition est choisie telle que $t_{\alpha_i} = t_j$.

Une fois que ces deux conditions sont vérifiées, les vecteurs d'incidence des transitions de l'ensemble T^* , les éléments des matrices Pre et $Post$ et ceux du marquage initial M_0 peuvent être calculés. En effet, le fait que le RdP à identifier est pur alors les équations suivantes sont vérifiées : $\forall p \in P, \forall t_{\alpha_i} \in T^*$:

$$\text{Si } W(p, t_{\alpha_i}) \geq 0 \Rightarrow Post(p, t_{\alpha_i}) = W(p, t_{\alpha_i}) \text{ et } Pre(p, t_{\alpha_i}) = 0, \quad (1.12)$$

$$\text{Si } W(p, t_{\alpha_i}) < 0 \Rightarrow Post(p, t_{\alpha_i}) = 0 \text{ et } Pre(p, t_{\alpha_i}) = -W(p, t_{\alpha_i}). \quad (1.13)$$

Exemple 9 : Soit $w = t_{\alpha_1}, t_{\alpha_2}, t_{\alpha_3}, t_{\alpha_4}, t_{\alpha_5}, t_{\alpha_6}, t_{\alpha_7} = t_1, t_2, t_1, t_3, t_4, t_3, t_4, t_1, t_1$ une séquence de transitions générée par un RdP (N, M_0) à identifier. Ce dernier est supposé composé de 4 places et de 4 transitions. À partir de la séquence w qui satisfait les conditions (1) et (2), le sous-ensemble de transitions T^* suivant est sélectionné avec $T^* = \{t_{\alpha_2}, t_{\alpha_3}, t_{\alpha_4}, t_{\alpha_5}\} = \{t_2, t_1, t_3, t_4\}$. Les marquages atteignables par ce RdP équivalent à T^* sont : $M_{\alpha_1} = [1 \ 2 \ 0 \ 0]^T$, $M_{\alpha_2} = [1 \ 1 \ 1 \ 0]^T$, $M_{\alpha_3} = [0 \ 3 \ 1 \ 0]^T$, $M_{\alpha_4} = [0 \ 1 \ 1 \ 1]^T$, $M_{\alpha_5} = [2 \ 1 \ 0 \ 0]^T$. Les vecteurs d'incidence de chaque transition dans l'ensemble T^* sont obtenus en appliquant l'équation de marquage du RdP : $W(., t_1) = M_{\alpha_3} - M_{\alpha_2}$, $W(., t_2) = M_{\alpha_2} - M_{\alpha_1}$, $W(., t_3) = M_{\alpha_4} - M_{\alpha_3}$, $W(., t_4) = M_{\alpha_5} - M_{\alpha_4}$. Puis en appliquant les équations (1.12) et (1.13), les éléments des matrices Pre et $Post$ sont :

$$Pre = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ et } Post = \begin{bmatrix} 0 & 0 & 0 & 2 \\ 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Finalement, à partir de l'équation $W(., t_1) = M_{\alpha_3} - M_{\alpha_2}$ la valeur du marquage initial M_0 est calculée et est égal à $[2 \ 0 \ 0 \ 0]^T$. Le RdP obtenu est représenté par la figure 1.18.

Le deuxième cas : Dans cette étape, le problème du non satisfaction des conditions 1 et/ou 2 est pris en considération. Afin d'y remédier à cela, ils l'ont décrit comme un problème d'optimisation en définissant un ensemble de contraintes et une fonction objectif à satisfaire par le RdP à identifier. Ces contraintes sont formulées en utilisant

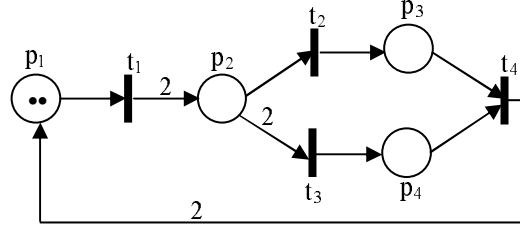


FIGURE 1.18 – Le modèle RdP identifié unique

les transitions de la séquence w et les marquages mesurables correspondant. Quant à la fonction objective, elle est choisie pour satisfaire des propriétés particulières. Par exemple, pour avoir un RdP avec un nombre minimum d'arcs et de marques dans le marquage initial, la fonction objectif prend la forme suivante :

$$f(Pre, Post, M_0) = \vec{1}_{1 \times m} \cdot (Pre + Post) \cdot \vec{1}_{n \times 1} + \vec{1}_{1 \times m} \cdot M_0. \quad (1.14)$$

Ensuite, le RdP qui minimise la fonction objectif est obtenu en résolvant le problème de la programmation linéaire en nombres entiers (PLNE) suivant :

$$\min f(Pre, Post, M_0) \quad (1.15)$$

sous les contraintes $E(P, T) \triangleq$

$$\left\{ \begin{array}{l} Pre, Post \in \mathbb{N}^{m \times n} \\ M_0 \in \mathbb{N}^m \\ Post^T \cdot \vec{1}_{m \times 1} + Pre^T \cdot \vec{1}_{m \times 1} \geq \vec{1}_{n \times 1} \\ Post \cdot \vec{1}_{n \times 1} + Pre \cdot \vec{1}_{n \times 1} \geq \vec{1}_{m \times 1} \\ M_0 - Pre(\cdot, t_{\alpha_1}) \geq \vec{0}_{m \times 1} \\ M_0 + Post(\cdot, t_{\alpha_1}) - Pre(\cdot, t_{\alpha_1}) = M_{\alpha_1} \\ \forall t_{\alpha_i} \in w \text{ avec } i \neq 1, Pre(\cdot, t_{\alpha_i}) \leq M_{\alpha_{i-1}} \\ \forall t_{\alpha_i} \in w \text{ avec } i \neq 1, Post(\cdot, t_{\alpha_i}) - Pre(\cdot, t_{\alpha_i}) = M_{\alpha_i} - M_{\alpha_{i-1}} \end{array} \right. \quad (1.16)$$

Ces contraintes caractérisent : la classe de RdP à établir, sa connexité et la condition de franchissement d'une transition t_{α_i} depuis le marquage initial ou bien un marquage quelconque.

Cette approche a été adaptée afin d'identifier en ligne un SED modélisable par RdP λ -free labelled avec et sans transitions non mesurables [DFM07], [DFM08a] [DFM⁺08b]. Le modèle RdP obtenu permet de produire les comportements originaux mais, rien ne peut garantir qu'il ne génère pas des "sur-comportements".

Exemple 10 : Nous considérons l'exemple précédent où nous cherchons le RdP qui peut générer la séquence de transitions $w = t_{\alpha_1}, t_{\alpha_2}, t_{\alpha_3}, t_{\alpha_4} = t_1, t_2, t_1, t_3$. Les marquages atteignables correspondant à w sont : $M_{\alpha_1} = [1 \ 2 \ 0 \ 0]^T$, $M_{\alpha_2} = [1 \ 1 \ 1 \ 0]^T$, $M_{\alpha_3} = [0 \ 3 \ 1 \ 0]^T$, $M_{\alpha_4} = [0 \ 1 \ 1 \ 1]^T$. Le fait que la condition 1 du premier cas n'est pas satisfaite implique que la résolution du problème de PLNE défini par les équations (1.14) et (1.16) permet d'obtenir le modèle RdP de la figure 1.19. Cependant, si la sé-

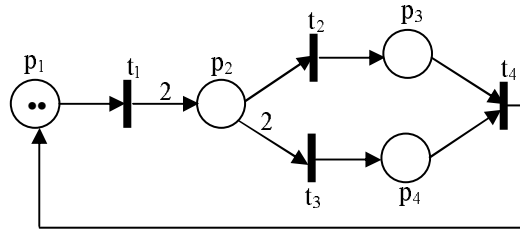


FIGURE 1.19 – Le modèle RdP identifié 1

quence de transitions observée $w = t_{\alpha_1}, t_{\alpha_2}, t_{\alpha_3}, t_{\alpha_4} = t_1, t_2, t_1, t_3$ n'est pas complète et que les marquages $M_{\alpha_1} = [1 \ 2 \ 0 \ 0]^T$ et $M_{\alpha_4} = [0 \ 1 \ 1 \ 1]^T$ sont mesurables et les marquages M_{α_2} et M_{α_3} ne le sont pas alors, la résolution du problème de PLNE défini précédemment permet d'établir le modèle RdP de la figure 1.20.

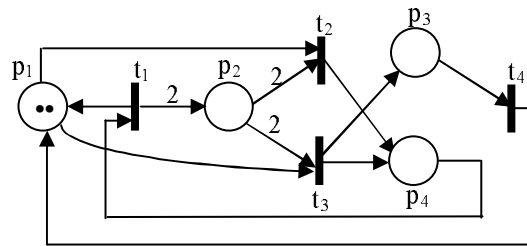


FIGURE 1.20 – Le modèle RdP identifié 2

1.5 Conclusion

Dans ce chapitre, nous avons présenté le problème d'identification d'un SED dont l'objectif est l'élaboration d'un modèle comportemental du système en analysant ses signaux d'entrée et/ou de sortie. Une classification des méthodes d'identification selon des différents critères a été proposée. Une synthèse générale sur les approches d'identification d'un SED par les automates à états finis et par RdP a été présentée. Cependant,

les modèles SED dans le cas de systèmes réels conduisent souvent à des modèles d'une taille excessive du fait de l'explosion combinatoire du nombre d'états. L'inconvénient majeur des approches à base d'automates est le temps de calcul élevé spécialement pour des systèmes ayant un nombre d'entrée et de sortie considérable. Notons également que ces approches ne décrivent pas de façon explicite l'exécution concurrente et distribuée contrairement aux RdP, qui sont plus adaptés à la prise en compte de ces phénomènes. Par ailleurs, les RdP ont, par rapport aux automates, l'avantage d'être un modèle bénéficiant d'une structure plus riche, s'adaptant parfaitement à la description des SED. Ils se distinguent par leur puissance de modélisation et leurs nombreuses propriétés structurelles et comportementales. Ces propriétés jouent un rôle important dans la définition des contraintes qui doivent être satisfaites par le RdP à identifier. La formulation de ces propriétés sous la forme mathématique facilite la définition du problème d'identification d'un SED par RdP sous la forme d'un problème d'optimisation discrète. Ceci donne la possibilité d'utiliser les techniques d'optimisation pour le résoudre. Ces dernières permettent de donner les différentes solutions faisables et de choisir l'optimale en fonction de l'objectif. Néanmoins, établir un modèle comportemental d'un SED sous la forme d'un RdP n'est pas un processus facile, plus particulièrement lorsqu'on ne possède pas des informations *a priori* sur le système à identifier et/ou sur le modèle identifié. Notons que, le problème d'identification d'un SED par RdP fait l'objet d'études récentes et les travaux relatifs à sa résolution sont rares.

L'analyse des approches à base de RdP proposées pour résoudre ce problème montre que, soient elles utilisent les symboles de sortie du système comme l'approche de Meda *et al.* [MTM00], [MM01], [MM02] [MM05]. Soient elles utilisent le langage généré par le SED à identifier [Hir92], [GS05], [CGS06b], [CGS07] ou un mot de ce langage [DFM06]. Ainsi, elles utilisent les événements du système et les états correspondants [DFM07; DFM08a].

Néanmoins, ces approches ont des hypothèses restrictives soit sur les données d'identification ou bien sur la structure de modèle à identifier. Prenant l'approche de Hiraichi par exemple, elle consiste à établir un modèle RdP en utilisant un ensemble de ses séquences d'événements. Ces dernières représentent le comportement positif du RdP à identifier. L'inconvénient majeur de cette méthode est que, l'ordre d'occurrence de ces événements doit être respecté. Aussi, elle est restreinte à une sous classe de RdP qui sont les RdP marqués purs et à choix libre.

L'hypothèse sur l'ordre des événements est présente aussi dans l'approche de Meda *et al.* De plus, dans cette approche, la description du système à identifier doit être connue et utilisée durant le processus d'identification. Ainsi, le système considéré doit être modélisé par au moins un RdP interprété. Ceci rend la méthode d'identification valable pour la validation du modèle.

Dans l'approche proposée par T. Bourdeaud'huy et P. Yim, la programmation par contraintes a été utilisée pour synthétiser les différentes solutions satisfaisant la plupart des contraintes de départ. Ces contraintes sont formulées en utilisant que les comportements positifs du RdP à identifier. Malheureusement, aucune technique n'est appliquée pour sélectionner la meilleure solution.

Les approches de Giua *et al.* et de Dotoli *et al.* proposent chacune une solution pour le problème d'identification d'un RdP en utilisant respectivement le langage généré par le RdP à établir ou une séquence de ce langage. Cependant, dans les deux approches des connaissances *a priori* concernant la structure de RdP à identifier sont indispensables pour appliquer la procédure d'identification. L'utilisation des contre exemples dans l'approche de Giua *et al.* est l'inconvénient majeur de cette méthode parce qu'il augmente sa complexité et nécessite un temps de calcul excessif pour avoir une solution optimale. De plus, l'analyse du modèle identifié issu de l'approche de Dotoli *et al.*, montre qu'il permet de générer les comportements du système considéré mais, rien ne garantit que ce modèle ne génère pas des sur-comportements.

Pour remédier à ces inconvénients, nous proposons dans ce travail deux approches d'identification d'une classe de SED par RdP. L'idée principale de ces approches est de définir le problème d'identification considéré comme un problème d'optimisation combinatoire. La première approche permet d'établir un modèle RdP d'un SED en utilisant que ses signaux de sortie. Quand à la seconde approche, elle consiste à identifier un SED en analysant ses signaux d'entrée et de sortie mesurables. Ces méthodes d'identification sont basées sur l'utilisation de la technique de la programmation linéaire binaire (PLB). Cette technique permet de résoudre ce problème en donnant toutes les solutions réalisables et en sélectionnant la solution optimale.

Chapitre 2

Identification d'une classe des SED par réseaux de Petri

Résumé : Dans ce chapitre, deux méthodes permettant d'établir un modèle comportemental d'un SED sous la forme d'un RdP sont proposées. Le SED à identifier est supposé modélisable par un RdP ordinaire pur, ce qui est le cas de plusieurs systèmes réels. L'une de ces méthodes est basée sur l'utilisation des signaux de sortie mesurables. Dans ce cadre, nous commençons par la détermination des conditions, nécessaires et suffisantes, permettant d'avoir une solution unique. Ensuite, nous traitons les cas où la solution n'est pas unique en définissant le problème d'identification sous la forme d'un problème d'optimisation discrète et en utilisant la technique de la Programmation Linéaire Binaire (PLB) pour le résoudre. La deuxième méthode présentée dans ce chapitre est basée sur l'analyse des signaux d'entrée et de sortie mesurables et consiste à calculer la structure du RdP (matrices *Pre* et *Post*) en déterminant la partie mesurable et en estimant la partie non mesurable ainsi que le marquage initial. Les résultats numériques obtenus illustrent les performances de ces deux approches en termes de temps de calcul et de la nature de la solution. Les résultats de ce chapitre ont été publiés dans : [BME⁺06a], [BME⁺06b], [BME⁺07].

2.1 Introduction

L'analyse des travaux proposés pour résoudre le problème d'identification d'un SED par RdP permet de les classer en trois catégories. La première est basée sur l'analyse des signaux de sortie du système (approches à base d'état) [MTM00] [MM05], la deuxième exploite les séquences d'événements générées par le système à identifier (approches à base d'événements) [GS05] [Hir92] et la dernière utilise une séquence d'événements et la séquence des vecteurs d'état équivalente (approches à base d'états et d'événements)[DFM06].

Cependant, ces approches ne peuvent pas être exploitées dans notre contexte de travail pour différentes raisons. En effet, les approches de Hiraichi [Hir92] et de Meda *et al.*[MTM00], [MM05] imposent un ordre particulier sur l'occurrence des événements (cf. §1.4.2.1 et §1.4.2.3). Or, cette condition est trop restrictive dans notre cas car nous considérons particulièrement les systèmes en boucle fermée caractérisés par des événements générés par le contrôleur. D'autre part, les approches de Giua *et al.* [GS05] et de Dotoli *et al.* [DFM06] proposent chacune une solution pour le problème d'identification en le définissant sous la forme d'un problème d'optimisation. Cependant, ces deux approches nécessitent des connaissances *a priori* concernant la structure du RdP à identifier. De plus, Giua *et al.* utilisent des contre exemples qui sont impossibles à générer dans notre cas car ces derniers représentent des comportements anormaux du système. Enfin, l'analyse de l'approche de Dotoli *et al.* montre que le modèle établi permet certes de produire les comportements observés mais peut aussi générer des "sur-comportements".

Pour pallier ces inconvénients, deux méthodes systématiques d'identification d'une classe de SED par RdP sont proposées dans ce chapitre. Ces méthodes s'apparentent à celles qui définissent le problème d'identification sous la forme d'un problème d'optimisation équivalent. La première méthode est basée sur l'utilisation des signaux de sortie mesurables. Quant à la deuxième, elle utilise des signaux d'entrée et de sortie mesurables. Ces méthodes se basent sur l'exploitation d'une technique d'optimisation discrète, qui est la Programmation Linéaire Binaire (PLB) et sont valables pour toute classe de SED modélisable par des RdP ordinaires purs.

Dans ce qui suit, nous présentons un aperçu général sur l'optimisation discrète qui sera utilisée ultérieurement puis nous définissons la classe de SED à identifier. Ensuite, nous exposons les deux approches d'identification proposées.

2.2 Introduction à l'optimisation discrète

L'optimisation consiste à minimiser ou maximiser une fonction définie sur un espace de dimension finie. Résoudre un tel problème, c'est rechercher, parmi un ensemble de solutions qui vérifient des contraintes données, la ou les solutions qui rendent minimale (ou maximale) la fonction mesurant la qualité de cette solution qui est *la fonction objectif*.

Selon la nature des variables utilisées (continues ou discrètes), il est possible de distinguer deux branches différentes de l'optimisation. En effet, si l'on minimise (maximise) une fonction $f(x)$ avec $x \in \mathbb{R}^n$, il s'agit de l'*optimisation en variables continues*, tandis que si $x \in \mathbb{Z}^n$ on a affaire à l'*optimisation combinatoire ou en variables discrètes* [All05]. Un problème d'optimisation combinatoire (discrète) est défini comme suit :

Définition 1 : Un problème d'optimisation combinatoire consiste à chercher le minimum s^* d'une fonction f , le plus souvent à valeurs entières ou réelles, sur un ensemble fini S :

$$f(s^*) = \min_{s \in S} f(s).$$

où f étant la fonction objectif [LPS03].

Notons que, la définition précédente concerne la minimisation, et il suffit de remarquer que maximiser f équivaut à minimiser $-f$.

Les techniques de résolution des problèmes d'optimisation dépendent de la nature de la fonction objectif et de l'ensemble des contraintes. Parmi les techniques d'optimisation combinatoire on peut trouver : la programmation quadratique, la programmation dynamique, la programmation linéaire et la programmation linéaire en nombres entiers [LPS03].

Dans cette thèse nous nous intéressons aux problèmes de la Programmation Linéaire (PL) qui sont des problèmes d'optimisation où la fonction objectif et les contraintes sont toutes linéaires. Un problème de programmation linéaire est défini par sa forme standard :

$$\begin{aligned} \min f &= c^T x, x \in \mathbb{R}^n \\ \text{s.c } Ax &\leq b \\ x &\geq 0 \end{aligned} \tag{2.1}$$

où $b, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ (m est le nombre de contraintes et n est le nombre de variables) et la contrainte $x \geq 0$ signifie que toutes les composantes de x sont positives ou nulles [All05].

Si les variables ne peuvent prendre que des valeurs entières alors on parle de la Programmation Linéaire en Nombres Entiers (PLNE). Un problème de PLNE est un problème de PL dont le domaine de faisabilité est un ensemble discret de points. Il peut être représenté mathématiquement comme suit :

$$\begin{aligned} \min f &= c^T x \\ \text{s.c} : Ax &\leq b \\ x &\geq 0, \quad x_j \text{ entier } (j = 1 \dots n) \end{aligned} \quad (2.2)$$

avec $A \in \mathbb{Z}^{m \times n}$ est une matrice de dimension $m \times n$ dont les coefficients a_{ij} sont entiers, c est un vecteur de dimension n et ses coefficients c_j sont entiers ($c \in \mathbb{Z}^n$), b est un vecteur de dimension m et ses coefficients b_i sont entiers aussi ($b \in \mathbb{Z}^m$) [All05].

Si les variables n'ont que les deux valeurs possibles 0 et 1, alors on parle de la PL en variables 0-1 ou bien la Programmation Linéaire Binaire (PLB). Notons que cette dernière technique sera utilisée dans la suite de cette thèse.

Exemple 1 : Dans le but d'illustrer le principe de base de la PLB, considérant l'exemple du problème linéaire en variables 0-1 décrit par le système linéaire suivant [All05] :

$$\left\{ \begin{array}{l} \max f = 9x_1 + 5x_2 + 6x_3 + 4x_4 \\ \text{s.c,} \quad 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \quad (1) \\ \quad \quad \quad x_3 + x_4 \leq 1 \quad (2) \\ \quad \quad \quad -x_1 + x_3 \leq 0 \quad (3) \\ \quad \quad \quad -x_2 + x_4 \leq 0 \quad (4) \\ \quad \quad \quad x_j \leq 1 \forall j \in \{1, \dots, 4\} \\ \quad \quad \quad x_j \geq 0 \forall j \in \{1, \dots, 4\} \\ \quad \quad \quad x_j \text{ entier } \forall j \in \{1, \dots, 4\} \end{array} \right.$$

L'énumération des différentes valeurs du couple de variables (x_1, x_2) permet de déterminer la meilleure solution de ce problème :

1. $x_1 = x_2 = 0$: (3) et (4) $\Leftrightarrow x_3 = x_4 = 0, f = 0$;
2. $x_1 = x_2 = 1$: (1) $\Leftrightarrow x_3 = x_4 = 0, f = 14$;
3. $x_1 = 1, x_2 = 0$: (1) $\Leftrightarrow x_3 = 0, (4) \Leftrightarrow x_4 = 0, f = 9$;

4. $x_1 = 0, x_2 = 1 : (3) \Leftrightarrow x_3 = 0$; on peut poser $x_4 = 1, f = 9$.

Donc, la solution optimale est $f = 14$.

Ce type de problèmes peut être résolu par des méthodes propres au PLNE comme la méthode de recherche arborescente par séparation et évaluation. Cette dernière, appelée *branch-and-bound* en anglais, consiste à effectuer une énumération implicite de l'espace des solutions par un arbre de recherche. La méthode de séparation et évaluation se base sur deux principes qui sont : le principe de séparation et le principe d'évaluation.

Principe de séparation :

La phase de séparation consiste à diviser le problème en un certain nombre de sous-problèmes qui ont chacun leur ensemble de solutions réalisables de telle sorte que l'union des sous-ensembles constitue l'ensemble initial S . Ainsi, en résolvant tous les sous-problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Ce principe de séparation peut être appliqué de manière récursive à chacun des sous-ensembles de problèmes obtenus. Notons que, pour les problèmes de PL en variables 0-1 la séparation se fait en donnant à chaque itération les valeurs 0 ou 1 à une variable x_i .

Principe d'évaluation :

L'évaluation d'un noeud de l'arbre de recherche a pour but de déterminer l'optimum de l'ensemble des solutions réalisables associé au noeud en question, ou de prouver mathématiquement que cet ensemble ne contient pas de solutions intéressantes pour la résolution du problème. Cette procédure consiste à calculer des bornes inférieures ou supérieures sur le sous-problème dans un noeud. Si la valeur du majorant, dans le cas d'un problème de minimisation, est plus grande que la solution réalisable trouvée jusqu'à présent alors le sous-ensemble ne contient pas d'optimum.

L'algorithme par séparation et évaluation pour une minimisation s'exécute comme indiqué dans l'algorithme 1.

Algorithme 1 : L'algorithme de séparation et évaluation

Données :

S : l'espace de recherche;

L = {S} : liste de sommets qui représentent les sous problèmes ;

//Fonctions;

Séparer() : la fonction qui fait l'opération de séparation;

Evaluer() : la fonction qui fait l'évaluation de la solution;

Suivant() : la fonction qui permet d'accéder aux sommets de l'arbre;

//initialisation;

U = 0 : la borne initiale ;

$S_{best} = -\infty$: la meilleure solution;

tant que L $\neq \emptyset$ **faire**

 S := Suivant(L) ; $S_1, \dots, S_n =$ Séparer(S) ;

pour chaque S_i **faire**

si Evaluer(S_i) > U ou S_i non réalisable **alors**

 └ Eliminer S_i

sinon

si S_i est réalisable **alors**

 └ $S_{best} = S_i$;

 └ U = Evaluer(S_i);

 └ Eliminer S_i

sinon

 └ L = L U S_i

Exemple 2 : Dans le but d'illustrer le principe de base de l'algorithme de séparation et évaluation, considérant l'exemple du problème linéaire en variables 0-1 décrit par (2.3) [All05] :

$$\begin{cases} \max f = 8x_1 + 5x_2 \\ \text{s.c,} & x_1 + x_2 \leq 6 \\ & 9x_1 + 5x_2 \leq 45 \\ & x_j \in \{0, 1\} \forall j \in \{1, 2\} \end{cases} \quad (2.3)$$

Le schéma de séparation et évaluation de ce système est représenté dans la figure 2.1. Le fait que les variables de ce système sont binaires, la procédure de séparation dans chaque noeud se fait en donnant à chaque fois à la variable concernée par la séparation les valeurs 0 ou 1. L'évaluation s'effectue à chaque noeud en calculant la valeur de la fonction objectif pour les variables fixées plus un majorant sur la fonction objectif pour les variables non fixées. En effet, à partir du sommet initial et pour $x_1 = 0$,

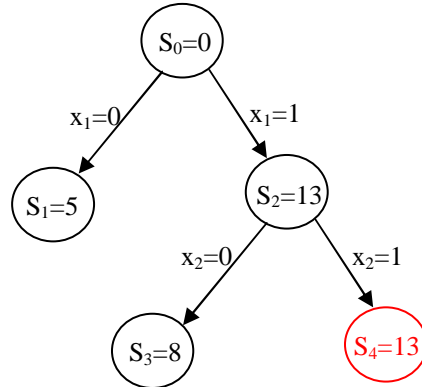


FIGURE 2.1 – Schéma de séparation et évaluation du système de l'exemple 2

le coût est égal à 5 et pour $x_1 = 1$ le cout est égal à 13. Le fait qu'on est dans un cas de maximisation, donc on arrête la séparation pour le sommet S1 parce que quelque soit la valeur qu'on va obtenir elle sera strictement inférieure à la borne supérieure. En poursuivant, le processus la solution optimale dans ce cas est $S_4 = 13$.

Dans nos méthodes, le logiciel CPLEX de ILOG est utilisé comme solveur des programmes linéaires. C'est un logiciel de programmation mathématique permettant de résoudre des programmes linéaires avec des variables réelles et/ou entières. Ce logiciel utilise plusieurs types d'algorithmes de résolution disponibles comme le simplexe et l'algorithme de séparation et évaluation dans le cas des problèmes de la programmation linéaire en nombres entiers.

2.3 Classe de systèmes à identifier

Dans cette thèse, nous nous intéressons à l'identification des SED modélisables par RdP ordinaires purs et caractérisés par des événements détectables. Le système à identifier est composé de deux parties ; opérative et commande ; fonctionnant en boucle fermée. La partie opérative (PO) est composée de capteurs et d'actionneurs. Elle reçoit les ordres envoyés aux actionneurs par la partie commande (PC) et envoie les informations à la partie commande via les capteurs. Les signaux d'actionneurs et de capteurs sont considérés respectivement comme des signaux d'entrée et de sortie du système, comme le montre la figure 2.2. Notons que, à notre connaissance, il n'existe aucun travail qui porte sur l'identification de ce type de systèmes en utilisant les RdP et que les travaux existants sont à base d'automates à états finis [Kle05], [KLL05].

L'idée principale de nos approches est de représenter les signaux d'entrée générés

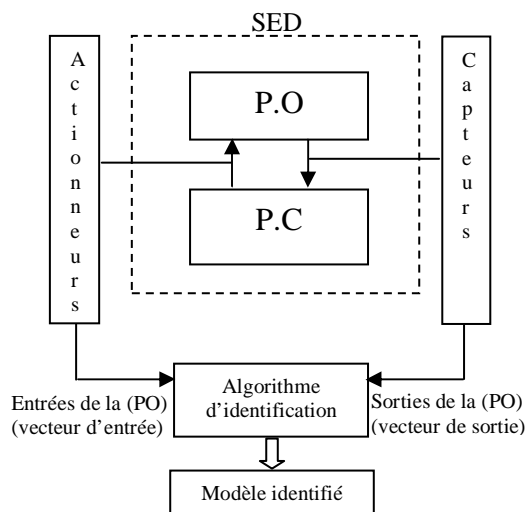


FIGURE 2.2 – L'architecture du système à identifier

par les actionneurs, équipant le système, par un ensemble d'événements se produisant durant le fonctionnement de ce système. Ces événements seront associés aux transitions du RdP à identifier. Aussi, partant du fait que plusieurs actionneurs peuvent être activés et/ou désactivés simultanément dans un système réel, nous proposons de représenter ce changement simultané par une seule transition. Ce choix est motivé par, le fait que, les RdP ne permettent pas le franchissement de plus qu'une seule transition à la fois.

De la même façon, l'ensemble des signaux de sortie mesurés par les capteurs équipant le système est représenté par un vecteur de sortie qui sera considéré dans la suite comme étant la partie mesurable du vecteur d'état du système. Cette partie sera représentée par un ensemble de places mesurables dans le RdP à établir. Quand à la partie non mesurable du vecteur d'état, elle sera représentée par un ensemble de places non mesurables. La détermination de ce dernier sera détaillée dans la suite et sa présence se justifiée par l'impossibilité d'équiper complètement le système par des capteurs.

Enfin, une fois les signaux d'entrée et de sortie du système sont enregistrés, ils seront exploités pour construire un modèle représentant les comportements observés. Bien évidemment l'obtention d'un modèle correct et complet nécessite l'observation de tous les comportements positifs du système. Ainsi, nous considérons qu'un nombre suffisant de cycles de fonctionnement du système est observé durant l'horizon d'enregistrement des données d'entrée/sortie. Cette hypothèse est, en effet, nécessaire pour obtenir un modèle correct et complet du système.

Exemple 3 : Système à deux réservoirs

Dans le but d'illustrer la façon dont on enregistre les signaux d'entrée et de sortie, considérons l'exemple du système présenté par la figure 2.3. Ce dernier est un benchmark introduit dans [Lef05] et il est constitué de deux réservoirs R1 et R2 reliés entre eux par deux conduites A et B comme le montre la figure 2.3. La conduite A est située au fond des réservoirs (référence des niveaux d'eau), la conduite B est située à une hauteur de 50 cm. Le système est composé aussi de deux actionneurs commandés en tout-ou-rien : une vanne V1 et une pompe P1. Chaque actionneur a deux états (Ouvert = 1, Fermé = 0). L'objectif du contrôleur dans ce cas est de garder le niveau de liquide m_1 dans le réservoir R1 au-dessous du seuil M1 et le niveau de liquide m_2 dans le réservoir R2 au-dessus du seuil M2 en remplissant et en vidant alternativement les deux réservoirs. Pour cette raison, trois capteurs de niveaux M1, M2 et MM2 ont été utilisés. Ces capteurs sont considérés comme des capteurs binaires (par exemple, $M1 = 1$ si $h \geq m_1$ et $M1 = 0$ sinon). Ainsi, le système est équipé de deux capteurs logiques C1 et C2 qui donnent des informations sur l'état des actionneurs P1 et V1 respectivement.

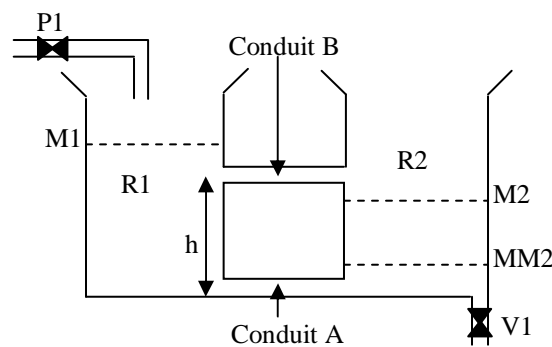


FIGURE 2.3 – Système à deux réservoirs

Les signaux des capteurs équipant ce système sont considérés comme ses signaux de sortie. Le regroupement de ces signaux de sortie dans un vecteur $E = \begin{pmatrix} M1 \\ M2 \\ MM2 \\ C1 \\ C2 \end{pmatrix}$ forme le vecteur d'état mesurable de ce système.

Ainsi, les signaux associés aux actionneurs constituent les signaux d'entrée de ce système. Les événements mesurables générés par ce système et associés à ses signaux d'entrée sont : $P1^\uparrow$ ($P1 =$ ouvert), $P1^\downarrow$ ($P1 =$ fermé), $V1^\uparrow$ ($V1 =$ ouvert), $V1^\downarrow$ ($V1 =$ fermé). Chaque changement au niveau des signaux d'entrée provoque au moins un

changement d'état du système. Par exemple, si l'on considère que le niveau du liquide dans le réservoir R2 est entre M1 et MM2, le vecteur d'état initial aurait la valeur $E_0 = [0 \ 0 \ 1 \ 0 \ 0]^T$. Dans ce cas, le contrôleur doit donner l'ordre d'ouvrir la pompe P1 en générant l'événement $P1^\uparrow$.

2.4 Proposition d'une approche d'identification à base de signaux de sortie

Le travail présenté dans cette section fait partie des travaux qui exploitent les symboles de sortie du système pour établir son modèle. Parmi ces travaux, on peut citer ceux de Meda *et al.* [MRM98], [MM02], [MM05]. Ces derniers se basent sur des hypothèses restrictives sur l'ordre d'occurrence des événements et suppose également que l'on possède un modèle RdP du système à identifier. Pour remédier à ces limites, nous proposons cette première approche qui permet d'établir un modèle comportemental d'un SED en s'affranchissant des hypothèses précédentes. Cette approche se base sur l'utilisation des signaux de sortie et permet de trouver la structure (les matrices *Post* et *Pre*) ainsi que le marquage initial du RdP qui modélise le système considéré. Notons que, les algorithmes d'identification proposés dans la suite de cette section sont valables sous les hypothèses suivantes :

Hypothèses :

- (H1) Tous les événements du système à identifier sont détectables ([AIE03], [ABT02]).
- (H2) Le système considéré est modélisable par un RdP ordinaire pur.

Dans cette section, le problème d'identification à traiter est présenté et l'approche proposée pour le résoudre est exposée. Ensuite, les conditions nécessaires et suffisantes garantissant l'unicité de la solution sont déterminées. Par ailleurs, le problème d'identification est décrit sous la forme d'un problème d'optimisation équivalent afin de traiter les cas où la solution unique n'existe pas.

2.4.1 Définition du problème

Dans cette approche les données issues des capteurs sont enregistrées sous la forme d'une succession de vecteurs d'état mesurables atteignables par le système à identifier. Cette séquence d'états peut être représentée par $\mathcal{E} = E_{\alpha_0}, E_{\alpha_1}, E_{\alpha_2}, \dots, E_{\alpha_p}$ avec E_{α_i} le vecteur d'état mesurable du système et $i \in \{0, \dots, p\}$ est un indice indiquant l'ordre d'enregistrement du vecteur d'état E_{α_i} . Le problème d'identification considéré dans ce cas peut être défini comme suit :

Définition 2 : Etant donné un SED S qui satisfait les hypothèses introduites précédemment et soit la séquence $\mathcal{E} = E_{\alpha_0}, E_{\alpha_1}, E_{\alpha_2}, \dots, E_{\alpha_p}$ qui représente l'ensemble des vecteurs d'état E_{α_i} enregistrés en observant S . Le problème d'identification de ce SED par RdP consiste à déterminer l'ensemble de places P , l'ensemble de transitions T , le marquage initial M_0 et la structure du RdP (les matrices $Post$ et Pre) qui modélise ce système.

Pour résoudre ce problème, nous proposons de représenter la séquence de vecteurs d'état \mathcal{E} par une séquence de marquage $\mathcal{M} = M_{\alpha_0}, M_{\alpha_1}, M_{\alpha_2}, \dots, M_{\alpha_k}, \dots, M_{\alpha_p}$. Cette dernière est obtenu en associant chaque vecteur d'état E_{α_i} atteignable par le SED à identifier à un vecteur de marquage M_{α_i} atteignable par le RdP à établir. Ainsi, nous pouvons déjà obtenir une première information sur le nombre minimal de places dans le RdP à identifier. Ce nombre représente le nombre de places mesurables et est égal à la dimension de la partie mesurable du vecteur de marquage. Il nous reste maintenant de déterminer le nombre transitions, le nombre de places non mesurables, la structure du RdP et le marquage initial.

2.4.2 Détermination de l'ensemble des transitions du système

Un SED change son état après l'occurrence d'au moins un événement (activation et/ou désactivation d'un ou plusieurs actionneurs). Ainsi, la détermination de l'ensemble d'événements, qui ont eu lieu lors de l'enregistrement des données d'identification, peut être obtenue en analysant la séquence \mathcal{E} afin d'en tirer une séquence d'événements notée $E_v = e_1, e_2, \dots, e_{k-1}$. Ces événements sont ensuite associés aux transitions constituant le RdP à établir. Par ailleurs, tenant compte du fait que chaque transition du RdP à établir peut être associée à un ou plusieurs événements, la séquence de transitions $\sigma = t_1, t_2, \dots, t_n$ peut être obtenue à partir de E_v .

Remarque 1 : Si l'on note par $|\sigma \cap t_i|$ le nombre d'apparitions de la transition t_i dans σ alors, $\forall t_i \in \sigma : |\sigma \cap t_i| \geq 1$.

Exemple 4 : On reprend l'exemple 7 du chapitre 1 présenté par Meda *et al.* [MRM98], [MTM00]. Soit la séquence d'états suivante générée par le système à identifier : $\mathcal{E} =$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

À partir de cette séquence d'états, la séquence d'événements suivante est détectée :

$$E_v = e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}.$$

L'analyse des vecteurs d'états en aval et en amont de chaque événement permet d'associer : aux événements e_1, e_6 et e_{11} la transition t_1 , aux événements e_2, e_9 et e_{12} la transition t_2 , aux événements e_3, e_{10} et e_{14} la transition t_3 , aux événements e_4, e_7 et e_{13} la transition t_4 et aux événements e_5, e_8 et e_{15} la transition t_5 . Donc, la séquence de transitions associée à cette séquence d'événements est $\sigma = t_1, t_2, t_3, t_4, t_5, t_1, t_4, t_5, t_2, t_3, t_1, t_2, t_4, t_3, t_5$ et l'ensemble de transitions est $T = \{t_1, t_2, t_3, t_4, t_5\}$.

Après avoir déterminé l'ensemble des transitions, dans ce qui suit, nous proposons d'étudier les conditions nécessaires et suffisantes pour obtenir une solution unique. Ces conditions portent principalement sur une sous-séquence de vecteurs de marquages, notée \mathcal{M}' , obtenue à partir de la séquence de marquages \mathcal{M} associée à la séquence d'état observée \mathcal{E} . Notons que l'obtention de \mathcal{M}' permettrait d'optimiser le temps de calcul nécessaire pour l'identification. En effet, comme le système est observé durant plusieurs cycles de fonctionnement, il se peut que certains marquages de la séquence \mathcal{M} soient redondants et n'apportent pas d'informations supplémentaires permettant d'affiner le modèle identifié.

Une fois les conditions garantissant l'unicité de la solution sont obtenues, nous étudierons, ensuite, les cas où ils ne sont pas satisfaites en formulant le problème d'identification sous la forme d'un problème d'optimisation discrète.

2.4.3 Etude de l'unicité de la solution

L'utilisation de la séquence de marquages \mathcal{M} associée à la séquence de vecteurs d'état \mathcal{E} et celle de transitions σ établies précédemment permet de caractériser l'évolution des marquages du RdP dont on cherche la structure. Nous pouvons représenter cette évolution comme suit : $M_{\alpha_0}[t_1 >, M_{\alpha_1}[t_2 >, \dots, [t_n > M_{\alpha_P}$. Dans cette section, nous discutons les conditions nécessaires et suffisantes qui doivent être vérifiées par la séquence de marquages \mathcal{M} pour garantir l'unicité de la solution du problème d'identification introduit dans la définition 2.

Une des conditions pour que le RdP identifié soit unique est que la séquence de marquage \mathcal{M} doit autoriser le franchissement d'au moins une séquence de transitions complète. Ceci est motivé par la possibilité de détecter l'occurrence de toutes les transitions du RdP à établir. En effet, à partir de la séquence de transitions détectée depuis \mathcal{M} , nous cherchons au moins une séquence de transitions complète. Notons qu'une séquence de transitions est dite complète si elle autorise le franchissement de toutes les transitions du système au moins une fois. Par conséquent, à partir de la séquence \mathcal{M} il est possible de sélectionner au moins une sous-séquence de marquages

$\mathcal{M}' = M_{\alpha_j}, M_{\alpha_{j+1}}, \dots, M_{\alpha_k}$ définie comme suit :

$$\mathcal{M}' = \{M_{\alpha_i} | \forall M_{\alpha_i} \in \mathcal{M}' : M_{\alpha_i} \in \mathcal{M} \text{ et } \forall M_{\alpha_i} \in \mathcal{M}' : |\mathcal{M}' \cap M_{\alpha_i}| \geq 1\}.$$

Une autre condition qui doit être satisfaite par la séquence \mathcal{M} concerne le marquage initial M_0 . Ce dernier doit être complètement mesurable et appartient à la séquence \mathcal{M} . En effet, pour avoir une solution unique il faut que le marquage initial soit complètement connu. De plus, les marquages de la sous-séquence \mathcal{M}' doivent être complètement mesurables. Dans ce cas, le nombre de places du modèle RdP à identifier est égal à la dimension du vecteur de marquage.

Nous introduisons dans ce qui suit un théorème qui caractérise la sous-séquence \mathcal{M}' et le marquage initial M_0 pour obtenir une solution unique.

Théorème 1 : Le problème d'identification décrit par la définition 2 admet une solution unique si et seulement si les conditions suivantes sont satisfaites simultanément :

1. Le marquage initial M_0 du modèle RdP à identifier doit être complètement mesurable et appartenant à la séquence de marquage \mathcal{M} .
2. Chaque marquage $M_{\alpha_i} \in \mathcal{M}'$ doit être complètement mesurable.
3. La sous-séquence \mathcal{M}' permet de détecter une séquence complète de transitions définie comme suit :

$$\forall t_i \in T : t_i \in \sigma' \text{ ce qui implique que } |\sigma'| = n \text{ et, } \forall t_i \in \sigma' : |\sigma' \cap t_i| = 1.$$

Démonstration :

1. Suffisance : La première condition concerne l'état initial du système à identifier, cet état doit être complètement connu. En termes de RdP, on peut dire que le marquage initial du RdP à identifier supposé associé à l'état initial du système doit être complètement mesurable. En effet, si M_0 est complètement connu alors les places qui constitue le RdP à identifier l'est aussi avec leurs marquages initiaux.

Nous avons supposé que la séquence des vecteurs d'état \mathcal{E} atteignables par le système à identifier permet de détecter l'occurrence de tous les événements de ce système du fait que celui-ci est à événements détectables. Donc, la séquence de marquages \mathcal{M} associée à la séquence de vecteurs d'état \mathcal{E} autorise le franchissement de toutes les transitions du système au moins une fois. Par conséquent, \mathcal{M} peut contenir au moins une sous-séquence \mathcal{M}' qui autorise le franchissement d'au moins une sous-séquence de transitions complète σ' . En plus, le nombre d'apparition de chaque transition dans cette sous-séquence est égal à 1 ($\forall t_i \in \sigma' : |\sigma' \cap t_i| = 1$) pour éviter la redondance de calcul du vecteur d'incidence de la transition considérée ($W(., t_i)$).

Les équations citées ci-dessous seront vérifiées vu que les marquages de la sous-séquence \mathcal{M}' sont complètement mesurables.

$$\forall t_i \in \sigma', \forall M_{\alpha_i} \in \mathcal{M}' : W(\cdot, t_i) = M_{\alpha_i} - M_{\alpha_{i-1}} \quad (2.4)$$

où M_{α_i} et $M_{\alpha_{i-1}}$ sont respectivement le marquage en aval et le marquage en amont de la transition t_i dans la séquence de marquages et transitions qui représente l'évolution du RdP dont on cherche la structure.

Le fait que le SED à identifier est supposé modélisable par un RdP ordinaire pur, l'équation suivante est satisfaite :

$$\forall t_i \in \sigma', \forall p \in P : W(p, t_i) \in \{-1, 0, 1\} \quad (2.5)$$

En utilisant l'équation (2.5) et $\forall p \in P, \forall t_i \in \sigma'$ on a :

$$\begin{cases} \text{Si } W(p, t_i) = 1 \Rightarrow Post(p, t_i) = W(p, t_i) = 1 \text{ et } Pret(p, t_i) = 0, \\ \text{Si } W(p, t_i) = 0 \Rightarrow Post(p, t_i) = Pret(p, t_i) = W(p, t_i) = 0, \\ \text{Si } W(p, t_i) = -1 \Rightarrow Post(p, t_i) = 0 \text{ et } Pret(p, t_i) = -W(p, t_i), \end{cases} \quad (2.6)$$

Ces équations permettent de déterminer les $2 \times n$ colonnes des matrices *Post* et *Pre*.

2. *Nécessité* : Supposant que la solution est unique (les matrices *Pre* et *Post* sont uniques), on examine les situations suivantes :

(a) Si la condition 3 du théorème 1 n'est pas satisfaite, \mathcal{M}' ne permet pas de détecter une sous-séquence de transitions complète alors, $|\sigma'| < n$. Dans ce cas, l'application des équations (2.4) (2.5) et (2.6) ne permet pas de donner toutes les colonnes des matrices *Pre* et *Post*. En effet, le fait que la séquence de transitions est incomplète nous ne pouvons déterminer qu'une partie de la structure du modèle RdP à identifier. La partie restante qui concerne les transitions manquantes doit être estimée.

(b) Si la condition 3 du théorème 1 est satisfaite et la condition 2 ne l'est pas alors, en appliquant l'équation (2.6) il n'est pas possible d'obtenir une solution unique. En effet, si la séquence de marquages \mathcal{M}' permet de détecter une séquence complète de transitions mais en même temps les marquages de cette séquence sont partiellement mesurables alors, nous ne pouvons déterminer également qu'une partie de la structure de RdP à identifier. La partie restante qui concerne les places manquantes et ses marquages initiaux doit être estimée.

(c) Si les conditions 2 et 3 du théorème 1 sont satisfaites mais la condition 1 n'est pas vérifiée alors, l'unicité de la solution n'est pas garantie. En effet, si l'état initial du système considéré est partiellement connu ou complètement inconnu donc, il sera

estimé. L'estimation de l'état initial du système peut donner plusieurs valeurs pour cet état.

Remarque 2 : La condition 3 du théorème 1 peut être relaxée comme suit :

3'. La sous-séquence de marquages \mathcal{M}' permet de détecter une séquence complète de transitions définie comme suit :

$\forall t_i \in T : t_i \in \sigma'$ ce qui implique que $|\sigma'| = n$ et, $\forall t_i \in \sigma' : |\sigma' \cap t_i| \geq 1$.

En effet, si on n'arrive pas à trouver une sous-séquence de marquages \mathcal{M}' qui permet de détecter une séquence complète de transitions telle que chaque transition t_i dans l'ensemble de transitions apparaisse exactement une fois dans cette séquence complète, nous devons relaxer cette contrainte. Cette relaxation peut nécessiter plus d'opérations de calcul à faire mais elle n'influe en aucun cas sur l'unicité de la solution.

Exemple 5 : Soit l'ensemble des vecteurs d'état, atteignables par le SED dont on cherche le modèle, enregistrés sous la forme suivante :

$$\mathcal{E} = \begin{pmatrix} 0 \\ 0 \\ 3 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 3 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \\ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 3 \\ 2 \\ 0 \end{pmatrix}.$$

La première étape de cette approche consiste à déterminer l'ensemble des transitions T qui constitué le modèle RdP à identifier. La séquence d'événements détectables à partir de \mathcal{E} est $E_v = e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}$. En associant à chaque état de la séquence \mathcal{E} un marquage atteignable par le RdP à identifier et en analysant les vecteurs de marquages associés aux vecteurs d'état en amont et ceux en aval de chaque événement dans la séquence d'événements E_v , on peut déterminer la transition associée à chaque événement. En effet, en analysant les vecteurs de marquages associés aux vecteurs d'état en amont et en aval des événements e_1, e_3, e_9 et e_{10} , on peut associer à ces événements la transition t_1 . Aussi, on associe aux e_2, e_6, e_{11} et e_{12} la transition t_2 , aux e_4, e_7, e_{13} et e_{14} la transition t_3 et aux événements e_5, e_8, e_{15} et e_{16} la transition t_4 pour la même raison. Par conséquent, la séquence de transitions détectée à partir de la séquence d'états \mathcal{E} est $\sigma = t_1, t_2, t_1, t_3, t_4, t_2, t_3, t_4, t_1, t_1, t_2, t_2, t_3, t_3, t_4, t_4$ et l'ensemble de transitions du RdP à identifier est $T = \{t_1, t_2, t_3, t_4\}$ avec $n = 4$. Le fait que les marquages associés aux vecteurs d'état du système sont complètement mesurables alors, le nombre de places dans le modèle RdP à identifier est égal à la dimension du vecteur d'état $m = 5$.

Dans la deuxième étape nous cherchons à caractériser une sous-séquence de marquages permettant de garantir l'unicité de la solution. Le fait que chaque état de la séquence \mathcal{E} est associé à un marquage atteignable par le RdP à identifier alors la séquence de marquages associée à cette séquence d'états est : $\mathcal{M} = M_0, M_1, M_2, M_3, M_4, M_1, M_2, M_5, M_0, M_1, M_6, M_3, M_7, M_8, M_9, M_5, M_0$. En analysant \mathcal{M} nous remarquons qu'il existe une sous-séquence de marquages $\mathcal{M}' = M_1, M_2, M_3, M_4, M_1$ qui permet de détecter toutes les transitions du RdP à identifier au moins une fois. La séquence \mathcal{M}' permet de détecter une séquence de transitions $\sigma' = t_2, t_1, t_3, t_4$ qui vérifie les conditions du théorème 1 donc, \mathcal{M}' est la séquence que nous cherchons. De plus, le marquage initial est complètement mesurable et appartient à la séquence de marquages \mathcal{M} . On peut conclure que l'évolution des marquages équivalents à la séquence de transitions complète σ' est $M_1[t_2 > M_2[t_1 > M_3[t_3 > M_4[t_4 > M_1$. En appliquant l'équation

$$(2.4), \text{ nous obtenons : } W(., t_2) = M_2 - M_1 = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, W(., t_1) =$$

$$M_3 - M_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -1 \\ -1 \\ 0 \end{pmatrix}, W(., t_3) = M_4 - M_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

$$W(., t_4) = M_1 - M_4 = \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ -1 \end{pmatrix}.$$

Puis, en appliquant les équations (2.5) et (2.6) nous obtenons les éléments des ma-

$$\text{trices } Pre \text{ et } Post : Pre = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, Post = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Finalement, le RdP identifié est représenté dans la figure 2.4. L'analyse des différents comportements de ce RdP permet de conclure qu'il génère exactement les comportements observés du système considéré.

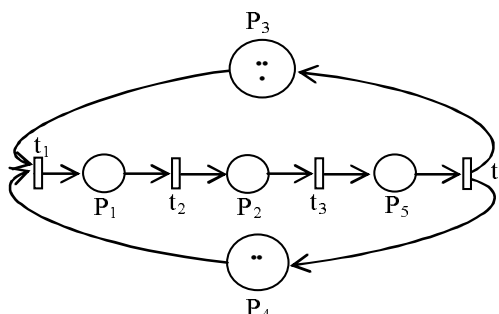


FIGURE 2.4 – Le modèle RdP identifié

2.4.4 Cas général

Nous avons présenté dans le théorème 1 les conditions nécessaires et suffisantes pour l'unicité de la solution. Cependant, ces conditions restent très contraignantes d'un point de vue pratique car elles nécessitent une instrumentation dense en capteurs. Pour remédier à cet inconvénient, nous présentons dans la suite des techniques permettant d'identifier le système en utilisant des informations partielles sur les états du système. Ainsi, nous nous intéressons, dans un premier temps, au cas où l'on arrive à déterminer une séquence de transitions complète mais on ne dispose pas des mesures de tous les états. Ensuite, nous considérons le cas où les états du système sont partiellement mesurables et/ou la séquence de transitions est incomplète. Dans les deux cas, le problème d'identification est défini sous la forme d'un problème d'optimisation, caractérisé par une fonction objectif et des contraintes à respecter. La fonction objectif représente l'indice de performance [Sre06] qui permet de sélectionner la meilleure solution parmi celles réalisables. Quant aux contraintes, elles sont formulées en utilisant certaines propriétés structurelles du RdP. Ensuite, le problème d'identification défini sera résolu, à chaque fois, en utilisant la PLB. Le choix de cette technique est motivée par le fait que la classe de SED à identifier est supposée modélisable par des RdP ordinaires, ce qui implique que les éléments des matrices Pre et $Post$ appartiennent à l'ensemble $\{0,1\}$.

2.4.4.1 Séquence de marquages partiellement mesurables et séquence de transitions complète

Cette situation correspond au cas où les conditions (1) et (2) du théorème 1 ne sont pas vérifiées (i.e. ; le marquage initial et les marquages de la sous-séquence \mathcal{M}' sont partiellement mesurables) et en même temps la condition (3) du même théorème est vérifiée. Ce cas est caractérisé par la connaissance du nombre de transitions du modèle à identifier (i.e., la séquence de transitions est complète) et par le nombre de places

mesurables (i.e., égal à la dimension du vecteur de marquages mesurables). Ainsi, le problème d'identification dans ce cas concerne le calcul de la structure du RdP, incluant les places non mesurables qu'on doit déterminer et l'estimation du marquage initial. Pour résoudre ce problème nous procédons comme suit :

1. Initialiser le nombre de places non mesurables à 1.
2. Définir le problème d'identification considéré sous la forme d'un problème de PLB, comme décrit dans le théorème 2.
3. Résoudre le problème de PLB défini précédemment (en utilisant C++ et Cplex).
4. Si la solution obtenue n'est pas optimale par rapport à l'objectif alors incrémenter le nombre de places non mesurables et revenir à l'étape 2. Sinon, arrêter la procédure d'identification.

Notons que lors de la définition du problème de PLB, la fonction objectif suivante sera utilisée :

$$f(Pre, Post, M_0) = \sum_{i=1}^m \sum_{j=1}^n (a_{i,j} \cdot Pre(p_i, t_j) + b_{i,j} \cdot Post(p_i, t_j)) + \sum_{i=1}^m c_i \cdot M_0(p_i) \quad (2.7)$$

où : $a_{i,j}$, $b_{i,j}$, c_i sont des coefficients de pondération à choisir par l'utilisateur.

Ce choix est motivé par le fait que le RdP à établir doit être minimal en nombre d'arcs et de marques dans le marquage initial. Enfin, en donnant la même ordre d'importance à ces coefficients, la fonction objectif (2.7) peut être réécrite sous la forme :

$$f(Pre, Post, M_0) = \vec{1}_{1 \times m} \cdot (Pre_{n \times m} + Post_{m \times n}) \cdot \vec{1}_{n \times 1} + \vec{1}_{1 \times m} \cdot M_0 \quad (2.8)$$

Théorème 2 : Soit S un SED caractérisé par des événements détectables et modélisable par un RdP ordinaire et pur noté $N(P, T, Pre, Post)$ à identifier. Soit \mathcal{M}' est une séquence de marquages (états) partiellement mesurables, atteignables par N et vérifiant $\forall M_{\alpha_i} \in \mathcal{M}' : M_{\alpha_i} \in R(N, M_0)$. \mathcal{M}' permet de détecter une sous-séquence complète de transitions σ' . Une solution optimale du problème d'identification décrit par la définition 2 est obtenue en résolvant le système linéaire suivant :

$$\min f(Pre, Post, M_0) \quad (2.9)$$

en respectant les contraintes :

$$\xi(\mathcal{M}', m, n) =$$

$$\left\{ \begin{array}{ll} \forall t_i \in T, \forall p \in P : Pre(p, t_i) \in \{0, 1\} \text{ et } Post(p, t_i) \in \{0, 1\} & (a) \\ Pre_{m \times n}^T \times \vec{1}_{m \times 1} + Post_{m \times n}^T \times \vec{1}_{m \times 1} \geq \vec{1}_{n \times 1} & (b) \\ Pre_{m \times n} \times \vec{1}_{n \times 1} + Post_{m \times n} \times \vec{1}_{n \times 1} \geq \vec{1}_{m \times 1} & (c) \\ Post_{m \times n} + Pre_{m \times n} \leq \vec{1}_{m \times n} & (d) \\ \forall M_{\alpha_i} \in \mathcal{M}', \forall t_i \in \sigma' : Post(., t_i) - Pre(., t_i) = M_{\alpha_i} - M_{\alpha_{i-1}} & (e) \\ \forall t_i \in T, \forall p \in P : Pre(., t_i) \leq M_{\alpha_{i-1}} & (f) \\ M_0 \in \mathbb{N}^m \text{ et } \forall M_{\alpha_i} \in \mathcal{M}' : M_{\alpha_i} \in \mathbb{N}^m & (j) \\ M_0 - Pre(., t_i) \geq \vec{0}_{m \times 1} & (h) \\ Pre(., t_i) \leq M_0 & (i) \end{array} \right. \quad (2.10)$$

Démonstration :

La contrainte (2.10)(a) est évidente parce que le SED à identifier est supposé modélisable par un RdP ordinaire. Pour que le modèle RdP identifié soit significatif et capable de générer les comportements du système considéré il doit être connexe (i.e., sans places et/ou transitions isolées). Donc, chaque transition t_i dans ce RdP doit être au moins en aval ou en amont d'une place p . Ceci signifie que la somme des éléments de chaque colonne des matrices Pre ou bien $Post$ doit être supérieure ou égale à 1. Mathématiquement, on peut dire : $\forall t_i \in T, \sum Pre(., t_i) \geq 1$ ou $\sum Post(., t_i) \geq 1 \Rightarrow \sum Pre(., t_i) + \sum Post(., t_i) \geq 1 \Rightarrow Pre_{m \times n}^T \times \vec{1}_{m \times 1} + Post_{m \times n}^T \times \vec{1}_{m \times 1} \geq \vec{1}_{n \times 1}$. En plus, chaque place p dans un RdP connexe doit être au moins en aval ou en amont d'une transition t , ce qui implique que la somme des éléments de chaque ligne des matrices Pre ou bien $Post$ doit être supérieure ou égale à 1. Mathématiquement, on peut représenter cette contrainte comme suit :

$$\forall p \in P, \sum Pre(p, .) \geq 1 \text{ ou } \sum Post(p, .) \geq 1 \Rightarrow \sum Pre(p, .) + \sum Post(p, .) \geq 1 \Rightarrow Pre_{m \times n} \times \vec{1}_{n \times 1} + Post_{m \times n} \times \vec{1}_{n \times 1} \geq \vec{1}_{m \times 1}.$$

Donc, cette propriété de connexité est représentée par les contraintes (2.10)(b),(c).

Le SED à identifier est modélisable par un RdP ordinaire et pur, chaque transition t dans ce RdP ne doit pas être en amont et en aval de la même place p . Ceci signifie que l'on ne doit pas avoir le cas où $Pre(p, t) = Post(p, t) = 1$. Cette propriété est décrite par la contrainte (2.10)(d).

Par ailleurs, les contraintes (2.10)(e),(f) permettent de vérifier l'équation fondamentale régissant le comportement du RdP à identifier. Quant à la contrainte (2.10)(j), elle permet de garantir que les marquages atteignables par le RdP à établir appartiennent à l'ensemble \mathbb{N}^m (i.e, le système à identifier est modélisable par un RdP ordinaire). De plus, en utilisant l'équation de marquage d'un RdP, on peut formuler les

contraintes (2.10)(h)(i) concernant le franchissement d'une transition t_i depuis le marquage initial M_0 . Les équations (2.10)(h)(i) seront utilisées pour estimer la partie non mesurable du marquage initial en connaissant la transition franchie depuis ce dernier. \square

D'autres contraintes peuvent être ajoutées au système linéaire défini par l'équation (2.10) si d'autres propriétés doivent être satisfaites par le RdP à établir. Par exemple, si le RdP ne doit pas contenir des places et/ou des transitions puits, les contraintes suivantes doivent être ajoutées respectivement :

$$Pre. \vec{1}_{n \times 1} \geq \vec{1}_{m \times 1} \quad (2.11)$$

$$Post^T. \vec{1}_{m \times 1} \geq \vec{1}_{n \times 1} \quad (2.12)$$

Ainsi, si le modèle RdP est sans places et/ou transitions sources alors, les contraintes suivantes doivent être ajoutées respectivement au modèle linéaire de l'équation (2.10) :

$$Post. \vec{1}_{n \times 1} \geq \vec{1}_{m \times 1} \quad (2.13)$$

$$Pre^T. \vec{1}_{m \times 1} \geq \vec{1}_{n \times 1} \quad (2.14)$$

Exemple 6 : Afin d'illustrer notre proposition, considérons le système présenté dans [MRM98] (cf. exemple 7 chapitre 1, exemple 4 du chapitre 2). Ce système est caractérisé par la séquence de marquages suivante : $\mathcal{M} = M_0, M_1, M_2, M_3, M_4, M_0, M_1, M_5, M_6, M_7, M_0, M_1, M_2, M_8, M_4, M_0$.

L'analyse de cette séquence permet de détecter la séquence de transitions $\sigma = t_1, t_2, t_3, t_4, t_5, t_1, t_4, t_5, t_2, t_3, t_1, t_2, t_4, t_3, t_5$ et de conclure que l'ensemble de transitions est $T = \{t_1, t_2, t_3, t_4, t_5\}$. L'utilisation de la séquence \mathcal{M} permet également de caractériser une sous séquence \mathcal{M}_1 permettant de détecter une sous séquence de transi-

tions complète noté σ' . Ces deux dernières sont respectivement : $\mathcal{M}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix},$

$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ et $\sigma' = t_1 t_2 t_3 t_4 t_5$.

Partant de l'hypothèse que le système est partiellement mesurable et caractérisé par un marquage initial partiellement mesurable $M_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ (i.e, le RdP à identifier

possède 4 places mesurables).

L'application de la procédure d'identification décrite au début de la sous section §2.4.3.1 permet d'obtenir initialement les deux modèles présentés par les figures 2.5 et 2.6. Notons que ces deux modèles sont obtenus en satisfaisant la fonction décrite par l'équation (2.8) et en vérifiant respectivement les contraintes (2.10), (2.14) et (2.10), (2.12), (2.14).

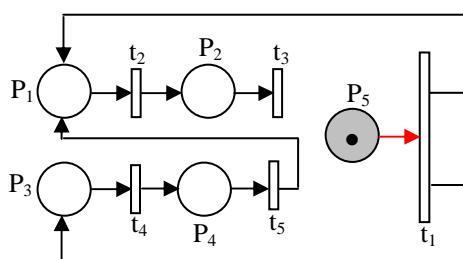


FIGURE 2.5 – Le modèle identifié 1 avec une place non mesurable

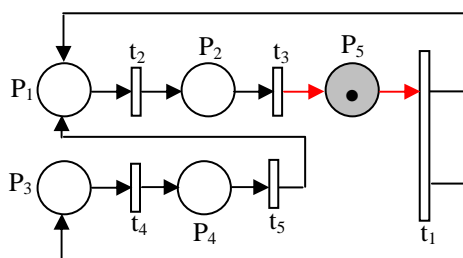


FIGURE 2.6 – Le modèle identifié 2 avec une place non mesurable

L'analyse des comportements du modèle RdP de la figure 2.5 permet de constater que ce dernier ne peut pas générer tous les comportements du système considéré. Ceci revient au fait que le modèle devient non vivant dès le franchissement de t_4 . Quant au modèle de la figure 2.6, il peut atteindre tous les états du système considéré mais, il peut également générer d'autres états qui ne sont pas équivalents à ceux de ce système. Donc, ces deux modèles ne peuvent pas être considérés comme de bons modèles de ce système.

En incrémentant le nombre de places non mesurables à 2, nous obtenons les RdP représentés par les figures 2.7 et 2.8.

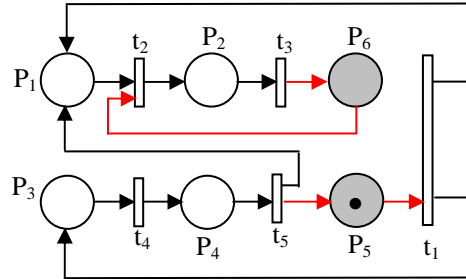


FIGURE 2.7 – Un modèle RdP identifié avec deux places non mesurables

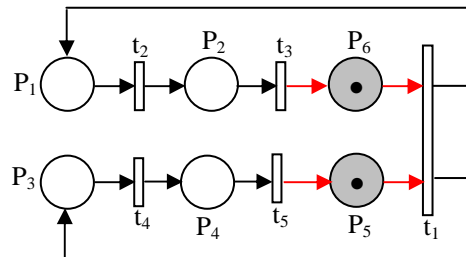


FIGURE 2.8 – La solution optimale

L'analyse des comportements générés par le RdP de la figure 2.7 montre qu'il génère tous les comportements du système mais, il génère aussi d'autres comportements complètement différents de ceux du système considéré. Donc, ce RdP n'est pas le bon modèle de ce système.

En revanche, l'analyse du modèle de la figure 2.8 montre qu'il génère exactement les comportements observés du système considéré. Ce modèle RdP est ainsi retenu à la fin de la phase d'identification. Notons que, la comparaison de ces résultats avec ceux de Meda *et al.* [MTM00] [MRM98] montre que l'on trouve exactement le même modèle.

2.4.4.2 Séquence de marquages partiellement mesurables et/ou séquence de transitions incomplète

Cette situation correspond au cas où les conditions (1) et (2) du théorème 1 sont vérifiées et la condition (3) du même théorème ne l'est pas (i.e., tous les marquages sont complètement mesurables mais les données d'identification ne permettent pas de détecter une séquence de transitions complète). Elle peut également correspondre au cas où les trois conditions ne sont pas vérifiées à la fois (i.e, les marquages sont partiel-

lement mesurables et les données d'identification ne permettent pas de détecter une séquence de transitions complète). Notons que, l'ensemble complet de données enregistrées est utilisé, dans ces deux situations, pour l'identification du système ($\mathcal{M}' = \mathcal{M}$).

Ainsi, dans le premier cas, le problème d'identification concerne principalement la détermination du nombre optimal de transitions du RdP à établir et le calcul de la structure de ce dernier. Par ailleurs, le deuxième cas concerne à la fois la détermination du nombre de places non mesurables, le nombre de transitions, le calcul de la structure du RdP et l'estimation du marquage initial. Pour résoudre ces deux problèmes nous proposons d'appliquer les deux procédures suivantes :

Procédure d'identification dans le cas où la séquence de marquages est complètement mesurable et la séquence de transitions est incomplète :

Début :

1. Fixer une borne supérieure \bar{n} sur le nombre de transitions.
2. Initialiser le nombre de transitions à déterminer : ce nombre est égal au nombre de transitions détectées à partir de la séquence de marquages \mathcal{M} .
3. Définir le problème d'identification considéré sous la forme d'un problème de PLB, comme décrit dans le théorème 3.
4. Résoudre le problème de PLB défini précédemment.
5. Si la solution obtenue n'est pas optimale par rapport à l'objectif alors incrémenter le nombre de transitions et revenir à l'étape 2. Sinon, arrêter la procédure d'identification.

Fin

Notons que lors de la définition du problème de PLB, la fonction objectif suivante sera utilisée :

$$f(n, Pre, Post) = \vec{1}_{1 \times m} \cdot (Pre_{m \times n} + Post_{m \times n}) \cdot \vec{1}_{n \times 1} \quad (2.15)$$

Ce choix est motivé par le fait que le RdP à établir doit être minimal en nombre de transitions et d'arcs.

Théorème 3 : Soit S un SED caractérisé par des événements détectables et modélisable par un RdP ordinaire et pur noté $N(P, T, Pre, Post)$ à identifier. Soit \mathcal{M} une séquence de marquages (états) complètement mesurables, atteignables par N et vérifiant $\forall M_{\alpha_i} \in \mathcal{M} : M_{\alpha_i} \in R(N, M_0)$. \mathcal{M} permet de détecter une séquence incomplète de transitions σ' . Une solution optimale du problème d'identification décrit par la définition 2 est

obtenue en résolvant le système linéaire suivant :

$$\begin{cases} \min_{n \leq \bar{n}} \min f(n, Pre, Post) \\ s.t''(2.10)'' \end{cases} \quad (2.16)$$

Procédure d'identification dans le cas où la séquence de marquages est partiellement mesurable et la séquence de transitions est incomplète :

Début :

1. Fixer une borne supérieure \bar{n} sur le nombre de transitions.
2. Fixer une borne supérieure \bar{m} sur le nombre de places.
3. Initialiser le nombre de place dans le modèle RdP à identifier au nombre de places mesurables.
4. Initialiser le nombre de transitions dans le modèle RdP à identifier au nombre de transitions détectées depuis la séquence de marquages \mathcal{M} .
5. Tant que le nombre de transitions n est inférieur ou égale à \bar{n} faire :
 - (a) Tant que le nombre de places m est inférieur ou égale à \bar{m} faire :
 - i. Incrémenter le nombre de places.
 - ii. Définir le problème d'identification considéré sous la forme d'un problème de PLB, comme décrit dans le théorème 4.
 - iii. Résoudre le problème de PLB défini précédemment.
 - iv. Si la solution obtenue n'est pas optimale par rapport à l'objectif alors revenir à l'étape (a). Sinon, incrémenter le nombre de transitions et aller à l'étape (5).

Fin Tq.

Fin Tq.

Fin

Notons que lors de la définition du problème de PLB, la fonction objectif suivante sera utilisée :

$$f(n, Pre, Post, M_0) = \vec{1}_{1 \times m} \cdot (Pre_{m \times n} + Post_{m \times n}) \cdot \vec{1}_{n \times 1} + \vec{1}_{1 \times m} \cdot M_0 \quad (2.17)$$

Ce choix est motivé par le fait que le RdP à établir doit être minimal en nombre de transitions, d'arcs et de marques dans le marquage initial.

Théorème 4 : Soit S un SED caractérisé par des événements détectables et modélisable par un RdP ordinaire et pur noté $N(P, T, Pre, Post)$ à identifier. Soit \mathcal{M} une

séquence de marquages (états) partiellement mesurables, atteignables par N et vérifiant $\forall M_{\alpha_i} \in \mathcal{M} : M_{\alpha_i} \in R(N, M_0)$. \mathcal{M} permet de détecter une séquence incomplète de transitions σ' . Une solution optimale du problème d'identification décrit par la définition 2 est obtenue en résolvant le système linéaire suivant :

$$\begin{cases} \min_{n \leq \bar{n}} \min f(n, Pre, Post, M_0) \\ s.t'' (2.10)'' \end{cases} \quad (2.18)$$

Exemple 7 : Afin d'illustrer notre proposition, considérons le système présenté dans [MRM98] (cf. exemple 7 chapitre 1, exemple 4 du chapitre 2) et supposons que l'on dispose uniquement de la séquence de marquages partiellement mesurables suivante :

$$\mathcal{M}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}. \text{ L'analyse de cette séquence permet de détecter une}$$

séquence de transitions incomplète $\sigma = t_1 t_2 t_3$ et de conclure que le RdP à identifier sera composé au moins de transitions t_1, t_2 , et t_3 . L'application de la procédure d'identification relative au théorème 4 permet d'obtenir les résultats suivants :

1. Pour un modèle identifié avec 4 transitions ($\bar{n} = 4$) et une place non mesurable, la résolution du système linéaire décrit par (2.18) en vérifiant les équations (2.12) et (2.14) admet la solution représentée par la figure 2.9.

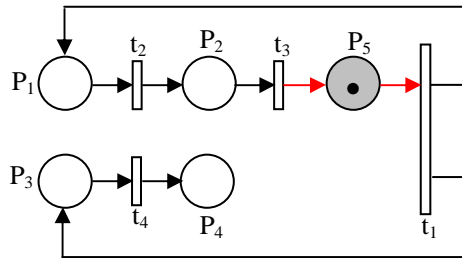
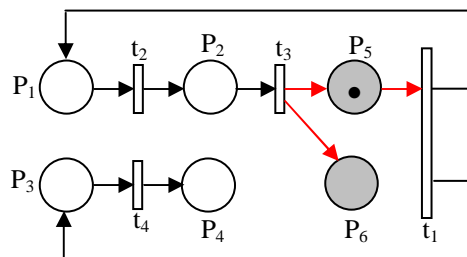


FIGURE 2.9 – Solution réalisable pour $\bar{n} = 4$ et une place non mesurable

L'analyse de ce dernier montre qu'il peut atteindre les états du système considéré mais il peut également accéder à des états complètement différents de ceux du système. Ainsi, ce modèle ne peut pas être retenu.

2. Pour un modèle identifié avec 4 transitions ($\bar{n} = 4$) et deux places non mesurables, la résolution du système linéaire décrit par (2.18) en vérifiant les équations (2.12), (2.13) et (2.14) admet la solution représentée par la figure 2.10.

L'analyse de ce dernier montre que ce modèle peut générer les comportements du système considéré mais, il peut aussi accéder à des états complètement différents de ceux du système. Donc, ce RdP ne peut pas être retenu. Notons finalement qu'aucune solution optimale n'a été trouvée en incrémentant le nombre de


 FIGURE 2.10 – Solution réalisable pour $\bar{n} = 4$ et deux places non mesurables

places et que la solution optimale a été obtenu en optant pour deux places non mesurables et 5 transitions ($\bar{n} = 5$). Cette solution représenté par la figure 2.8 est identique à celle donnée par Meda *et al.* [MRM98].

Nous venons de présenter les différents cas théoriques que l'on peut rencontrer lors de l'identification d'un SED en utilisant ces signaux de sortie. Des algorithmes permettant une identification du modèle ont été proposés pour chacun de ces cas. Cependant, l'application pratique de ces algorithmes nécessite parfois une connaissance même partielle sur le système. Ainsi, si l'on possède des informations sur la dimension du vecteur d'état du système, l'algorithme relatif au théorème 3 sera appliqué. En revanche, si l'on est devant un système mal connu (i.e., aucune connaissance sur les états) alors, l'algorithme relatif au théorème 4 qui sera utilisé.

2.4.5 Analyse de l'approche proposée

Dans cette partie nous proposons de discuter des propriétés de la méthode d'identification proposée en se basant sur la classification introduite dans le premier chapitre (cf § 1.3). Nous discutons également de la complexité numérique de l'algorithme.

En réalité, cette méthode permet d'obtenir un modèle interne d'une classe de SED en observant et en analysant ses signaux de sortie. Ces derniers sont récupérés à l'aide de capteurs équipant le système et sont enregistrés dans une séquence de sortie unique. Ainsi, notre méthode a l'avantage d'être *simple*.

Dans cette approche, le processus d'identification est effectué *hors ligne* en utilisant des informations sur le nombre de places mesurables (dimension du vecteur d'état mesurable du système). Ceci représente le nombre exact de places du RdP à identifier ou bien une borne inférieure sur ce nombre. Ce qui permet de conclure que notre méthode d'identification est *relative*.

Les signaux de sortie utilisés comme données d'identification sont analysés de fa-

çon *séquentielle* et l'algorithme d'identification consiste à sélectionner pour chaque cas une solution optimale selon une fonction objectif et des contraintes à respecter par le RdP à identifier. Donc, pour chaque cas un RdP unique est obtenu, ce qui donne l'avantage à la méthode d'identification d'être *constructive*.

L'évaluation des performances d'un algorithme consiste à calculer sa complexité en le comparant avec d'autres algorithmes réalisant les mêmes fonctionnalités. La complexité d'un algorithme est une fonction d'estimation du temps de calcul indépendamment de son implémentation (machine + langage), en fonction de la taille de données d'entrée et dans le pire cas. De façon plus formelle, la complexité d'un algorithme A est une fonction $C_A(N)$, donnant le nombre d'instructions caractéristiques exécutées par A dans le pire des cas, pour une donnée de taille N ([LPS03]). Notons que, les opérations de complexité 1 (complexité constante, indépendante de la taille de la donnée) sont des opérations non considérables devant les opérations de complexité n (dépendante de la taille de la donnée) pour le calcul de la complexité d'un algorithme. Les différents algorithmes ont des coûts différents en termes de temps d'exécution (nombre d'opérations effectuées par l'algorithme) et taille mémoire (taille nécessaire pour stocker les différentes structures de données pour l'exécution). Ces deux concepts sont appelés la complexité en temps et en espace de l'algorithme.

Pour évaluer la complexité du problème d'optimisation défini précédemment, il est facile de dire que dans le pire cas les inconnus sont $c = m(2.n + 1 + s)$ où s est le nombre de vecteurs de marquages dans la séquence \mathcal{M} , m et n représentent respectivement le nombre de places et de transitions. Donc, la complexité de ce problème est linéaire en fonction du nombre de places, de transitions et des vecteurs de marquages.

2.5 Proposition d'une approche d'identification à base de signaux d'E/S mesurables

Dans l'approche précédente, une classe de SED a été identifiée en utilisant les signaux de sortie du système acquis à l'aide des capteurs. L'utilisation de ces données rend, parfois, la convergence vers la solution optimale difficile plus particulièrement lorsque le nombre de paramètres à estimer est grand. Pour remédier à cet inconvénient une autre approche est proposée où les signaux fournis par les actionneurs sont aussi exploités. Cette approche apparente aux travaux utilisant les signaux d'entrée et de sortie du système pour établir son modèle [DFM06].

2.5.1 Définition du problème

Dans cette approche, les signaux d'entrée et de sortie mesurables associés aux signaux de capteurs et d'actionneurs sont exploités. Ces signaux sont enregistrés sous la forme d'une succession asynchrone d'états et d'événements dans une séquence d'entrée et de sortie (E/S) notée δ . L'opération d'enregistrement est effectuée à chaque observation d'au moins un événement ou à chaque changement d'état dans le temps. On peut représenter cette séquence par $\delta = E_0, e_i, E_i, e_j, \dots, e_k, E_l, E_h, e_m, E_p, \dots, e_q, E_s$ où $E_i \in \mathbb{N}^m$ est le vecteur d'état mesurable avec $i \in \{0, \dots, s\}$ et e_i représentent les événements mesurables du système avec $j \in \{0, \dots, q\}$. Donc, les états du système peuvent être partiellement observés par les signaux de sortie et les événements par les signaux d'entrée. Le problème d'identification à traiter dans cette partie est défini comme suit :

Définition 3 : Etant donnée S un SED à identifier et soit δ une séquence de vecteurs d'état et d'événements enregistrés en observant S . Le problème d'identification de ce SED consiste à déterminer l'ensemble de places P , l'ensemble de transitions T , le marquage initial non mesurable et la structure du RdP (les matrices $Post$ et Pre) qui permet de générer exactement le comportement de S décrit par δ .

Pour résoudre ce problème, nous proposons d'associer les vecteurs d'état mesurables E_i atteignables par S aux vecteurs de marquages M_{α_i} atteignables par le RdP à établir et les événements mesurables e_i aux transitions t_{α_i} . Le vecteur de marquage, dans ce cas, sera composé de deux parties $M = \begin{pmatrix} M_m \\ M_{\bar{m}} \end{pmatrix}$, où la première partie représente le marquage des places mesurables et la deuxième le marquage estimé des places non mesurables. Ainsi, l'ensemble des transitions du RdP à identifier est composé de deux sous ensembles, une représente les transitions associées aux événements mesurables et l'autre les transitions associées aux événements non mesurables.

2.5.2 Résolution du problème d'identification

Avant de présenter l'approche de résolution du problème d'identification donné par la définition 3, il est à noter que nous n'étudierons pas cette fois-ci les conditions nécessaires et suffisantes pour l'unicité de la solution. Ceci est dû au fait que l'existence de cette dernière exige la mesurabilité de tout le vecteur de marquage (i.e., ce qui revient aux conditions données dans le théorème 1 sauf que cette fois-ci la séquence de transitions complète peut être déterminée en utilisant les signaux issus des actionneurs).

Pour résoudre le problème d'identification défini par la définition 3, nous proposons de procéder comme suit :

1. Déterminer la partie mesurable (places et transitions mesurables) du RdP à identifier.
2. Estimer la partie non mesurables (places et transitions non mesurables) du RdP à élaborer
3. Calculer la structure et le marquage initial non mesurable du RdP à identifier en exploitant la technique de la PLB.

Ces étapes seront détaillées dans les sous-sections suivantes.

2.5.2.1 Détermination de la partie mesurable

En analysant la séquence des vecteurs d'état et d'événements décrite dans δ , nous pouvons déduire le nombre de places et de transitions mesurables qui constituent le modèle RdP. En effet, le nombre de places mesurables est égal à la dimension du vecteur d'état mesurable E_i et représente une borne inférieure sur le nombre total des places du RdP à établir. Quant au nombre de transitions mesurables, il est inférieur ou égal au nombre d'événements mesurables dans la séquence δ (chaque transition t_{α_i} du RdP à établir est associée à au moins un événement e_i produit par le système). Enfin, la partie mesurable du marquage initial M_{m_0} est connue et est égal au vecteur d'état mesurable E_0 . Ainsi, l'algorithme d'identification nécessite, dans ce cas, la détermination de la partie non mesurable du RdP à identifier.

2.5.2.2 Estimation de la partie non mesurable

La partie non mesurable est composée de deux types de noeuds : places et transitions non mesurables. La détermination de ces derniers peut être effectuée en incrémentant le nombre de places et de transitions non mesurables comme précédemment (cf. §2.4.4.2). Cependant, comme on dispose cette fois-ci des mesures issues des actionneurs, il serait plus pertinent de les exploiter afin d'estimer la partie non mesurable, ce qui permettrait de minimiser le temps de calcul.

La détermination du nombre de transitions non mesurables est obtenue en analysant la séquence δ car tous les événements générés par le système sont supposés être détectables. Ceci permet de déterminer le nombre exact de transitions de la partie non mesurable. Afin de déterminer ce nombre nous proposons de procéder comme suit :

Début :

1. Initialiser le nombre de transitions non mesurable à 0 ($n_{\bar{o}} = 0$).
2. Créer un ensemble vide Θ ($\Theta = \{\emptyset\}$).
3. Parcourir la séquence δ et faire :

- (a) Si le système passe de E_i à E_j ($E_i \neq E_j$) sans l'occurrence d'événements mesurables alors :
- i. Calculer $E_{ij} = E_j - E_i$.
 - ii. Si $E_{ij} \notin \Theta$ alors :
 - Incrémenter le nombre de transitions non mesurables ($n_{\bar{o}} = n_{\bar{o}} + 1$) et aller à (b).
 - iii. Sinon,
 - (b) - Passer aux états suivants dans δ et revenir à (a).
 - Fin Si.
- Fin Si.

Fin.

Par ailleurs, le nombre de places non mesurables sera inférieur à une borne supérieure donnée $\bar{m}_{\bar{o}}$. En effet, le nombre de places non mesurables sera initialisé à 1 et incrémenté progressivement jusqu'à ce que l'on trouve une solution optimale.

Remarque 3 : Il est important de noter que, le nombre de places non mesurables peut être aussi estimé en analysant la séquence δ . Cependant, cette estimation permet uniquement d'avoir une borne supérieure sur ce nombre. En effet, comme notre objectif est de trouver un RdP ayant une structure minimale et générant tous les comportements observables du système, nous pouvons considérer que l'occurrence de chaque événement provoque au maximum un changement au niveau de la partie non mesurable du vecteur d'état. Ainsi, le nombre de places non mesurable peut être estimé en appliquant la procédure suivante :

Début :

1. Initialiser le nombre de places non mesurable à 0 ($m_{\bar{o}} = 0$).
 2. Créer un ensemble vide Λ ($\Lambda = \{\emptyset\}$).
 3. Parcourir la séquence δ et faire :
 - (a) Si le système reste dans l'état E_j malgré l'occurrence d'un événement mesurable e_i (i.e., $E_j e_i E_j$) alors :
 - i. Si $E_j \notin \Lambda$ alors :
 - Incrémenter le nombre de places non mesurables ($m_{\bar{o}} = m_{\bar{o}} + 1$) et aller à (b).
 - ii. Sinon,
 - (b)- Passer aux états suivants dans δ et revenir à (a).
 - Fin Si.
- Fin Si.

Fin.

Après avoir estimé la partie non mesurable, nous devons trouver la structure du RdP qui modélise le système et estimer la partie non mesurable de son marquage initial $M_{\bar{m}_0}$. Pour cette raison, nous définissons ce problème sous la forme d'un problème de PLB que nous résolvons en utilisant les données suivantes : m_o (n_o) le nombre de places (transitions) mesurables, $m_{\bar{o}}$ ($n_{\bar{o}}$) le nombre de places (transitions) non mesurables, M_{m_0} le marquage initial mesurable et la séquence d'états atteignables par le système à identifier décrite dans δ .

2.5.2.3 Calcul de la structure de RdP

Le calcul de la structure du RdP qui modélise le système considéré se fait en définissant le problème d'identification sous la forme d'un problème d'optimisation et en le résolvant en utilisant la PLB. Comme tout problème d'optimisation, le problème de PLB est défini par une fonction objectif et un ensemble de contraintes linéaires. Les contraintes, dans cette approche, sont formulées en utilisant les vecteurs de marquages mesurables associés aux vecteurs d'état atteignables par le système à identifier et les transitions mesurables (non mesurables) associées aux événements mesurables (non mesurables). Ainsi, la fonction objectif est définie en utilisant les éléments des matrices Pre et $Post$ et ceux du marquage initial comme suit :

$$f(Pre, Post, M_0) = \sum_{i=1}^m \sum_{j=1}^n (a_{i,j} \cdot Pre(p_i, t_j) + b_{i,j} \cdot Post(p_i, t_j)) + \sum_{i=1}^m c_i \cdot M_0(p_i) \quad (2.19)$$

où $a_{i,j}$, $b_{i,j}$, c_i sont des coefficients de pondération à choisir par l'utilisateur.

Ce choix est motivé par le fait que le RdP à établir doit être minimal en nombre d'arcs et de marques dans le marquage initial. Enfin, en donnant la même ordre d'importance à ces coefficients, la fonction objectif (2.19) peut être réécrite comme suit :

$$f(Pre, Post, M_0) = \vec{1}_{1 \times m} \cdot (Pre_{m \times n} + Post_{m \times n}) \cdot \vec{1}_{n \times 1} + \vec{1}_{1 \times m} \cdot M_0 \quad (2.20)$$

Théorème 5 : Une solution du problème d'identification introduit dans la définition 4 et qui minimise la fonction objectif f décrite par l'équation (2.20) est obtenue en résolvant le système linéaire suivant :

$$\min f(Pre, Post, M_0) \quad (2.21)$$

en respectant les contraintes suivantes :

$\zeta(\delta, m, n) =$

$$\left\{ \begin{array}{ll} \forall t_i \in T, \forall p \in P : Pre(p, t_i) \in \{0, 1\} \text{ et } Post(p, t_i) \in \{0, 1\} & (a) \\ Pre_{m \times n}^T \times \vec{1}_{m \times 1} + Post_{m \times n}^T \times \vec{1}_{m \times 1} \geq \vec{1}_{n \times 1} & (b) \\ Pre_{m \times n} \times \vec{1}_{n \times 1} + Post_{m \times n} \times \vec{1}_{n \times 1} \geq \vec{1}_{m \times 1} & (c) \\ Post_{m \times n} + Pre_{m \times n} \leq \vec{1}_{m \times n} & (d) \\ \forall M_{\alpha_i} \in \mathcal{M}', \forall t_i \in \sigma' : Post(\cdot, t_i) - Pre(\cdot, t_i) = M_{\alpha_i} - M_{\alpha_{i-1}} & (e) \\ \forall t_i \in T, \forall p \in P : Pre(\cdot, t_i) \leq M_{\alpha_{i-1}} & (f) \\ M_0 \in \mathbb{N}^m \text{ et } \forall M_{\alpha_i} \in \mathcal{M}' : M_{\alpha_i} \in \mathbb{N}^m & (j) \\ M_0 - Pre(\cdot, t_i) \geq \vec{0}_{m \times 1} & (h) \\ Pre(\cdot, t_i) \leq M_0 & (i) \end{array} \right. \quad (2.22)$$

Ces contraintes représentent généralement des propriétés structurelles qui doivent être satisfaites par le modèle RdP à identifier. Elles imposent au RdP d'être connexe, ses marquages atteignables doivent satisfaire l'équation de marquage et les conditions de franchissement d'une transition. D'autres contraintes peuvent être ajoutées au système défini par les équations (2.20) et (2.22) si d'autres propriétés doivent être satisfaites par le modèle RdP à établir. Par exemple, si ce dernier ne doit pas contenir des places et/ou des transitions puits (sources), les contraintes décrites dans les équations (2.11), (2.12), (2.13) et (2.14) doivent être ajoutées à celles de l'équation (2.22).

Il est important de noter qu'à partir des données d'identification on peut déduire des P-invariants et des T-invariants, s'ils existent, de la partie mesurable. Ils peuvent être formulés sous forme des contraintes qui doivent être vérifiées par le modèle RdP à identifier.

Utilisation des P-invariants et T-invariants de la partie mesurable :

Dans cette méthode nous pouvons exploiter les différentes informations déduites à partir des vecteurs d'états et d'événements mesurables enregistrés dans δ . D'une part, en analysant les vecteurs de marquages mesurables associés aux vecteurs d'état, les P-invariants des places mesurables peuvent être déduits. En réalité, si nous remarquons que la somme d'un sous ensemble d'éléments du vecteur d'état reste inchangé dans tous les vecteurs d'états atteignables par le système à identifier alors, nous pouvons associer à ces éléments, la somme pondérée des marquages d'un sous-ensemble des places mesurables dans les différents vecteurs de marquages associés à ces vecteurs d'état. Ce sous-ensemble peut former un P-invariant du RdP. Ce dernier noté X et décrit par l'équation (2.23) peut être ajouté au système linéaire défini précédemment afin qu'il soit satisfait par le RdP à identifier.

$$X^T \cdot (Post - Pre) = 0 \quad (2.23)$$

où X est un vecteur d'entiers dont les composants représentent les marques associés aux places du RdP.

D'autre part, à partir de la séquence de transitions détectée depuis la séquence d'E/S et ses marquages équivalents, on peut déduire les T-invariants du RdP à identifier. En effet, la séquence de transitions qui permet de revenir à l'état initial est une séquence répétitive. Le vecteur de franchissement Y associé à cette séquence est considéré comme un T-invariant du RdP à identifier. Ce dernier noté Y et décrit par l'équation (2.24) peut être ajouté au système linéaire défini précédemment afin qu'il soit satisfait par le RdP à identifier.

$$(Post - Pre).Y = 0 \quad (2.24)$$

Finalement, nous pouvons résumer notre algorithme d'identification comme décrit dans l'algorithme 2.

Exemple 8 : Afin d'illustrer l'algorithme d'identification précédent considérant l'exem-

$$\begin{aligned} \text{ple d'un SED à identifier dont la séquence d'E/S est : } \delta = & \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix} e_1 \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} e_2 \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} e_3 \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} \\ & \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix} e_1 \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} e_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} e_2 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} e_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} e_2 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} e_3 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} e_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} e_2 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} e_3 \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ & \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix} e_1 \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} e_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} e_2 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} e_1 \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} e_3 \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} e_2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} e_3 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ & e_2 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} e_3 \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix} \begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix} . \end{aligned}$$

En analysant δ , nous pouvons conclure que le nombre de places mesurables m_o dans le modèle RdP à identifier est égal à 3. Les marquages initiaux de ces places, notées p_1 , p_2 et p_3 , sont respectivement $M(p_1) = 0$, $M(p_2) = 3$ et $M(p_3) = 3$. Cette séquence permet également de conclure que le RdP contient 3 transitions mesurables ($n_o = 3$). Ces transitions notées t_1 , t_2 et t_3 correspondent respectivement aux événements e_1 , e_2 et e_3 .

En analysant les données d'identification enregistrées dans δ , nous remarquons que le système change son état de $\begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}$ à $\begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix}$ sans l'occurrence d'événements mesurables. Ceci permet de conclure que ce changement d'état est dû à l'occurrence d'un événement non mesurable que l'on note e_4 et auquel on associe la transition t_4 . De la

Algorithme 2 : Algorithme d'identification

Données : δ , les données d'identification.

Début

1 Détermination de la partie mesurable :

- 1.1. Transformer la séquence d'E/S à une séquence de marquages en associant à chaque vecteur d'état mesurable E_i un vecteur de marquage mesurable M_{α_i} et à chaque événement mesurable e_i une transition mesurable t_{α_i} .
- 1.2. Déterminer P_m l'ensemble de places mesurables de cardinal m_o et T_m l'ensemble de transitions mesurable de cardinal n_o (cf. §2.5.2.1).

2 Estimation de la partie non mesurable :

- 2.1. Identifier $T_{\bar{m}}$ l'ensemble de transitions non mesurables de cardinal $n_{\bar{o}}$ qui doivent être ajoutées à la partie mesurable (cf. §2.5.2.2).
- 2.2. Fixer une borne supérieure $\bar{m}_{\bar{o}}$ sur le nombre de places non mesurables $P_{\bar{m}}$ (cf. §2.5.2.2).

3 Calcul de la structure du RdP à identifier et le marquage initial de la partie non mesurable :

- 3.1. Initialiser le nombre de places non mesurables à 1.
- 3.2. Tant que le nombre de places non mesurables est inférieur ou égale à $\bar{m}_{\bar{o}}$ faire :
 - 3.2.1. Utiliser $m_o, n_o, m_{\bar{o}}, n_{\bar{o}}$ et M_{m_0} pour calculer les éléments des matrices Pre et $Post$ et le marquage initial des places non mesurables $M_{\bar{m}_0}$ en définissant et en résolvant un problème de PLB. Ce dernier est caractérisé par une fonction objectif représentée par l'équation (2.20) et un ensemble de contraintes décrites par les équations ((2.22), (2.11),(2.12),(2.13) et (2.14)).
 - 3.2.3. Si aucune solution optimale n'est trouvée alors :
 - Incrémente le nombre de places non mesurables et revenir à l'étape 3.2.
- Fin Si.
- Fin Tant que.

Fin

même façon, il est facile de remarquer que l'occurrence des événements e_5, e_6 et e_7 que l'on représente par la transition t_4 . En effet, l'analyse des vecteurs d'état mesurables en amont et en aval de ces événements donne la même information que celle reliée à e_4 . Par conséquent, le nombre de transitions non mesurables $n_{\bar{o}}$ dans le RdP à identifier est égal à 1.

Par ailleurs, le nombre de places non mesurables est initialisé à 1. En utilisant m_o , n_o , $m_{\bar{o}}$ et $n_{\bar{o}}$ on doit déterminer la structure du modèle RdP et le marquage initial des places non mesurables. Pour cela, on doit résoudre le problème de PLB défini par la fonction objectif décrite par l'équation (2.20) sous les contraintes (2.22), (2.11), et (2.14). Ainsi, on remarque que la séquence de transitions t_1, t_2, t_3, t_4 est une séquence répétitive alors, on peut utiliser le T-invariant associé à cette séquence comme une information supplémentaire pour mieux caractériser le RdP à identifier. Pour formuler toutes ces contraintes, on utilise les vecteurs d'états de la séquence δ , le vecteur d'état initial $\begin{pmatrix} 0 \\ 3 \\ 3 \end{pmatrix}$, $n = n_o + n_{\bar{o}}$ le nombre de transitions, $m = m_o + m_{\bar{o}}$ le nombre estimé des places pour résoudre ce problème.

Il est important de noter que, pour $m_{\bar{o}} = 1$, le problème d'identification n'admet aucune solution faisable. Cependant, pour $m_{\bar{o}} = 2$, le problème d'identification admet une solution réalisable optimale représentée dans la figure 2.11. Le modèle obtenu dans ce cas peut générer tous les comportements du système considéré.

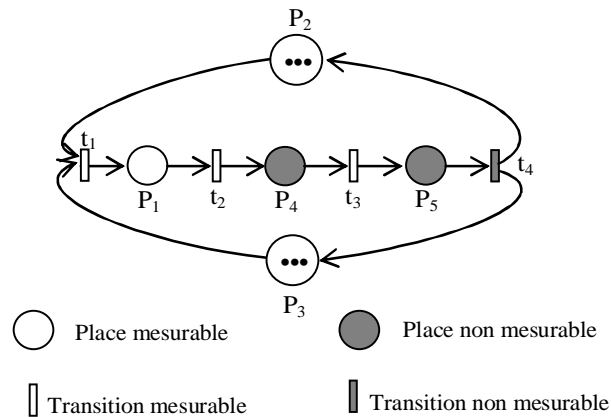


FIGURE 2.11 – Le modèle identifié optimal

2.5.3 Analyse de l'approche proposée

Dans cette partie nous discutons les propriétés de la méthode d'identification et du modèle identifié en se basant sur la classification introduite dans le premier chapitre (cf. § 1.3). Nous évaluons également la complexité de l'algorithme.

Cette méthode d'identification permet d'établir un modèle comportemental d'une classe de SED en observant et en exploitant ses signaux d'entrée et de sortie mesurables. Ces derniers sont récupérés à l'aide des capteurs et d'actionneurs équipant le système et ils sont enregistrés dans une séquence d'E/S unique. Ainsi, cette méthode a l'avantage d'être *simple*.

Dans cette méthode, le processus d'identification est effectué en utilisant des informations sur le nombre de places mesurables (partie mesurable du vecteur d'état). Ceci représente en réalité une borne inférieure sur le nombre de places du modèle RdP à identifier. Ce qui permet de conclure que la méthode d'identification proposée est *relative*.

Les données d'identification représentées par les signaux d'entrée et de sortie sont analysées de façon *séquentielle* et l'algorithme d'identification permet de sélectionner une solution faisable optimale si elle existe. Ceci permet de conclure que la méthode d'identification est *constructive*.

L'étude de la complétude du modèle identifié est faite en prenant en compte que les comportements positifs de ce système. Notons que, ces comportements sont ceux enregistrés dans la séquence d'E/S δ . Afin de prouver que le modèle identifié est complet, on doit démontrer que ce dernier doit générer la séquence d'E/S δ . En effet, on doit affirmer qu'il existe au moins un chemin de marquages dans le graphe de marquages du modèle RdP identifié qui représente δ . Ceci est motivé par le fait que nous avons associé les vecteurs d'état atteignables par le système à identifier aux vecteurs de marquages supposés atteignables par le RdP qui le modélise et les événements aux transitions. Pour ce faire, nous introduisons les définitions suivantes :

Définition 4 : Un graphe d'accessibilité d'un RdP noté R_G est un graphe composé de noeuds qui correspondent aux marquages atteignables par ce RdP et d'arcs aux transitions franchies qui permet de passer d'un marquage à un autre.

Définition 5 : Un chemin de marquage $\pi_i \in R_G$ est un chemin orienté composé d'une succession de marquages et de transitions qu'on peut représenter comme suit :

$$\pi_i = M_0, t_{\alpha_1}, \dots, M_{\alpha_i}, t_{\alpha_i}, M_{\alpha_j}, \dots, M_{\alpha_k}.$$

Comme il a été indiqué précédemment, le vecteur de marquage du modèle identifié est composé de deux parties : une mesurable M_m et l'autre non mesurable $M_{\bar{m}}$. Pour chaque marquage $M_{\alpha_i} \in R_G$, sa partie mesurable $M_{m\alpha_i}$ est associée au vecteur d'état mesurable E_i dans la séquence δ . On peut conclure que : $\forall M_{\alpha_i} \in R_G, \exists E_i \in \delta | M_{m\alpha_i} = E_i$ avec $M_{m\alpha_i}, E_i \in \mathbb{N}^m$.

Dans R_G , le passage d'un marquage M_{α_i} à un marquage M_{α_j} est fait par un arc unique

qui porte l'étiquette d'une transition unique t_{α_i} . Dans cette méthode, nous avons supposé qu'une transition t_{α_i} est équivalent à au moins un événement e_i qui peut être mesurable ou non mesurable et qui provoque le changement d'état du système de E_i à E_j avec $M_{m\alpha_i} = E_i$ et $M_{m\alpha_j} = E_j$. Pour cette raison on peut noter que :

$$\begin{aligned} &\text{Pour } M_{m\alpha_i}, M_{m\alpha_j} \in R_G \text{ avec } (M_{m\alpha_i} \xrightarrow{t_{\alpha_i}} M_{m\alpha_j}) \text{ et Si :} \\ &1. \forall M_{\alpha_i} \in R_G, \exists E_i \in \delta | M_{m\alpha_i} = E_i \\ &2. \forall M_{\alpha_j} \in R_G, \exists E_j \in \delta | M_{m\alpha_j} = E_j \text{ alors} \\ &\quad \exists e_i \text{ équivalent à } t_{\alpha_i} | E_i e_i E_j \in \delta. \end{aligned}$$

Cette condition doit être satisfaite pour tous deux marquages successifs M_{α_i} et $M_{\alpha_j} \in R_G$. En satisfaisant cette condition, on peut trouver un chemin de marquages $\pi_i \in R_G$ équivalent à la séquence d'E/S δ . Dans ce cas, on peut conclure que le modèle identifié peut générer la séquence d'E/S δ , ce qui implique qu'il est *complet*.

Pour évaluer la complexité de l'algorithme d'identification proposé précédemment, nous considérons s le nombre de vecteurs d'état mesurables, m_o la dimension du vecteur d'état mesurable et q le nombre d'événements mesurables dans la séquence d'E/S δ . Pour identifier la partie mesurable et estimer celle non mesurable, dans le pire cas, nous devons parcourir tous les vecteurs d'état élément par élément et la séquence d'événements correspondante. Nous effectuons, dans le pire cas, $(s.m_o + q)$ opérations. De plus, pour déterminer la structure du RdP à identifier et le marquage initial des places non mesurables, les inconnus sont $m(2n + s + 1)$. Enfin, on peut dire que la complexité de cet algorithme est linéaire en fonction du nombre de places, de transitions et du nombre de vecteurs d'état atteignables par le SED à identifier.

2.6 Conclusion

Deux méthodes d'identification d'un SED par RdP ont été proposées dans ce chapitre. Ces méthodes font partie de celles qui définissent le problème d'identification sous la forme d'un problème d'optimisation équivalent.

La première méthode est basée sur l'analyse des signaux de sortie du système. Dans cette méthode, nous avons commencé par étudier les conditions nécessaires et suffisantes d'existence d'une solution unique de ce problème. Par ailleurs, lorsque l'unicité de la solution n'est pas garantie, nous avons vu comment le problème d'identification est défini sous la forme d'un problème d'optimisation discrète et résolu en utilisant la PLB. Nous avons fini par dégager les propriétés de cette méthode où nous avons conclu qu'elle a l'avantage d'être simple, relative, séquentielle et constructive.

Cependant, l'utilisation des signaux de sortie rend, parfois, la convergence vers la solution optimale difficile surtout dans les cas des systèmes de grandes tailles et avec un grand nombre de variables d'état à estimer. Pour remédier à cet inconvénient, nous avons proposé une deuxième méthode d'identification permettant d'exploiter les signaux d'entrée et de sortie mesurables du système à identifier. Cette méthode consiste, tout d'abord, à identifier la partie mesurable et à estimer la partie non mesurable en analysant les vecteurs d'état et d'événements mesurables enregistrés dans la séquence δ . Puis, calculer la structure d'un RdP qui modélise le système considéré et le marquage initial des places non mesurables en définissons et en résolvant un problème de PLB. La méthode proposée permet d'élaborer un modèle RdP complet et unique, elle a l'avantage d'être relative, séquentielle et constructive.

L'utilisation des signaux d'entrée et de sortie mesurables et le bon choix des contraintes permet d'accélérer et de garantir la convergence vers un RdP optimal. Comme il a été prouvé précédemment, ce modèle permet d'atteindre tous les états du système réel. Néanmoins, il peut parfois accéder à des états qui n'ont pas d'équivalents dans l'ensemble d'états du système. Ces derniers sont considérés comme des états interdits (illégaux) dont on doit empêcher le RdP identifié à atteindre.

Pour résoudre ce problème, des algorithmes de synthèse d'un contrôleur RdP sont développés. Ils permettent d'ajouter des places de contrôle au modèle RdP identifié pour l'empêcher à accéder à des états interdits et garantir l'atteignabilité des états désirables seulement. Ces algorithmes seront présentés en détail dans le chapitre suivant.

Chapitre 3

Synthèse d'un contrôleur RdP pour résoudre le problème d'états interdits

Résumé : Dans ce chapitre, nous nous intéressons au problème que l'on a rencontré au chapitre 2 et qui concerne les sur-comportements qui peuvent être parfois générés par le RdP identifié. La première partie de ce chapitre est consacrée à une étude bibliographique portant sur ce problème. Dans la deuxième partie, nous proposons deux approches de résolution de ce problème. La première approche est basée sur l'utilisation des chemins d'influence des transitions interdites. Quant à la deuxième, elle consiste à utiliser les séquences de transitions interdites. Ces approches consistent à ajouter, au modèle RdP issu de la phase d'identification, des places de contrôle permettant d'éviter l'atteignabilité des états non désirés. Nous finirons ce chapitre par des conclusions. Les résultats de ce chapitre ont été publiés dans : [BME⁺08a], [BME⁺08b], [BME⁺08c].

3.1 Introduction

L'application des méthodes d'identification proposées dans le chapitre 2 permet d'établir un modèle complet capable de générer tous les comportements du système réel. Cependant, ce modèle peut générer parfois des états, relatifs à des comportements indésirables, qui n'ont pas d'équivalents dans l'ensemble des états du système réel. Ces derniers sont considérés comme des états interdits et seront ainsi traités dans le cadre des problèmes d'états interdits.

La complexité des systèmes à contrôler rend les méthodes intuitives incapables de garantir une supervision efficace. Ce besoin a donné naissance à la théorie de la supervision initiée par Ramadge et Wonham [RW87a], [RW87b], [RW89]. L'objectif est de construire le contrôleur le plus permissif qui respecte les spécifications de l'utilisateur. Dans cette théorie, le système ainsi que l'ensemble des comportements valides sont donnés chacun par un langage. A partir de ces deux langages, le but de la synthèse consiste alors à déterminer un sous-ensemble des comportements du système qui appartiennent aussi à la spécification. Ce sous-ensemble caractérise le contrôleur et représente les comportements du système subissant son action.

Cependant, le problème principal de la théorie de supervision est l'existence d'événements incontrôlables. Lors de la synchronisation d'un événement incontrôlable du modèle du procédé avec celui du modèle de la spécification, le modèle contrôlé ne peut pas respecter cette synchronisation. Dans ce cas, un ensemble d'états, appelé *états interdits*, apparaît. Des algorithmes à base d'automates ont été proposés pour résoudre ce problème [RW89][KG95]. Néanmoins, le contrôleur est donné sous forme d'un automate dont la taille le rend inexploitable. Des méthodes efficaces à base de RdP ont été, aussi, proposées pour résoudre le problème d'états interdits [YML⁺96], [GRX03b], [BPG07]. L'objectif de ces méthodes est de synthétiser un contrôleur maximum permissif permettant d'inhiber l'atteignabilité des états interdits et de garantir les comportements désirés.

Au début de ce chapitre, une étude bibliographique concernant le problème d'états interdits est présentée. Ceci permet de positionner notre travail par rapport aux travaux existant et montrer que ces derniers ne peuvent pas être exploités dans notre cas. Par la suite, deux approches de synthèse d'un contrôleur RdP sont proposées. La première approche propose de caractériser les places de contrôle à ajouter au modèle RdP identifié en utilisant les chemins d'influence des transitions interdites. Quant à la deuxième approche, elle consiste à exploiter les séquences de transitions interdites.

3.2 Réseaux de Petri et contrôle

Généralement, dans un problème de commande supervisée, les spécifications s'expriment sous la forme d'un comportement désiré du système. Elles peuvent être, également, définies par un ensemble d'états interdits ou par un ensemble de transitions d'état interdites. Les spécifications exprimées par un ensemble d'états interdits représentent un problème d'états interdits. Quant à celles exprimées par un ensemble de transitions d'état interdites, elles représentent un problème de transitions d'état interdites. Dans ce qui suit, nous définissons ces deux catégories de problèmes, puis, nous exposons quelques approches existantes pour la résolution du problème d'états interdits afin de situer nos approches.

3.2.1 Problème de transitions d'état interdites

Pour illustrer le problème de transitions d'état interdites (PTEI), considérant l'exemple d'un procédé dont le modèle RdP est représenté par la figure 3.1(a).

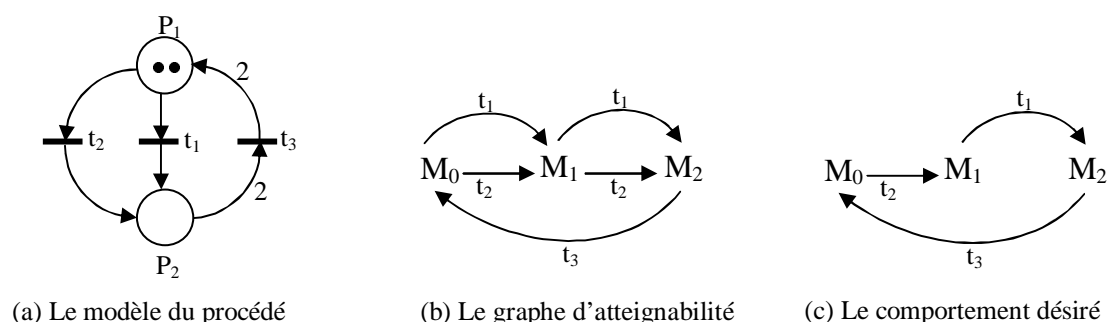


FIGURE 3.1 – Problème de transitions d'état interdites

Si l'on considère que les séquences légales sont de la forme $(t_2 t_1 t_3)^*$, comme montré sur la figure 3.1(c), alors les comportements interdits obtenus à partir du graphe de marquages représenté par la figure 3.1(b) sont $(M_0 \xrightarrow{t_1} M_1)$ et $(M_1 \xrightarrow{t_2} M_2)$. M_1 et M_2 restent des marquages admissibles et les transitions t_1 et t_2 peuvent toujours être franchies dans le comportement souhaité mais seulement sous des conditions particulières. D'une manière formelle, un problème de transitions d'état interdites est défini comme suit :

Définition 1 [Lee06] : Un problème de transitions d'état interdites (PTEI) est un problème de commande supervisée à base de RdP. Le comportement désiré du modèle du procédé (N_p, M_0) doit satisfaire les conditions suivantes :

- (a) $M_0 \in CD$ (CD : Comportement Désiré)
- (b) $\forall M \in CD, (M_0 \xrightarrow{\sigma} M) \in CD$

avec σ : est une séquence de transitions franchies.

La première condition, traduisant l'admissibilité du marquage initial, est nécessaire pour l'existence d'un contrôleur. La deuxième condition impose l'atteignabilité de tout marquage admissible à partir de M_0 . Notons que, dans un PTEI, on peut avoir des transitions entre états admissibles qui n'appartiennent pas nécessairement au comportement désiré.

3.2.2 Problème d'états interdits

Le problème d'états interdits est l'un des problèmes les plus traités dans le cadre de la synthèse d'un contrôleur RdP. Dans ce contexte, le concept d'états interdits consiste à supprimer certains états du graphe d'atteignabilité du RdP afin d'obtenir un graphe d'atteignabilité correspondant uniquement aux comportements tolérés. Pour expliquer la notion du problème d'états interdits (PEI), considérant l'exemple d'un système en boucle ouverte dont le comportement est décrit par la figure 3.2(a).

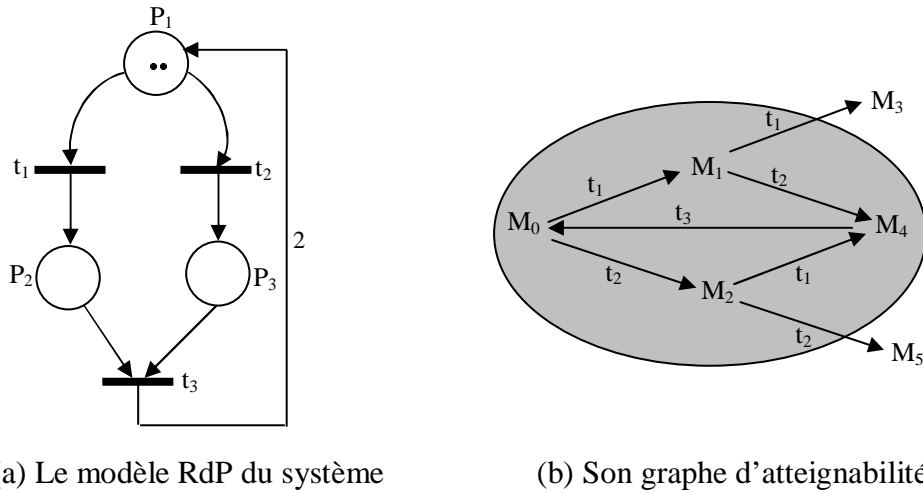


FIGURE 3.2 – Problème d'états interdits

Le graphe d'atteignabilité correspondant à ce modèle (voir la figure 3.2(b)) montre bien que ce système peut atteindre deux états de blocage qui sont $M_3 = [0 \ 2 \ 0]^T$ et $M_5 = [0 \ 0 \ 2]^T$. Ces derniers sont considérés comme des états qu'il faut interdire en synthétisant un contrôleur RdP.

Définition 2 [Lee06] : Un problème d'états interdits (PEI) est un problème de commande supervisée à base de RdP. Le comportement désiré de modèle du procédé (N_p, M_0) doit satisfaire les conditions suivantes :

- (a) $M_0 \in CD$ (CD : Comportement Désiré)
- (b) $\forall M \in CD, (M_0 \xrightarrow{\sigma} M) \in CD$
- (c) $\forall (M \xrightarrow{t} M') \in R(N_p, M_0), M, M' \in CD \Rightarrow (M_0 \xrightarrow{t} M') \in CD$

où σ est une séquence de transitions franchies depuis le marquage initial et $R(N_p, M_0)$ est le graphe d'atteignabilité.

Il est à noter qu'un problème d'états interdits est bien un problème de transitions d'état interdites mais l'inverse n'est pas forcément vrai. En effet, dans un PEI toute transition d'état interdite conduit nécessairement à un état interdit.

3.3 Approches de synthèse d'un contrôleur RdP

De nombreux travaux ont été proposés pour résoudre ces problèmes et plus particulièrement le PEI. Holloway et Krogh [HK90], ont proposé une approche pour une classe de RdP en exploitant certaines propriétés structurelles. Cette approche utilise les prédicats linéaires définis sur les modèles de type graphe d'événements saut et vivant. Une approche similaire valable pour des modèles de type graphe d'états a été développée par Boel *et al.* [BBB95]. La généralisation de ces approches aux classes de graphes d'événements et graphes d'états a été introduite par Holloway *et al.* [HGZ96]. Giua s'intéressait principalement aux spécifications de contrôle sous la forme d'inégalités linéaires sur les marquages ou bien des Contraintes Généralisées d'Exclusion Mutuelles (CGEM) [GDS92]. D'autres approches basées sur la théorie des régions ont été aussi proposées pour résoudre ces problèmes. Ainsi, dans [GRX01a], [GRX01b], [GRX01c] [GRX02a], [GRX03a], [GRX03b], [GRX02b] [GRX02c], Ghaffari *et al.* ont adapté cette théorie pour synthétiser un contrôleur RdP, quant à Achour *et al.*, ils ont utilisé cette théorie pour résoudre le PEI caractérisé par un ensemble de CGEM sous observation partielle d'événements [ARX04] [ARX05]. Ils ont également exploité cette théorie pour synthétiser un contrôleur RdP satisfaisant des contraintes temporelles [AR05][AR07]. Dans le même contexte, Lee *et al.* ont proposé dans [LTD06][Lee06] une nouvelle méthode de synthèse de la commande basée sur le formalisme RdP d'un système manufacturier reconfigurable.

On peut classer les travaux qui s'intéressent au PEI en deux grandes branches :

- (1) Les approches utilisant les contraintes linéaires. Parmi ces approches, qui synthétisent le contrôleur en exprimant les spécifications sous la forme de contraintes linéaires, on peut citer : l'approche de Giua *et al.* [GDS92], celle de Yamalidou et

Moody [YML⁺96] et celle de Basile *et al.*[BPG07].

(2) les approches utilisant la théorie des régions. Parmi ces approches, qui synthétise le contrôleur à partir d'un automate à états finis, on trouve celle de Ghaffari *et al.*[Gha02], de Achour *et al.*[ARX04][ARX05] et de Lee *et al.*[Lee06].

Dans ce qui suit, nous présentons ces approches avec plus de détails afin de les situer dans le cadre de notre travail.

3.3.1 Approches utilisant les contraintes linéaires

Les spécifications de contrôle, dans plusieurs cas, peuvent être représentées par des contraintes linéaires ou *CGEMs*. Une *CGEM* est une condition qui limite la somme pondérée de marques dans un ensemble de places. Parmi les méthodes de synthèse d'un contrôleur RdP utilisant les *CGEMs*, on peut citer celle proposée par Giua *et al.* [GDS92]. Cette dernière a été adaptée par Yamalidou et Moody [YML⁺96] pour construire un contrôleur en boucle fermée pour un système modélisé par RdP en utilisant les invariants de places. Dans cette approche, les spécifications sont représentées par des contraintes linéaires en utilisant les marquages des places du RdP.

3.3.1.1 Approche de Giua *et al.*

Dans cette approche [GDS93], les auteurs se sont intéressés au problème de la synthèse d'un contrôleur RdP qui satisfait des spécifications de contrôle de type *CGEM*. Ces contraintes sont décrites par des inégalités linéaires sur le marquage de certaines places pour définir l'ensemble des états admissibles. Chaque *CGEM* est définie par une paire (\vec{c}, k) telle que : \vec{c} est un vecteur d'entiers et k est une constante entière. Les auteurs ont défini l'ensemble des états admissibles par :

$$\mathcal{M}_{(\vec{c}, k)} = \{M \in \mathcal{M} \mid \vec{c}^T M \leq k\} \quad (3.1)$$

où l'ensemble des états admissibles décrit par un ensemble de *CGEM* (C, \vec{k}) avec $C = [\vec{c}_1, \dots, \vec{c}_r]$ et $\vec{k} = (k_1, \dots, k_r)$ est donné par :

$$\mathcal{M}_{C, \vec{k}} = \bigcap_{i=1}^r \mathcal{M}_{(\vec{c}_i, k_i)}.$$

Ils ont prouvé que, dans le cas où toutes les transitions sont contrôlables, une contrainte (\vec{c}, k) de type *CGEM* peut être satisfaite d'une façon optimale en ajoutant au RdP initial (non contrôlé) (N, M_0) , de P places et de matrice d'incidence W , une place *moniteur* s . Cette place est en amont (respectivement en aval) de toutes les transitions d'entrée (respectivement de sortie) de toute place p appartenant au support

\vec{c} avec $\vec{c}[p] > 0$. Le RdP résultant (N^s, M_0^s) aura un ensemble de places $P \cup \{s\}$, une matrice d'incidence W^s définie par $W^s = \begin{pmatrix} W \\ -\vec{c}^T \cdot W \end{pmatrix}$ et un marquage initial $M_0^s = \begin{pmatrix} M_0 \\ k - \vec{c}^T \cdot M_0 \end{pmatrix}$.

Notons que, le problème de *CGEM* est résolu en ajoutant autant de places que de contraintes et les contraintes redondantes, qui caractérisent le même état admissible, sont satisfaites en ajoutant une seule place moniteur.

Cette méthode permet certes de construire un contrôleur RdP efficace, mais nécessite la représentation des spécifications de contrôle par des *CGEM* ce qui est impossible dans notre cas.

Exemple 1 : Soit le RdP de la figure 3.3 où toutes les transitions sont contrôlables.

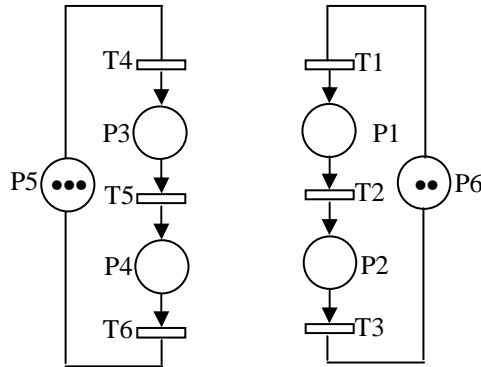


FIGURE 3.3 – Le modèle RdP à contrôler avec une *CGEM*

Supposons que les places p_1 et p_4 représentent deux opérations qui se font sur la même ressource, qui n'est disponible qu'en un seul exemplaire. La contrainte de partage de cette ressource peut se traduire par la condition suivante :

$$M_{p_1} + M_{p_4} \leq 1$$

Cette contrainte est une *CGEM* avec $\vec{c} = (1 \ 0 \ 0 \ 1)^T$ et $k = 1$. D'après Giua *et al.*, elle peut être satisfaite par une place moniteur s à ajouter au modèle initial. Cette place est caractérisée par un vecteur d'incidence $W^s(s, \cdot) = [-1 \ 1 \ 0 \ 0 \ -1 \ 1]$ et un marquage initial $M_0^s = 1$ comme montré sur la figure 3.4.

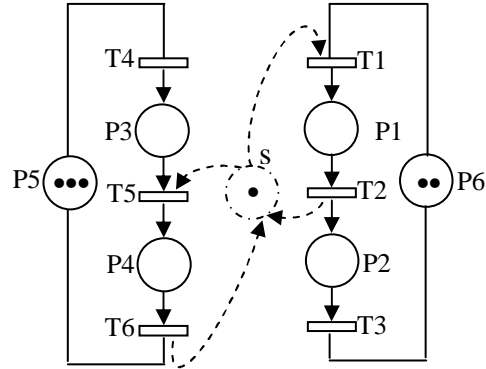


FIGURE 3.4 – Le modèle RdP contrôlé correspondant au modèle de la figure 3.3

3.3.1.2 Approche de Basile *et al.*

Cette approche [BPG07] fait partie des travaux qui s'intéressent au problème de synthèse d'un contrôleur RdP en satisfaisant des spécifications de contrôle de type *CGEM*. Dans cette approche, les auteurs ont introduit une nouvelle partition de l'ensemble de transitions en se basant sur les coûts de contrôle et ceux d'observation associés à ses transitions. En effet, ils ont considéré une fonction z_o qui associe à chaque transition t son coût d'observation et une autre fonction z_c qui associe à chaque transition t son coût de contrôle.

Dans le premier cas, les auteurs ont considéré que ces coûts dépendent du nombre d'arcs en amont et en aval de t . En effet, si la place moniteur a un arc en amont de la transition t avec un poids $w_c^-(p, t)$, le coût qui désactive cette transition est donné par $w_c^-(p, t)z_c(t)$. Par ailleurs, si la place moniteur a un arc en aval de la transition t avec un poids $w_c^+(p, t)$, le coût qui désactive cette transition est donné par $w_c^+(p, t)z_o(t)$. Ainsi, les auteurs ont prouvé que la place moniteur optimale peut être obtenue en résolvant un problème de programmation linéaire en nombre entier (PLNE) dont la fonction objectif est décrite en fonction de ces coûts.

Dans le deuxième cas, les auteurs considèrent que les coûts d'observation et de contrôle associés à une transition t représentent les coûts d'installation de matériels associés à cette transition (capteur, actionneurs,...). Puis, ils ont prouvé que, la place moniteur la plus optimale est obtenue en résolvant un problème de programmation entière avec une fonction objectif non linéaire décrite en utilisant ces coûts.

Malheureusement, cette méthode n'est pas applicable dans notre cas car on ne dispose pas d'un modèle RdP obtenu à partir d'une modélisation physique mais que l'on cherche à interdire certains comportements du RdP identifié.

3.3.1.3 Approche de Moody *et al.*

Contrairement à l'approche de Giua et al où les auteurs cherchent à synthétiser un contrôleur maximum permissif, cette approche consiste à développer un contrôleur RdP qui puisse répondre de façon sous optimale à des spécifications de contrôle sur l'état de type *CGEM*. L'idée principale de cette approche réside dans l'ajout des places de contrôle au modèle RdP non contrôlé pour créer des invariants de places. Rappelons qu'un invariant de place X , dans un RdP N caractérisé par une matrice d'incidence W , est un vecteur d'entiers satisfaisant l'équation $X^T W = 0$. L'objectif du contrôleur est de forcer le système à respecter des contraintes linéaires de type (\vec{c}, k) qui peuvent être représentées par l'expression suivante :

$$\sum_{i=1}^n a_i \cdot M(p_i) \leq k \quad (3.2)$$

où $M(p_i)$ est le marquage de la place p_i , a_i est un entier représentant le poids du marquage d'une place et k est une borne. Le principe de cette approche est de transformer chaque inégalité de type (3.2) en une égalité en ajoutant une variable positive et entière m_{s_i} appelée la variable d'écart comme suit :

$$\sum_{i=1}^n a_i \cdot M(p_i) + m_{s_i} = k \quad (3.3)$$

Cette variable représente une nouvelle place s_i qui contient le marquage supplémentaire nécessaire pour satisfaire l'égalité. Si nous considérons que W est la matrice d'incidence du RdP non contrôlé, la matrice W^c du système contrôlé sera composée de deux matrices : la matrice originale du modèle W et la matrice d'incidence du contrôleur W^s donnée par :

$$W^s = -c^T \cdot W \quad (3.4)$$

Par ailleurs, le marquage initial des places de contrôle est donnée par : $M_{s_0} = k - c^T \cdot M_0$ avec M_0 est le marquage initial du RdP à contrôler.

Notons que, l'application de cette méthode n'est pas possible dans notre cas. En effet, cette approche nécessite la connaissance de toutes les places concernées par les spécifications de contrôle. Or, dans notre cas, le RdP contient généralement des places non mesurables pour lesquelles nous ne pouvons pas formuler des contraintes.

Exemple 2 : On considère le RdP de la figure 3.5 caractérisé par une matrice d'inci-

dence $W = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & 1 & -1 & 0 & 1 \\ 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$ et un marquage initial $M_0 = [3, 0, 0, 3]^T$.

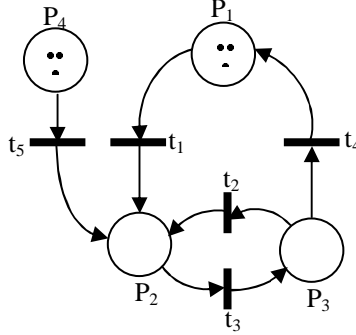


FIGURE 3.5 – Exemple d'un RdP à contrôler

Soit la spécification suivante : contrôler ce RdP afin que les places p_2 et p_3 ne contiennent jamais plus d'un jeton. En utilisant la notion de $CGEM$, cette spécification est traduite par l'équation suivante :

$$M(p_2) + M(p_3) \leq 1 \quad (3.5)$$

ce qui implique que $k = 1$ et $c_1 = 0, c_2 = 1, c_3 = 1, c_4 = 0$ ($c^T = [0, 1, 1, 0]$). L'inégalité (3.5) peut être transformée en une égalité (3.6) en ajoutant une variable entière positive m_{s_1} :

$$M(p_2) + M(p_3) + m_{s_1} = 1 \quad (3.6)$$

Cette égalité correspond au P-semiflt $X^T = [0 \ 1 \ 1 \ 0 \ 1]$, ce qui permet de déterminer $W^s = -c^T \cdot W = [-1 \ 0 \ 0 \ 1 \ -1]$ et $M_0(s_1) = 1$. Le RdP contrôlé est représenté par la figure 3.6.

3.3.2 Approches utilisant la théorie des régions

Cette section présente des méthodes optimales pour la synthèse d'un contrôleur sous la forme de RdP basées sur la théorie des régions. Le contrôleur à synthétiser est le plus permissif possible en tenant compte des spécifications de contrôle, de la contrainte de vivacité et des transitions non contrôlables. Les spécifications considérées sont de type états interdits.

3.3.2.1 Approche de Ghaffari *et al.*

Dans cette approche, les auteurs cherchent à déterminer les places, appelées places de contrôle, à ajouter au modèle RdP à contrôler pour restreindre son graphe d'accessibilité à un sous-ensemble de marquages admissibles. Ces places sont caractérisées

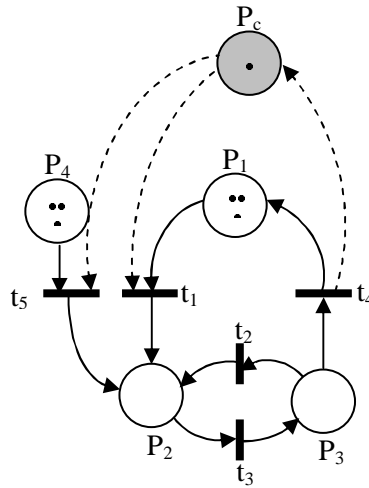


FIGURE 3.6 – Le modèle RdP contrôlé correspondant au modèle de la figure 3.5

en exploitant la théorie des régions. Cette approche procède en deux phases principales. La première phase permet de déterminer l'ensemble de marquages admissibles à partir du graphe de marquages du RdP à contrôler. Cette phase peut se résumer dans l'application de la procédure suivante :

1. Déterminer l'ensemble des marquages interdits.
2. Générer le graphe des marquages partiels.
3. Déterminer l'ensemble des marquages dangereux.
4. Rechercher l'ensemble des marquages autorisés.
5. Rechercher le comportement autorisé du procédé.

La deuxième phase de cette approche consiste à caractériser les places de contrôle à ajouter au modèle RdP à contrôler pour assurer le comportement désiré en utilisant la théorie des régions. L'objectif de cette théorie est de déterminer un RdP pur (N, M_0) caractérisé par une matrice d'incidence W et ayant T comme ensemble de transitions tel que son graphe de marquages soit isomorphe à un graphe orienté fini G dont les arcs sont étiquetés par les éléments de T .

Pour caractériser les places de contrôle à ajouter au RdP initial, les auteurs se sont basés sur l'équation de marquage d'un RdP et certaines propriétés d'un graphe fini. Ces propriétés sont respectivement : l'équation de cycle, les conditions d'atteignabilité d'un marquage quelconque depuis le marquage initial, les conditions de séparation d'événements et les conditions de séparations d'états.

En effet, pour toute place p et pour toute transition t validée par M du RdP (N, M_0) que l'on cherche à établir, l'équation suivante est valable :

$$M'(p) = M(p) + W(p, t), \forall (M, M') \in G \text{ tel que } M[t > M' \quad (3.7)$$

Ainsi, l'application de l'équation précédente pour chaque cycle non orienté de G tout au long des marquages de ce cycle mène à l'équation de cycle donnée par :

$$\sum_{t \in T} W(p, t) \vec{\delta}[t] = 0, \forall \delta \in G \quad (3.8)$$

où δ est un cycle non orienté de G et $\vec{\delta}[t]$ est la somme algébrique de toutes les occurrences de t dans δ .

Par ailleurs, l'application de l'équation (3.7) pour tous les arcs d'un chemin Γ_M , un chemin reliant M_0 à un marquage M , permet d'obtenir l'équation (3.9).

$$M(p) = M_0(p) + W(p, \cdot) \Gamma_M^{\vec{}} \quad (3.9)$$

avec $\Gamma_M^{\vec{}}$ est le vecteur d'occurrence de Γ_M .

Ainsi, l'équation traduisant la condition d'atteignabilité d'un marquage quelconque depuis le marquage initial est obtenue à partir de (3.9) et elle est donnée par :

$$M_0(p) + W(p, \cdot) \Gamma_M^{\vec{}} \geq 0 \quad (3.10)$$

De plus, l'équation décrivant la condition de séparation d'événements relative à un couple (M, t) où t n'est pas validée par M est :

$$M_0(p) + W(p, \cdot) \Gamma_M^{\vec{}} + W(p, t) < 0 \quad (3.11)$$

Enfin, les marquages atteignables par le RdP contrôlé sont deux à deux différents si et seulement si la condition appelée condition de séparation d'états est vérifiée. Cette condition est décrite par l'équation suivante :

$$\forall M, M' \in G, \exists p \text{ telle que } C(p, \cdot) \Gamma_M^{\vec{}} \neq C(p, \cdot) \Gamma_{M'}^{\vec{}} \quad (3.12)$$

Partant des propriétés précédentes, Ghaffari [Gha02] a montré que le RdP (N, M_0) ayant pour graphe d'atteignabilité le graphe G existe si et seulement si l'ensemble P de places $\{M_0(p), W(p, \cdot)\}$ vérifié les conditions suivantes :

1. Chaque place p satisfait les équations de cycles (3.8) de G et les conditions d'atteignabilité (3.10) des marquages de G ;
2. L'ensemble des places P vérifie la condition (3.12) ;

3. Pour chaque couple (M, t) où t n'est pas franchissable depuis M , il existe au moins une place p qui satisfait la condition de séparation d'événements (3.11).

Notons que cette théorie a été exploitée par Achour *et al.* [ARX04] [ARX05] pour résoudre le PEI caractérisé par un ensemble de $CGEM$ sous observation partielle d'événements. Ils ont aussi utilisé cette théorie pour synthétiser un contrôleur RdP qui satisfait une nouvelle classe de contraintes qui sont les contraintes temporelles [AR05][AR07].

Malheureusement, cette méthode n'est pas applicable dans notre cas parce que le RdP à contrôler contient des places non mesurables sur lesquelles nous ne pouvons pas imposer des contraintes.

Exemple 3 [Gha02] : Soit un modèle RdP et son graphe de marquage donné par la figure 3.7. Notons que, dans le graphe d'atteignabilité seulement les marquages des places p_1, p_2 et p_3 sont indiqués.

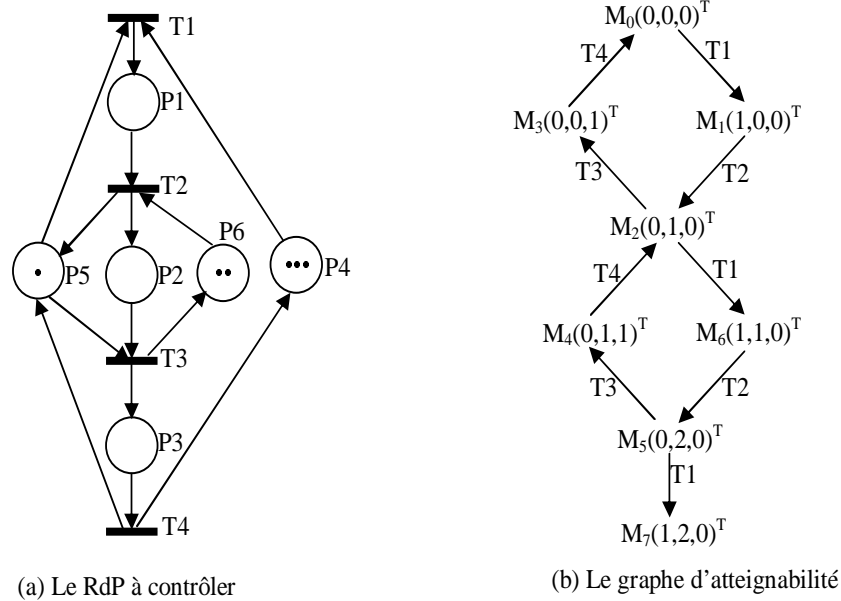


FIGURE 3.7 – Un exemple de problèmes d'états interdits

Le RdP peut entrer dans une situation de blocage représentée par l'état $M_7 = (1 \ 2 \ 0 \ 0 \ 0 \ 0)^T$. Une spécification de contrôle peut donc consister à interdire au modèle d'accéder à M_7 , ce qui revient à empêcher toute transition d'état qui y mène. En supposant que toutes les transitions sont contrôlables, le comportement admissible pour cet exemple d'états interdits inclut tous les noeuds excepté M_7 et tous les arcs

excepté $(M_5 \xrightarrow{T_1} M_7)$. On doit donc, déterminer les places de contrôle à ajouter au RdP initial pour conserver un comportement vivant. Dans ce cas, il y a une seule condition de séparation d'événements à satisfaire par une nouvelle place de contrôle p_c :

$$M_{c7}(p_c) = M_{c0}(p_c) + 2W_c(p_c, T_1) + 2W_c(p_c, T_2) + W_c(p_c, T_1) < 0 \quad (3.13)$$

Les conditions d'atteignabilité des marquages autorisés sont :

$$\begin{aligned} M_{c0}(p_c) &\geq 0; \\ M_{c1}(p_c) &= M_{c0}(p_c) + W_c(p_c, T_1) \geq 0; \\ M_{c2}(p_c) &= M_{c0}(p_c) + W_c(p_c, T_1) + W_c(p_c, T_2) \geq 0; \\ M_{c3}(p_c) &= M_{c0}(p_c) + W_c(p_c, T_1) + W_c(p_c, T_2) + W_c(p_c, T_3) \geq 0; \\ M_{c4}(p_c) &= M_{c0}(p_c) + 2W_c(p_c, T_1) + W_c(p_c, T_2) \geq 0; \\ M_{c5}(p_c) &= M_{c0}(p_c) + 2W_c(p_c, T_1) + 2W_c(p_c, T_2) \geq 0; \\ M_{c6}(p_c) &= M_{c0}(p_c) + 2W_c(p_c, T_1) + 2W_c(p_c, T_2) + W_c(p_c, T_3) \geq 0; \end{aligned} \quad (3.14)$$

Par ailleurs, le graphe de marquages contient deux cycles qui ont la même équation :

$$W_c(p_c, T_1) + W_c(p_c, T_2) + W_c(p_c, T_3) + W_c(p_c, T_4) = 0 \quad (3.15)$$

Le système constitué des équations (3.13), (3.14) et (3.15) peut être résolu par l'ajout d'une seule place de contrôle p_c telle que $W_c(p_c, \cdot) = (-1 \ 0 \ 0 \ 1)$ et $M_{c0}(p_c) = 2$. Le modèle contrôlé est donné par la figure 3.8.

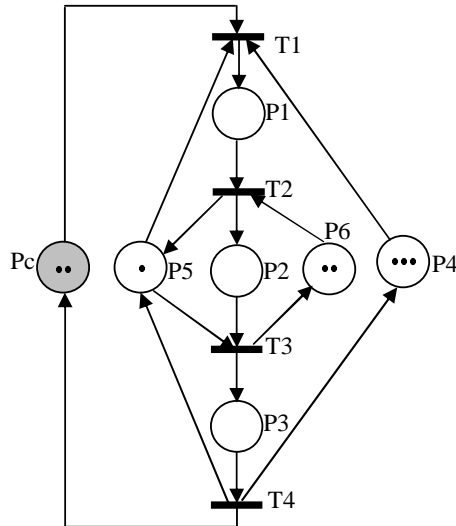


FIGURE 3.8 – Une solution du problème de blocage avec une place de contrôle

3.3.2.2 Approche de Lee *et al.*

Dans cette approche, une nouvelle méthode de synthèse d'une commande supervisée basée sur le formalisme RdP d'un système manufacturier reconfigurable est présentée. Un nouveau type de spécifications modélisables par RdP a été proposé. Le problème associé à ces spécifications est noté problème de séquences de transitions d'état interdites (SITE). Pour résoudre ce problème, les auteurs ont représenté le RdP du procédé et celui de spécifications en boucle ouverte. Notons que, le RdP de spécifications traduit un comportement souhaité et non pas un comportement interdit. Il est construit à l'aide de transitions de modèle RdP du procédé. Les transitions communes entre les deux modèles sont supposées être synchronisées.

La première étape de cette approche consiste à construire le Graphe d'Accessibilité Synchronique Contraint (GASC). Pour ce faire, les auteurs utilisent le modèle du procédé et celui de la spécification, supposent que ces modèles contiennent des transitions synchrones et considèrent le marquage initial comme étant une concaténation des marquages des deux RdP. Par ailleurs l'algorithme de construction du GASC permet de déterminer : l'ensemble de marquages autorisés, les séquences de transitions autorisées, l'ensemble des marquages interdits et les séquences de transitions interdites.

Dans la deuxième étape, les cycles correspondant aux séquences de transitions autorisées dans le GASC sont exploités en utilisant la théorie des régions pour déterminer une solution optimale du problème. Malheureusement, cette méthode est basée sur la représentation des spécifications de contrôle et du système à contrôler par des RdP en boucle ouverte ce qui n'est pas possible dans notre cas.

3.3.3 Discussion

La littérature concernant la commande supervisée d'un SED par RdP est abondante. Des approches efficaces ont été proposées pour traiter les PEI ou les PTEI. Cependant, ces approches ne peuvent pas être exploitées directement dans notre cas parce qu'il nous est impossible de représenter les états interdits avec des contraintes. En effet, pour synthétiser un contrôleur, il est indispensable d'avoir une spécification de contrôle. Généralement, la spécification de contrôle d'un système interprète des exigences à respecter durant le fonctionnement de ce dernier. Par exemple, une machine quelconque ne peut pas usiner deux pièces à la fois. En connaissant ces exigences, on peut les représenter sous formes de contraintes qui sont décrites généralement par des contraintes linéaires. Ces contraintes sont formulées en utilisant des éléments du vecteur de marquage et/ou celui de franchissement des transitions. Chaque comportement du modèle de ce système qui ne respecte pas ces contraintes est un comportement illégal. Cependant, dans notre cas, nous ne disposons ni d'une description physique du

système modélisé par le RdP à contrôler ni d'une spécification de contrôle car ce RdP est obtenu en analysant les signaux d'entrée et de sortie mesurables. En effet, bien qu'il soit possible parfois de formuler des contraintes sur la partie mesurable, aucune contrainte ne pourrait être formulée sur la partie non mesurable.

Ainsi, pour empêcher le modèle RdP identifié d'atteindre des états interdits nous proposons dans ce qui suit deux approches de synthèse d'un contrôleur RdP. Ces approches concernent le problème d'états interdits où les RdP à contrôler sont de transitions contrôlables et contiennent des places mesurables et non mesurables. Par ailleurs, partant du fait que toutes les transitions sont contrôlables (les transitions sont associées aux événements générés par le contrôleur dont on ne dispose pas de spécification) il devient évident que le graphe de marquages du modèle RdP à contrôler ne contient aucun marquage dangereux. Nous supposons aussi que ce graphe de marquages ne contient aucun état de blocage. Cette dernière hypothèse s'explique par le fait que les signaux d'entrée et de sortie observés lors de la phase d'identification représentent les comportements normaux (positifs) du système considéré.

Le problème d'états interdits (PEI) considéré dans ce cas sera traité comme un problème plus général, à savoir le problème de transitions d'état interdites (PTEI). Ce choix est motivé par le fait que nous visons à développer des méthodes de synthèse d'un contrôleur RdP permettant d'empêcher à la fois le RdP d'atteindre des états interdits et d'inhiber les transitions conduisant vers ces états. La première méthode proposée est basée sur l'utilisation des chemins d'influence des transitions interdites. Ces chemins, calculés à partir du graphe de marquage du modèle RdP, sont constitués de transitions et de marquages interdits. Quant à la deuxième méthode, elle consiste à utiliser les séquences de transitions interdites déduites du graphe de marquages du modèle RdP à contrôler. Ces séquences représentent le langage interdit généré par le RdP à contrôler.

Ci-après, les deux méthodes proposées pour la synthèse d'un contrôleur RdP sont présentées.

3.4 Synthèse d'un contrôleur RdP à base de chemins d'influence

Cette section présente une première approche que nous proposons pour synthétiser un contrôleur RdP. Le problème considéré ici inclut mais n'est pas restreint au problème d'états interdits, c'est le problème de transitions d'état interdites. Notons que, la méthode proposée dans cette section est valable pour tout modèle RdP ordinaire pur et borné. Rappelons que l'ensemble de places du modèle RdP considéré est constitué

de deux sous-ensembles : un sous-ensemble de places mesurables et un sous-ensemble de places non mesurables. Ce qui implique que le vecteur de marquage du RdP est composé d'une partie mesurable et d'une partie non mesurable : $M_i = \begin{bmatrix} M_{im} \\ M_{i\bar{m}} \end{bmatrix}$.

Le problème de la synthèse d'un contrôleur RdP considéré dans ce cas peut être résolu par l'ajout de places de contrôle $\{p_{c1}, \dots, p_{ck}\}$. Ces dernières peuvent être obtenues en répondant aux questions suivantes : (1) est ce que l'on doit ajouter autant de places de contrôle que de transitions d'état interdites ? (2) quelles sont les transitions en aval et en amont de chaque place de contrôle ? (3) quel est le marquage initial de chaque place ajoutée ?

Ainsi, pour répondre à ces questions nous proposons d'utiliser les chemins d'influence des transitions interdites. Ces chemins sont calculés à partir du graphe de marquage du modèle RdP à contrôler. En effet, une fois que les transitions d'état interdites sont identifiées, les chemins d'influence des transitions interdites sont calculés pour déterminer les places de contrôle à ajouter.

Avant de présenter les détails de l'algorithme de synthèse d'un contrôleur RdP, considérons les définitions suivantes qui seront utilisées dans la suite.

Définition 3 : Un chemin d'influence $\pi_{\alpha_i} = t_{\alpha_i}, M_{\alpha_i}, t_{\alpha_j}, \dots, t_{\alpha_k}, M_{\alpha_f}$ d'une transition interdite t_{α_i} dans le graphe de marquages du modèle RdP à contrôler, est un chemin orienté composé d'une succession de marquages interdits et de transitions interdites. Il relie la transition interdite t_{α_i} à un marquage légal M_{α_f} tel que toutes les transitions et tous les marquages entre t_{α_i} et M_{α_f} sont interdits.

Notons que, la transition en amont du premier marquage interdit dans un chemin d'influence π_{α_i} est notée $t_{in}(\pi_{\alpha_i})$ et la transition en aval du dernier marquage interdit dans ce chemin est notée $t_{fi}(\pi_{\alpha_i})$. Un chemin d'influence π_{α_i} d'une transition interdite t_{α_i} peut être représenté par la figure 3.9.

Remarque 1 : Une transition interdite t_{α_i} peut avoir plusieurs chemins d'influence et l'ensemble de ces chemins est noté par : $\Pi_i = \{\pi_{\alpha_i}^1, \pi_{\alpha_i}^2, \dots, \pi_{\alpha_i}^k\}$ avec $k \in \{1, \dots, |\Pi_i|\}$ et $|\Pi_i|$ est le cardinal de l'ensemble Π_i .

Définition 4 : Un chemin d'influence π_{α_j} d'une transition interdite t_{α_j} est dit inclus dans un chemin d'influence π_{α_i} d'une transition interdite t_{α_i} , et on note $\pi_{\alpha_j} \subset \pi_{\alpha_i}$, si et seulement si les conditions suivantes sont satisfaites simultanément :

1. $\forall t_{\alpha_k} \in \pi_{\alpha_j} \mid t_{\alpha_k} \in \pi_{\alpha_i}$.
2. $\forall M_{\alpha_i} \in \pi_{\alpha_j} \mid M_{\alpha_i} \in \pi_{\alpha_i}$.
3. π_{α_i} doit être écrit en utilisant π_{α_j} comme suit : $\pi_{\alpha_i} = t_{\alpha_i}, M_{\alpha_i}, \dots, \pi_{\alpha_j}$.

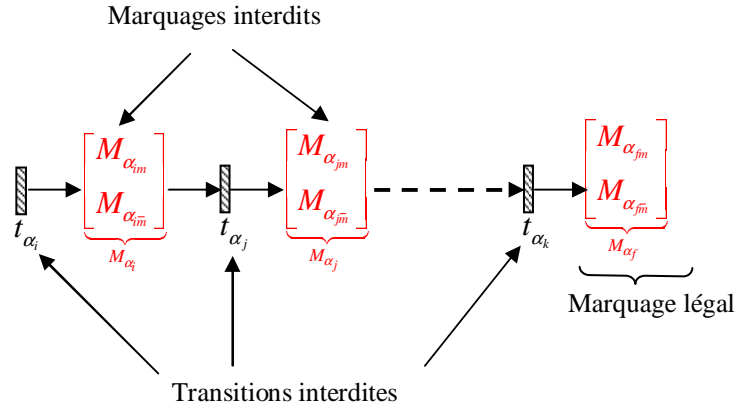


FIGURE 3.9 – Un chemin d'influence π_{α_i} d'une transition t_{α_i}

La première et la deuxième condition permettent de vérifier la relation d'inclusion de chaque élément de π_{α_j} dans π_{α_i} . Ces conditions signifient que chaque transition interdite et chaque marquage interdit de π_{α_j} appartient aussi à π_{α_i} . Cependant, ces deux conditions permettent de confirmer l'appartenance de chaque élément de π_{α_j} à π_{α_i} sans prendre en compte l'ordre de succession de ces marquages et transitions. Pour cette raison, si la troisième condition n'est pas vérifiée alors la relation d'inclusion devient seulement une relation d'intersection entre deux chemins.

Notons que, une transition interdite peut avoir plusieurs chemins d'influence qui vérifient la relation d'inclusion entre eux. En se basant sur ces définitions, nous proposons un algorithme qui permet d'affiner le modèle RdP issu de la phase d'identification par l'ajout des places de contrôle pour l'empêcher d'accéder à des marquages interdits. Cet algorithme s'exécute selon les étapes suivantes :

1. Identification de l'ensemble de marquages interdits.
2. Détermination de l'ensemble de transitions d'état interdites équivalentes.
3. Synthèse d'un contrôleur RdP en utilisant les chemins d'influence de transitions interdites déduits à partir du graphe de marquage du modèle RdP à contrôler.

Ces étapes seront expliquées dans ce qui suit.

3.4.1 Identification des marquages interdits

Au mieux de nos connaissances les travaux portant sur la synthèse de contrôleurs RdP définissent explicitement les marquages (états) interdits par des contraintes utilisant les éléments du vecteur de marquage et/ou ceux de vecteurs de franchissement des transitions. Or, dans notre contexte, nous disposons uniquement d'un ensemble d'états mesurables du système. Ainsi, nous devons déterminer les marquages interdits

en confrontant les comportements du système réel et ceux générés par le RdP obtenu lors de la phase d'identification. L'ensemble des états accessibles par le système réel est noté par \mathcal{E} comme indiqué dans le chapitre 2. Quant à l'ensemble de comportements générés par le RdP à contrôler est représenté par le graphe de marquages de ce RdP noté G . Dans notre cas, un marquage M_{α_i} atteignable par le modèle RdP à contrôler est dit un marquage interdit si et seulement si la partie mesurable de ce marquage ne correspond à aucun état atteignable par le système réel, c.à.d, $\forall M_{\alpha_i} \in G$, M_{α_i} est dit interdit si et seulement si $\nexists E_i \in \mathcal{E} : M_{\alpha_{im}} = E_i$ sinon, c'est un marquage légal. L'ensemble de marquages interdits est obtenu en appliquant la procédure suivante :

Données : \mathcal{E} et G .

Objectif : identifier M_{fr} l'ensemble de marquages interdits.

Debut

1. Initialiser $M_{fr} = \emptyset$.
2. Pour chaque marquage $M_{\alpha_i} \in G$ faire :
 - (a) S'il existe un état dans l'ensemble d'états \mathcal{E} tel que $M_{\alpha_{im}} = E_i$ avec E_i est la partie mesurable de cet état alors :
 - M_{α_i} est un marquage légal.
 - (b) Sinon M_{α_i} est un marquage interdit.
 - Mettre à jour l'ensemble de marquages interdits : $M_{fr} := M_{fr} \cup \{M_{\alpha_{im}}\}$.

Fin Pour.

Fin.

Une fois l'ensemble de marquages interdits est obtenu, nous devons déterminer les transitions en aval et en amont de chaque marquage dans cet ensemble pour empêcher leurs franchissements.

3.4.2 Détermination de l'ensemble des transitions d'état interdites

Une transition d'état notée $(M_{\alpha_i} [t_{\alpha_i} > M_{\alpha_j}])$ est dite transition d'état interdite si et seulement si une des conditions ou les deux conditions suivantes sont vérifiées :

- (a) M_{α_i} est un marquage interdit.
- (b) M_{α_j} est un marquage interdit.

Ces deux conditions peuvent être reformulées comme suit : Une transition d'état $(M_{\alpha_i} \xrightarrow{t_{\alpha_i}} M_{\alpha_j}) \in G$ est dite interdit ssi :

- $\exists E_i \in \mathcal{E} : M_{\alpha_{im}} = E_i$ et $\nexists E_j \in \mathcal{E} : M_{\alpha_{jm}} = E_j$ ou,
- $\nexists E_i \in \mathcal{E} : M_{\alpha_{im}} = E_i$ et $\exists E_j \in \mathcal{E} : M_{\alpha_{jm}} = E_j$ ou,
- $\nexists E_i, E_j \in \mathcal{E} : M_{\alpha_{im}} = E_i$ et $M_{\alpha_{jm}} = E_j$.

Preuve : l'atteignabilité d'un marquage interdit peut être empêchée en interdisant le franchissement des transitions en amont de ce marquage. Par ailleurs, chaque transition franchie depuis un marquage interdit est une transition interdite. Donc, chaque transition soit en aval ou en amont d'un marquage interdit est une transition interdite. L'ensemble de transitions d'état interdites est obtenu en appliquant la procédure suivante :

Données : \mathcal{E} , G et M_{fr} .

Objectif : identifier Ψ l'ensemble de transitions d'état interdites.

Debut

1. Initialiser $\Psi = \emptyset$.
 2. Pour chaque transition d'état $(M_{\alpha_i} \xrightarrow{t_{\alpha_i}} M_{\alpha_j}) \in G$ faire :
 - (a) Si la condition (a) et/ou (b) est vérifiée alors $(M_{\alpha_i} \xrightarrow{t_{\alpha_i}} M_{\alpha_j})$ est une transition d'état interdite.
 - Mettre à jour l'ensemble de transitions d'état interdite : $\Psi := \Psi \cup (M_{\alpha_i} \xrightarrow{t_{\alpha_i}} M_{\alpha_j})$.
 - (b) Sinon, $(M_{\alpha_i} \xrightarrow{t_{\alpha_i}} M_{\alpha_j})$ est une transition d'état légale.
Fin Si.
- Fin Pour.

Fin.

3.4.3 Synthèse d'un contrôleur RdP : Algorithme 3

En utilisant les chemins d'influence des transitions interdites déduits à partir du graphe de marquages du modèle RdP à contrôler, le contrôleur RdP est élaboré en appliquant l'algorithme 3, qui consiste à :

1. Générer le graphe de marquage du modèle RdP à contrôler.
2. Identifier l'ensemble de marquages interdits et déduire l'ensemble de transitions d'état interdites équivalent.
3. Calculer l'ensemble des chemins d'influence de chaque transition interdite.
4. Mettre à jour l'ensemble des chemins d'influence de chaque transition interdite en vérifiant la relation d'inclusion entre eux.
5. Mettre à jour l'ensemble des chemins d'influence de toutes les transitions interdites en vérifiant la relation d'inclusion entre eux.
6. Utiliser les chemins d'influence de l'ensemble final pour caractériser les places de contrôle à ajouter au modèle RdP identifié.

Algorithme 3 : Algorithme de synthèse d'un contrôleur RdP (1)

Données : \mathcal{E} , l'ensemble d'états du système, (N, M_0) : le RdP à contrôler.

Debut

1. Générer G le graphe de marquages du modèle RdP à contrôler (N, M_0) .
2. Déterminer M_{fr} l'ensemble de marquages interdits et déduire Ψ l'ensemble de transitions d'état interdites (cf §3.4.1, §3.4.2).
3. **pour** chaque transition t_{α_i} dans l'ensemble Ψ **faire**
 - 3.1. Calculer l'ensemble de ses chemins d'influence Π_i .
 - 3.2. Mettre à jour Π_i en vérifiant la relation d'inclusion entre les chemins d'influence de la même transition comme suit :
 - **pour** chaque $\pi_{\alpha_i}^m \in \Pi_i$ et $\pi_{\alpha_i}^n \in \Pi_i$ **faire**
 - si** $\pi_{\alpha_i}^m \subset \pi_{\alpha_i}^n$ **alors**
 - Mettre à jour Π_i en éliminant $\pi_{\alpha_i}^m$ ($\Pi_i := \Pi_i \setminus \{\pi_{\alpha_i}^m\}$).
4. Mettre tous les chemins d'influence de toutes les transitions interdites dans le même ensemble noté par Π' tel que $\Pi' = \{\Pi_i | i \in \{1, \dots, k\}\}$ (k est l'indice de la transition interdite).
5. Mettre à jour Π' en vérifiant la relation d'inclusion entre chaque deux chemins d'influence de chaque deux transitions distinctes comme suit :
 - **pour** chaque deux chemins $\pi_{\alpha_i}^q, \pi_{\alpha_j}^l \in \Pi'$ **faire**
 - si** $\pi_{\alpha_i}^q \subset \pi_{\alpha_j}^l$ **alors**
 - Mettre à jour Π' ($\Pi' := \Pi' \setminus \{\pi_{\alpha_i}^q\}$).
6. **pour** chaque chemin d'influence $\pi_{\alpha_i}^r \in \Pi'$ **faire**
 - 6.1. Déterminer $t_{in}(\pi_{\alpha_i}^r)$ la transition en amont de premier marquage interdit dans ce chemin et $t_{fi}(\pi_{\alpha_i}^r)$ la transition en aval de dernier marquage interdit dans le même chemin.
 - 6.2. Ajouter une place de contrôle p_{ci} au modèle RdP actuel.
 - 6.3. Connecter la place de contrôle ajoutée au modèle RdP actuel par des arcs de poids égal à 1 dont $t_{in}(\pi_{\alpha_i}^r)$ sera en aval de p_{ci} et $t_{fi}(\pi_{\alpha_i}^r)$ sera en amont de p_{ci} .
 - 6.4. Marquer la place p_{ci} avec un marquage initial équivalent au nombre de franchissements autorisés de la transition interdite $t_{in}(\pi_{\alpha_i}^r)$ depuis le marquage initial du modèle RdP à contrôler.
 - 6.5. **si** il existe plusieurs chemins d'influence avec les mêmes transitions t_{in} et t_{fi} **alors**
 - ajouter une seule place de contrôle p_{ci} et la marquer avec un marquage initial égal au nombre maximum que la transition interdite $t_{in}(\pi_{\alpha_i}^r)$ est autorisée de franchir

Fin.

Par rapport aux méthodes proposées dans la littérature, notre algorithme permet de synthétiser un contrôleur RdP en connaissant seulement les états observés du système réel et les comportements du RdP à contrôler. Ce dernier est obtenu en utilisant des signaux d'entrée/sortie et sans aucune spécification de contrôle. L'algorithme proposé est basé sur l'analyse des chemins d'influence des transitions interdites qui sont à leurs tours déduits à partir du graphe de marquages du modèle RdP à contrôler. Le nombre de places à ajouter par cet algorithme n'est pas forcément égale au nombre de transitions d'état interdites car chaque chemin d'influence $\pi_{\alpha_i}^r$ dans l'ensemble final des chemins Π' caractérise une place de contrôle à ajouter.

En outre, le RdP contrôlé atteint tous les états représentant les comportements positifs du système sans atteindre ni des états de blocages ni des états dangereux. Ceci indique que le contrôleur est maximum permissif. Ainsi, pour évaluer la complexité de cet algorithme nous considérons m le nombre de places dans le RdP à contrôler, n le nombre de transitions et $|\pi_{\alpha_i}^r|$ la longueur du plus grand chemin d'influence dans l'ensemble Π' , dans le pire cas les inconnus sont $c = |\pi_{\alpha_i}^r|(m \times n) + m + n$.

3.4.4 Exemple illustratif 1

Pour illustrer cet algorithme, un système caractérisé par un ensemble d'états \mathcal{E} est considéré avec :

$$\mathcal{E} = \left\{ \underbrace{\begin{pmatrix} 0 \\ 0 \\ 3 \\ 2 \end{pmatrix}}_{E_0}, \underbrace{\begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}}_{E_1}, \underbrace{\begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \end{pmatrix}}_{E_2}, \underbrace{\begin{pmatrix} 0 \\ 0 \\ 2 \\ 1 \end{pmatrix}}_{E_3}, \underbrace{\begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix}}_{E_4}, \underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}}_{E_5}, \underbrace{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}}_{E_6} \right\}.$$

L'application de la deuxième méthode d'identification proposée dans le chapitre 2 permet d'établir le modèle RdP de la figure 3.10.

L'analyse de ce modèle montre qu'il produit tous les comportements du système réel mais il atteint également des états qui ne sont pas atteignables par le système. Ces états sont considérés comme des états interdits qu'il faut inhiber en ajoutant des places de contrôle.

L'application de la première étape qui consiste à calculer le graphe de marquages G du modèle RdP à contrôler donne le résultat représenté par la figure 3.11.

L'application de la procédure introduite dans la sous-section 3.4.1 permet de conclure l'ensemble de marquages interdits $M_{fr} = \{M_7, M_8, M_9\}$.

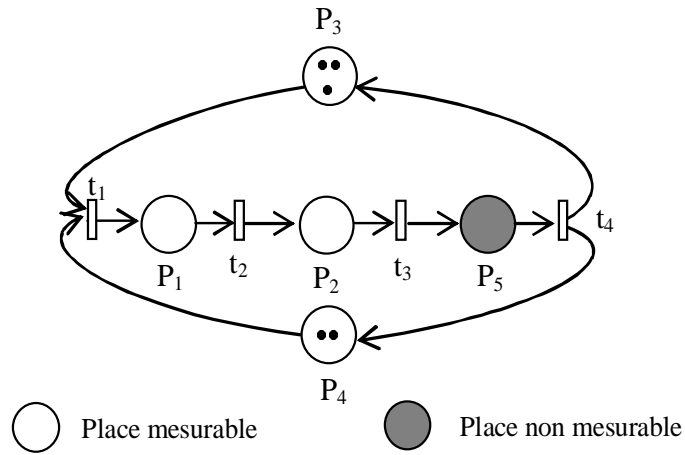


FIGURE 3.10 – Le modèle RdP à contrôler

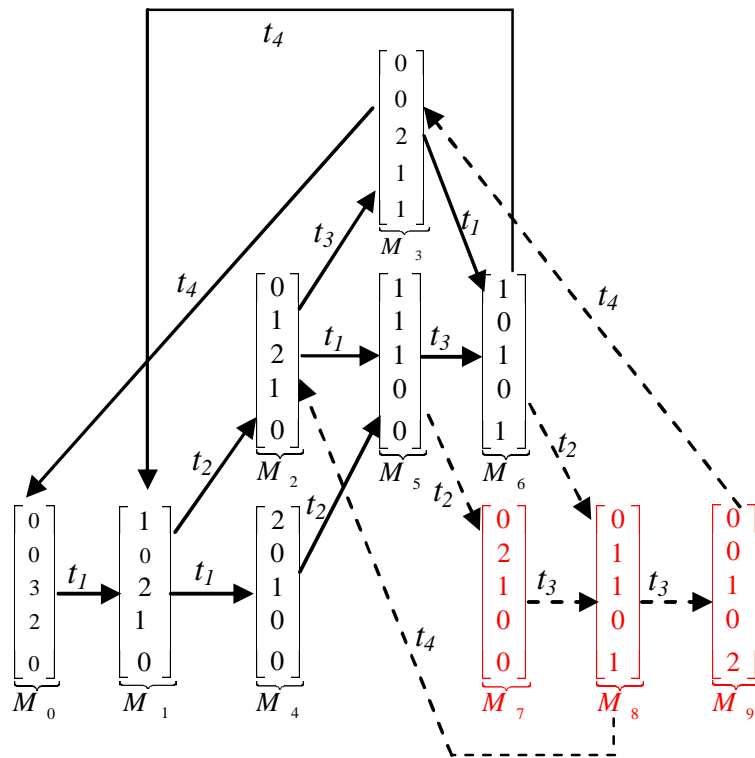


FIGURE 3.11 – Le graphe de marquages du modèle RdP à contrôler

Ainsi, l'exécution de la procédure indiquée dans la sous-section 3.4.2 montre que l'ensemble de transitions d'état interdites est :

$$\Psi = \{(M_5 \xrightarrow{t_2} M_7), (M_7 \xrightarrow{t_3} M_8), (M_8 \xrightarrow{t_2} M_9), (M_9 \xrightarrow{t_4} M_3), (M_6 \xrightarrow{t_2} M_8), (M_8 \xrightarrow{t_4} M_2)\}.$$

Par ailleurs, les chemins d'influence de chaque transition interdite dans Ψ sont :

○ Pour t_2 : $\pi_2^1 = t_2M_7t_3M_8t_3M_9t_4M_3$, $\pi_2^2 = t_2M_8t_3M_9t_4M_3$, $\pi_2^3 = t_2M_7t_3M_8t_4M_2$, $\pi_2^4 = t_2M_8t_4M_2$, donc $\Pi_2 = \{\pi_2^1, \pi_2^2, \pi_2^3, \pi_2^4\}$.

○ Pour t_3 : $\pi_3^1 = t_3M_8t_3M_9t_4M_3$, $\pi_3^2 = t_3M_9t_4M_3$, $\pi_3^3 = t_3M_8t_4M_2$, donc $\Pi_3 = \{\pi_3^1, \pi_3^2, \pi_3^3\}$.

○ Pour t_4 : $\pi_4^1 = t_4M_3$, $\pi_4^2 = t_4M_2$, donc $\Pi_4 = \{\pi_4^1, \pi_4^2\}$.

La vérification de la relation d'inclusion des chemins d'influence des transitions interdites montre que $\pi_3^2 \subset \pi_3^1$. Dans ce cas, nous devons éliminer π_3^2 de l'ensemble des chemins d'influence de t_3 , ce qui permet d'obtenir $\Pi_3 = \{\pi_3^1, \pi_3^3\}$. Ainsi, l'ensemble des chemins d'influence à cette phase est $\Pi' = \{\Pi_2, \Pi_3, \Pi_4\}$.

De la même manière, la vérification de la relation d'inclusion des chemins d'influence de l'ensemble Π' montre que : $\pi_4^2 \subset \{\pi_3^3, \pi_2^4, \pi_2^3\}$, $\pi_4^1 \subset \{\pi_3^1, \pi_2^2, \pi_2^1\}$, $\pi_3^3 \subset \pi_2^3$ et $\pi_3^1 \subset \pi_2^1$.

Ainsi, l'ensemble final de tous les chemins est : $\Pi' = \Pi_2 = \{\pi_2^1, \pi_2^2, \pi_2^3, \pi_2^4\} = \{t_2M_7t_3M_8t_3M_9t_4M_3, t_2M_8t_3M_9t_4M_3, t_2M_7t_3M_8t_4M_2, t_2M_8t_4M_2\}$.

Finalement, l'analyse de Π' montre que tous les chemins d'influence ont la même transition en amont du premier marquage interdit. Cette transition est $t_{in} = t_2$. Il montre aussi que tous les chemins d'influence possèdent la même transition en aval du dernier marquage interdit. Cette transition est $t_{fi} = t_4$. Ainsi, le contrôleur RdP se résume en une place notée p_{c1} . Cette place est en amont de t_2 et en aval de t_4 comme l'indique la figure 3.14. Quant à son marquage initial, il est égal à 1 ($M_{c0}(p_{c1}) = 1$) car le franchissement de la transition t_2 est autorisé au maximum une fois depuis le marquage initial.

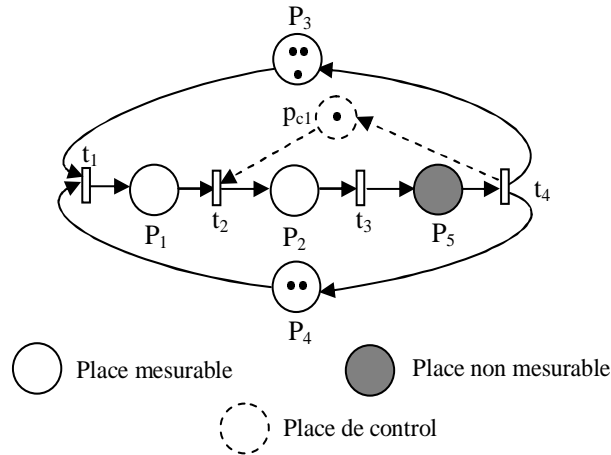


FIGURE 3.12 – Le modèle RdP contrôlé en appliquant l’algorithme 3

3.5 Synthèse d’un contrôleur RdP à base de séquences de transitions interdites

Cette section présente une deuxième approche que nous proposons pour synthétiser un contrôleur RdP. Dans cette approche, les places de contrôle à ajouter au modèle RdP à contrôler sont caractérisées en utilisant les séquences de transitions interdites. Ces dernières sont calculées à partir du graphe de marquage du modèle RdP à contrôler et elles sont considérées comme un langage interdit généré par ce RdP.

La synthèse du contrôleur dans cette approche se fait en trois étapes. Les deux premières consistent à déterminer l’ensemble de marquages interdits M_{fr} et l’ensemble de transitions d’état interdites Ψ . Ces deux étapes ont été déjà présentées dans les sections §3.4.1 et §3.4.2. Ainsi, dans la suite de cette partie nous présentons la troisième étape qui permet de déterminer l’ensemble de séquences de transitions interdites. Une fois que ce dernier est obtenu, l’algorithme de synthèse du contrôleur sera présenté et illustré à l’aide d’un exemple académique.

3.5.1 Détermination de l’ensemble des séquences de transitions interdites

Avant de présenter la procédure, qui permet de déterminer ces séquences, les notions et les définitions suivantes qui seront utilisées dans la suite sont introduites.

Définition 5 : Une séquence de transitions σ_i est dite interdite si et seulement si

elle autorise le franchissement d'au moins une transition interdite. En effet, $\sigma_i = t_1, t_2, \dots, t_k, t_l, \dots, t_f$ est dite interdite ssi : $\exists t_i \in \sigma_i : *M(t_i) \in M_{fr}$ ou $M^*(t_i) \in M_{fr}$ avec $*M(t_i)$ et $M^*(t_i)$ représentent respectivement les transitions en amont et en aval de t_i dans G sinon, elle est légale.

Remarque 2 : Dans notre cas, chaque séquence de transitions interdite est composée de deux sous séquences (voir la figure 3.13) comme suit : $\sigma_i = \underbrace{t_1, \dots, t_k}_{\sigma_i^l} \mid \underbrace{t_l, \dots, t_f}_{\sigma_i^{in}}$, avec

σ_i^l est la sous séquence de transitions légale autorisée de franchir depuis le marquage initial M_0 et elle peut être égale à ϵ (i.e., $\sigma_i^l = \epsilon$).

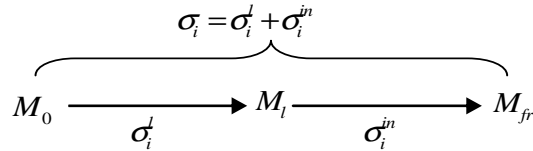


FIGURE 3.13 – marquages légaux et marquages interdits

Ceci signifie qu'il y a des états interdits atteignables depuis le marquage initial après le franchissement d'une seule transition. Ainsi, σ_i^{in} est une sous séquence de transitions interdites de franchir depuis σ_i^l . Notons que, $*\sigma_i^{in}$ et σ_i^{in*} représentent respectivement la première et la dernière transition de σ_i^{in} . Donc, tous les marquages atteignables depuis M_0 après le franchissement de la sous séquence de transitions σ_i^l sont des marquages légaux. Cependant, tous les marquages atteignables depuis M_k (marquage atteignable après le franchissement de t_k) après le franchissement de la sous séquence de transitions σ_i^{in} sont des marquages interdits.

Définition 6 : Soit $\sigma_i = t_1, t_2, \dots, t_k$ une séquence de transitions que l'on peut considérer comme un mot du langage généré par le RdP à contrôler. Le préfixe de σ_i , noté $Pref(\sigma_i)$, est un mot de la forme t_1, \dots, t_j avec $0 \leq j \leq k$ et $k = |\sigma_i|$. Pour tout $q \in \mathbb{N}$: $Pref(\sigma_i)^{(q)}$ représente le préfixe de la séquence σ_i de longueur q .

Remarque 3 : Le langage généré par le RdP à contrôler est un langage fini et préfixe-clos car il est borné (i.e., le graphe de marquages est un ensemble fini).

En utilisant ces notions et définitions, les séquences de transitions interdites peuvent être obtenues en appliquant la procédure suivante :

Données : G le graphe de marquage du modèle RdP à contrôler, Ψ l'ensemble de transitions d'état interdites.

Objectif : calculer les séquences de transitions interdites.

Debut

1. Initialiser l'ensemble des séquences de transitions interdites $S = \emptyset$.
 2. Calculer S' l'ensemble de toutes les séquences de transitions légales et atteignables depuis le marquage initial M_0 .
 3. Calculer l'ensemble de transitions illégales qui succèdent chaque séquence σ_i dans S' noté $Suc(\sigma_i)$.
 4. Pour chaque transition t_i dans l'ensemble $Suc(\sigma_i)$ faire :
 - (a) Créer la séquence interdite $\sigma_j = \sigma_i t_i$.
 - (b) Déterminer l'ensemble des transitions qui succèdent σ_j .
 - (c) Pour chaque transition t_k de cet ensemble faire :
 - i. Si t_k est une transition interdite alors :
 - Créer la séquence $\sigma_k = \sigma_j t_k$:
 - Mettre à jour l'ensemble de séquences interdites $S := S \cup \{\sigma_j\}$.
 - $\sigma_j = \sigma_k$ et revenir à (b).
 - ii. Sinon, $S := S$.
- Fin Pour.

Fin Pour.

Fin.

3.5.2 Synthèse d'un contrôleur RdP : Algorithme 4

En se basant sur les définitions et les notions introduites précédemment, un algorithme qui consiste à ajouter des places de contrôle au modèle RdP initial afin d'empêcher l'atteignabilité des marquages interdits est proposé. Cet algorithme s'exécute selon les étapes suivantes :

1. Construction du graphe de marquages du modèle RdP à contrôler.
2. Identification des marquages interdits et détermination de l'ensemble de transitions d'état interdites équivalent.
3. Calcul des séquences de transitions interdites et les exploiter afin de caractériser les places de contrôle à ajouter.

Ainsi, le contrôleur RdP est synthétiser en utilisant l'algorithme 4.

Algorithme 4 : Algorithme de synthèse d'un contrôleur RdP (2)

Données : \mathcal{E} : l'ensemble d'états du système,
 (N, M_0) : le RdP à contrôler.

Debut

1. Générer G le graphe de marquages du modèle RdP à contrôler (N, M_0) .
2. Déterminer M_{fr} l'ensemble de marquages interdits et déduire Ψ l'ensemble de transitions d'état interdites comme expliqué dans les sous-sections 3.4.1 et 3.4.2.
3. Déterminer S l'ensemble des séquences de transitions interdites franchies depuis le marquage initial M_0 en parcourant le graphe de marquages du modèle RdP à contrôler comme expliqué dans §3.5.1.
4. Mettre à jour S l'ensemble de séquences de transitions interdites comme suit :
5. **pour** chaque séquence interdite $\sigma_i \in S$ **faire**
 1. Calculer l'ensemble de ses préfixes noté $Pref(\sigma_i)$.
 2. Comparer les préfixes de σ_i avec les préfixes de toutes les séquences de transitions dans S comme suit :
si il existe au moins une séquence de transitions σ_j telle que les éléments de l'ensemble de préfixes de σ_i sont inclus dans l'ensemble de préfixes de σ_j ($Pref(\sigma_i) \subset Pref(\sigma_j)$) **alors**
Mettre à jour S en éliminant σ_i ($S := S \setminus \{\sigma_i\}$).
sinon
 $\sqsubset S := S$.
6. **pour** chaque séquence de transitions σ_i dans S **faire**
 - 6.1. Déterminer la sous-séquence de transitions légale σ_i^l et la sous-séquence de transitions interdite σ_i^{in} .
 - 6.2. **pour** chaque sous-séquence de transitions interdite σ_i^{in} **faire**
 - 6.2.1. Déterminer $^*\sigma_i^{in}$ et σ_i^{in*} la première et la dernière transition de σ_i^{in} respectivement.
 - 6.2.2. Ajouter une place de contrôle p_{ci} au modèle RdP identifié en amont de $^*\sigma_i^{in}$ et en aval de σ_i^{in*} .
 - 6.2.3. Marquer la place de contrôle ajoutée par un marquage initial égal au nombre de fois que la transition interdite $^*\sigma_i^{in}$ est autorisée de franchir depuis le marquage initial et représenté mathématiquement par : $M_{c0}(p_{ci}) = |^*\sigma_i^{in} \cap \sigma_i^l|$.
 - 6.2.4. **si** il existe plusieurs sous-séquences interdites qui ont les même $^*\sigma_i^{in}$ et σ_i^{in*} **alors**
 - Ajouter une seule place de contrôle avec un marquage initial égal à $\max_{i, \sigma_i \in S} (|^*\sigma_i^{in} \cap \sigma_i^l|)$.

Fin.

Cet algorithme permet d'affiner le modèle RdP issu de l'identification par l'ajout des places de contrôle. Ces dernières sont caractérisées en utilisant les séquences de transitions interdites déduites à partir du graphe de marquages du modèle RdP à contrôler. Chaque séquence de transitions interdites dans l'ensemble final des séquences interdites S caractérise une place de contrôle à ajouter. Le nombre de places à ajouter n'est pas forcément égal au nombre de séquences interdites. En effet, plusieurs séquences interdites peuvent caractériser la même place de contrôle.

En outre, le RdP contrôlé peut accéder à tous les états représentant les comportements positifs du système sans atteindre ni des états de blocages ni des états dangereux. Ceci indique que le contrôleur est maximum permissif. Ainsi, pour évaluer la complexité de cet algorithme nous considérons m le nombre de places dans le RdP à contrôler, n le nombre de transitions et $|\sigma_i|$ la longueur de la plus grande séquence de transitions dans l'ensemble S dans le pire cas les inconnus sont $c = |\sigma_i|(n^2 + m \times n) + m + n$.

3.5.3 Exemple illustratif 2

Pour illustrer cet algorithme nous reprenons l'exemple précédent dont le graphe de marquage du RdP à contrôler est représenté par la figure 3.11.

L'application de la procédure introduite dans la sous-section 3.4.1 permet de conclure que les marquages interdits sont $M_{fr} = \{M_7, M_8, M_9\}$.

Par ailleurs, l'exécution de la procédure de la sous-section 3.4.2 permet d'obtenir l'ensemble de transitions d'état interdites suivant : $\Psi = \{(M_5 \xrightarrow{t_2} M_7), (M_7 \xrightarrow{t_3} M_8), (M_8 \xrightarrow{t_2} M_9), (M_9 \xrightarrow{t_4} M_3), (M_6 \xrightarrow{t_2} M_8), (M_8 \xrightarrow{t_4} M_2)\}$.

Le graphe de marquage permet d'obtenir l'ensemble S' contenant les séquences de transitions légales et franchissables depuis le marquage initial. Cet ensemble est : $S' = \{\sigma_1 = t_1 t_2 t_3 t_1, \sigma_2 = t_1 t_2 t_1 t_3, \sigma_3 = t_1 t_1 t_2, \sigma_4 = t_1 t_1 t_2 t_3, \sigma_5 = t_1 t_2 t_3 t_4\}$.

Ainsi, l'application de la procédure de détermination de l'ensemble des séquences de transitions interdites montre que $Suc(\sigma_1) = \{t_2\}$. Ceci permet de trouver la première séquence illégale $\sigma_1^1 = t_1 t_2 t_3 t_1 t_2$. Les transitions qui succèdent cette dernière sont : $Suc(\sigma_1^1) = \{t_3, t_4\}$ et permettent d'avoir les séquences interdites $\sigma_1^2 = \sigma_1^1 t_3 = t_1 t_2 t_3 t_1 t_2 t_3$ et $\sigma_1^3 = \sigma_1^1 t_4 = t_1 t_2 t_3 t_1 t_2 t_4$. De la même façon, on trouve que $Suc(\sigma_1^2) = \{t_4\}$ et $Suc(\sigma_1^3) = \{t_3\}$. Ces dernières permettent de trouver une seule séquence interdite donnée par : $\sigma_1^4 = \sigma_1^2 t_4 = t_1 t_2 t_3 t_1 t_2 t_3 t_4$. Ainsi, l'ensemble final des séquences interdites que l'on obtient à partir de σ_1 est $\delta_{11} = \{\sigma_1^1, \sigma_1^2, \sigma_1^3, \sigma_1^4\}$. Notons que, la procédure a été arrêtée à ce niveau là parce que les transitions qui succèdent aux séquences

σ_1^3 et σ_1^4 sont légales.

L'application de la même procédure pour les autres séquences de S' permet de donner les résultats suivant :

L'ensemble de séquences interdites obtenues à partir de σ_2 est : $\delta_{21} = \{\sigma_2^1 = t_1t_2t_1t_3t_2, \sigma_2^2 = t_1t_2t_1t_3t_2t_3, \sigma_2^3 = t_1t_2t_1t_3t_2t_4, \sigma_2^4 = t_1t_2t_1t_3t_2t_3t_4\}$.

L'ensemble de séquences interdites obtenues à partir de σ_3 est : $\delta_{31} = \{\sigma_3^1 = t_1t_1t_2t_2, \sigma_3^2 = t_1t_1t_2t_2t_3, \sigma_3^3 = t_1t_1t_2t_2t_3t_4, \sigma_3^4 = t_1t_1t_2t_2t_3t_3, \sigma_3^5 = t_1t_1t_2t_2t_3t_3t_4\}$.

L'ensemble de séquences interdites obtenues à partir de σ_4 est : $\delta_{41} = \{\sigma_4^1 = t_1t_1t_2t_3t_2, \sigma_4^2 = t_1t_1t_2t_3t_2t_4, \sigma_4^3 = t_1t_1t_2t_3t_2t_3, \sigma_4^4 = t_1t_1t_2t_3t_2t_3t_4\}$.

Finalement, l'ensemble de toutes les séquences de transitions interdites est $S = \{\sigma_1^1, \sigma_1^2, \sigma_1^3, \sigma_1^4, \sigma_2^1, \sigma_2^2, \sigma_2^3, \sigma_2^4, \sigma_3^1, \sigma_3^2, \sigma_3^3, \sigma_3^4, \sigma_3^5, \sigma_4^1, \sigma_4^2, \sigma_4^3, \sigma_4^4\}$.

Après avoir déterminé l'ensemble S , nous devons calculer, pour chacune de ces séquences, l'ensemble de ses préfixes. Ainsi, en prenant l'exemple des séquences σ_1^1 et σ_1^3 , nous obtenons :

$$Pref(\sigma_1^1) = \{\epsilon, t_1, t_1t_2, t_1t_2t_3, t_1t_2t_3t_1, t_1t_2t_3t_1t_2\},$$

$$Pref(\sigma_1^3) = \{\epsilon, t_1, t_1t_2, t_1t_2t_3, t_1t_2t_3t_1, t_1t_2t_3t_1t_2, t_1t_2t_3t_1t_2t_4\}.$$

L'analyse de ces deux ensemble montre que $Pref(\sigma_1^1) \subset Pref(\sigma_1^3)$. Ceci permet d'éliminer σ_1^1 de l'ensemble des séquences interdites ($S := S \setminus \{\sigma_1^1\}$).

L'application de la même procédure pour les autres séquences donne l'ensemble final des séquences interdites $S = \{\sigma_1^3, \sigma_1^4, \sigma_2^3, \sigma_2^4, \sigma_3^3, \sigma_3^5, \sigma_4^2, \sigma_4^4\}$.

Pour chaque séquence de S , nous devons maintenant déterminer ses sous séquences de transitions légales et interdites. Ainsi, en considérant par exemple la séquence $\sigma_3^5 = t_1t_1t_2t_2t_3t_3t_4$, nous obtenons une sous séquence légale $\sigma_3^{5l} = t_1t_1t_2$ et une sous séquence interdite $\sigma_3^{5in} = t_2t_3t_3t_4$. Cette dernière est caractérisée par : $^*\sigma_3^{5in} = t_2$ et $\sigma_3^{5in*} = t_4$.

Finalement, l'analyse des séquences interdites restantes montre que ces dernières sont aussi caractérisées par : $^*\sigma_i^{in} = t_2$ et $\sigma_i^{in*} = t_4$. Ainsi, le contrôleur RdP se résume, dans ce cas, en une place notée p_{c1} . Cette dernière est en amont de t_2 et en aval de t_4 comme l'indique la figure 3.14. Quant à son marquage initial, il est égale à 1 ($M_{oc}(p_{c1}) = \max(|^*\sigma_i^{in} \cap \sigma_i^l|) = 1$) car la transition $^*\sigma_3^{5in} = t_2$ est autorisée de franchir une seule fois depuis le marquage initial.

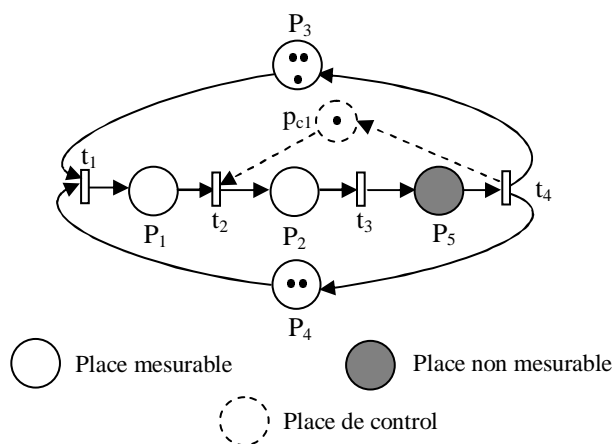


FIGURE 3.14 – Le modèle RdP contrôlé en appliquant l’algorithme 4

3.6 Conclusion

Dans ce chapitre nous avons commencé par une étude bibliographique concernant le problème de la commande supervisée. Nous nous sommes focalisés sur les travaux qui s’intéressent au problème d’états interdits du fait qu’il est le problème concerné dans cette partie de la thèse. Puis nous avons montré que, malgré que la littérature concernant la commande supervisée d’un SED par RdP soit abondante, les approches existantes ne peuvent pas être exploitées directement dans notre cas. Pour cette raison, nous avons proposé deux approches de synthèse d’un contrôleur RdP pour résoudre un problème d’états interdits. Ces approches permettent d’affiner le modèle RdP issu de la phase d’identification par l’ajout des places de contrôle.

Pour déterminer les places de contrôle à ajouter au modèle RdP à contrôler, la première approche proposée consiste à utiliser les chemins d’influence des transitions interdites. Chaque place de contrôle est caractérisée par au moins un chemin d’influence. Quant à la deuxième approche, elle consiste à exploiter les séquences de transitions interdites. Ces séquences représentent le langage interdit généré par ce RdP. Dans cette approche, les caractéristiques d’une place de contrôle sont déduites depuis aux moins une séquence de transitions interdites.

Ces deux approches ont l’avantage de nécessiter aucune spécification de contrôle sur le système considéré. Par ailleurs, les contrôleurs synthétisés sont maximum per-

messifs et permettent au modèle RdP issu de la phase d'identification de générer uniquement les comportements observés. La pertinence des approches proposées a été montrée à travers des exemples de simulation. Ainsi, dans le prochain chapitre, nous présentons les résultats de validation de ces approches sur une plateforme industrielle "La machine PRODUCTIS" .

Chapitre 4

Application : La machine "PRODUCTIS"

Résumé : L'objectif de ce chapitre est de tester la pertinence des algorithmes proposés dans le chapitre 2 et 3 sur un système réel : la machine PRODUCTIS. Il s'agit d'appliquer les deux méthodes d'identification proposées dans le deuxième chapitre de cette thèse pour élaborer le modèle RdP de ce système. Puis, utiliser les deux approches de contrôle introduites dans le troisième chapitre afin d'affiner ce modèle. Les résultats numériques obtenus permettent d'évaluer les performances en terme de temps de calcul et de la qualité de la solution.

4.1 Introduction

Dans les chapitres précédents nous avons présenté de nouvelles approches d'identification d'une classe des SED en utilisant les RdP. Ces approches sont appliquées, dans ce chapitre, sur un système réel : une machine de distribution de médicaments appelée "PRODUCTIS" installée à l'IUT de Reims. Nous commençons par présenter la machine avec ses différents postes en introduisant le rôle de chacun de ces postes. Puis, nous introduisons les données récupérées depuis les capteurs et les actionneurs qui équipent le système. Ensuite, nous appliquons les méthodes d'identification que nous avons proposées précédemment pour élaborer le modèle comportemental de ce système. Nous intégrons et nous analysons les différents résultats numériques obtenus. Finalement, nous clôturons par des conclusions.

4.2 Présentation de la machine "PRODUCTIS"

Le système réel sur lequel nous avons appliqué nos algorithmes, est un dispositif destiné à exploiter, gérer, maintenir, régler et piloter un système intégré de conditionnement de comprimés. Ce système est représenté par la figure 4.1.



FIGURE 4.1 – La machine PRODUCTIS

Cet équipement dispose de deux types de flacons : des petits et des grands. Chaque

type de flacons a un bouchon spécifique. Le flacon et le bouchon sont placés sur une palette adaptée à leurs dimensions. Ensuite, la palette est placée à l'entrée de la machine. Le déplacement de la palette d'un poste à un autre se fait par l'intermédiaire d'une chaîne à écailles. A chaque poste, la palette est arrêtée par une butée, qui doit être déverrouillée pour permettre à la palette de passer à l'inter-station puis au poste suivant. Chaque poste peut recevoir une seule palette à la fois. L'équipement est composé de cinq postes comme indiqué par la figure 4.2 :

- Poste 1 : distribution de comprimés blancs par comptage.
- Poste 2 : fermeture des flacons par bouchonnage, et évacuation.
- Poste 3 : distribution de comprimés verts par comptage.
- Poste 4 : fermeture des flacons par bouchonnage, et sans évacuation des produits.
- Poste 5 : chargement des palettes.

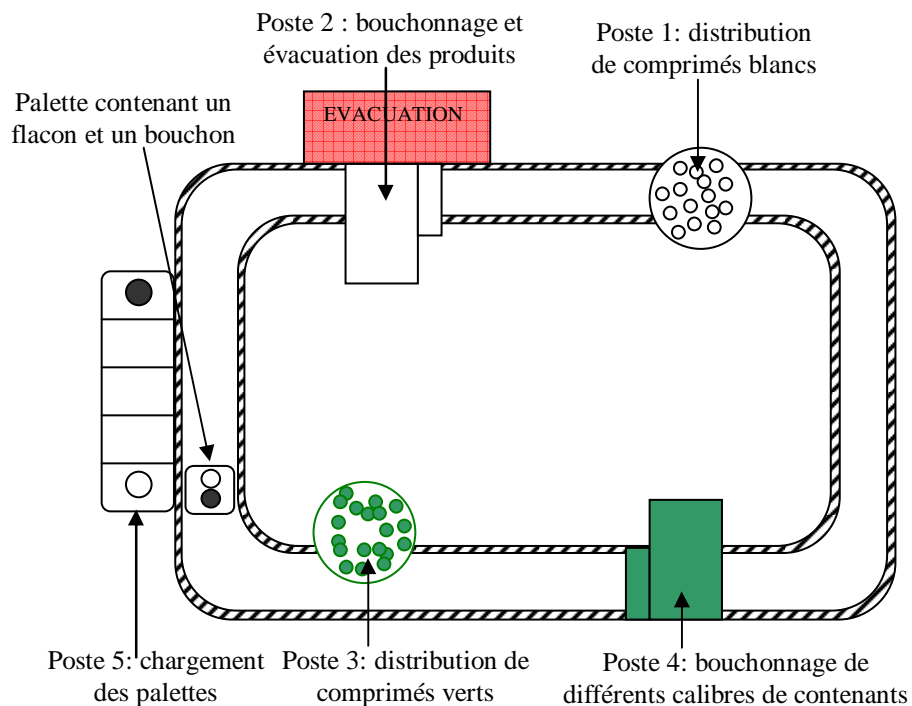


FIGURE 4.2 – Les postes de la machine

4.2.1 Poste 1 : Distribution de comprimés blancs par comptage

Ce poste, comme montré par la figure 4.3, est constitué d'une trémie transparente pour le stockage des comprimés, avec deux disques séparateurs pour leur distribution,

un détecteur photoélectrique pour contrôler le niveau de comprimés dans la trémie, deux vérins double effet pour positionner les disques et assurer la distribution des comprimés et d'un doigt d'arrêt pour stopper la palette sur le convoyeur. Pour ce poste, nous disposons d'un capteur permettant de contrôler la présence d'une palette (avec un flacon et un bouchon) avant de verser les comprimés. Pour libérer une capsule on actionne un vérin rotatif bistable et nous fixons le nombre de capsules à verser. Avant d'actionner le vérin, on doit vérifier d'abord qu'il y a bien une palette avec un flacon et que celui-ci n'est pas bouchonné. Tant que le nombre de capsules n'est pas tombé ou que le compteur fait défaut, on recommence le versement des capsules. Une fois que le compteur est à zéro, nous avons la quantité de capsules demandées.

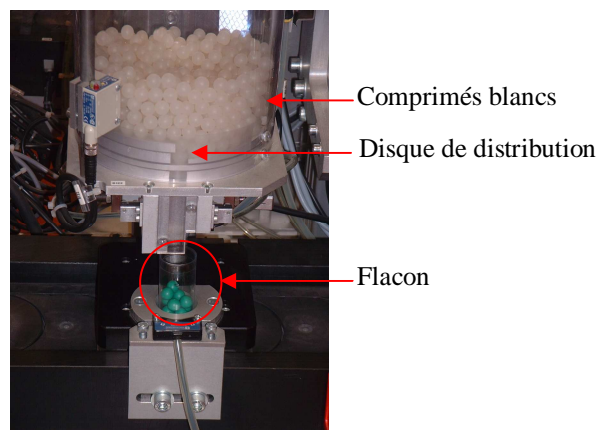


FIGURE 4.3 – Poste 1 : distribution de comprimés blancs

4.2.2 Poste 2 : Fermeture des gros flacons par bouchonnage

Ce poste comprend : un doigt d'arrêt pour stopper la palette sur la boucle, pour le bouchonnage des tubes, un manipulateur pneumatique en portique sur deux axes, horizontal et vertical, avec quatre capteurs de contrôle de position sur le manipulateur, et une ventouse équipée d'un Venturi. Avant d'effectuer l'opération de bouchonnage, il faut vérifier qu'il y a bien une palette avec un flacon gros non bouchonné rempli de capsules et un bouchon. Ceci se fait à l'aide d'un capteur de présence au niveau du poste. Ce poste est représenté par la figure 4.4.

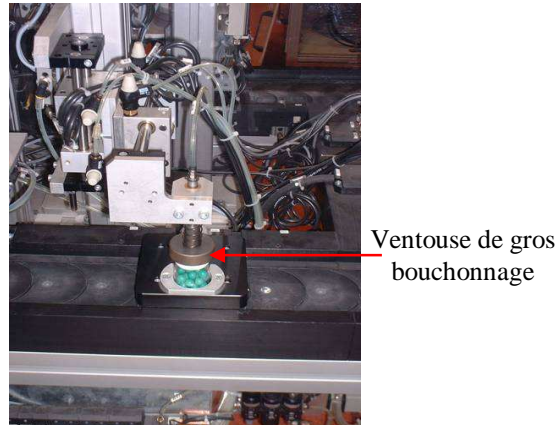


FIGURE 4.4 – Poste 2 : fermeture des gros flacons par bouchonnage

4.2.3 Poste 3 : distribution de comprimés verts par comptage

Similaire au poste 1, ce poste dispose d'un capteur permettant de contrôler la présence d'une palette (avec flacon non bouchonné et un bouchon) avant de verser les comprimés. Ce poste est représenté par la figure 4.5.

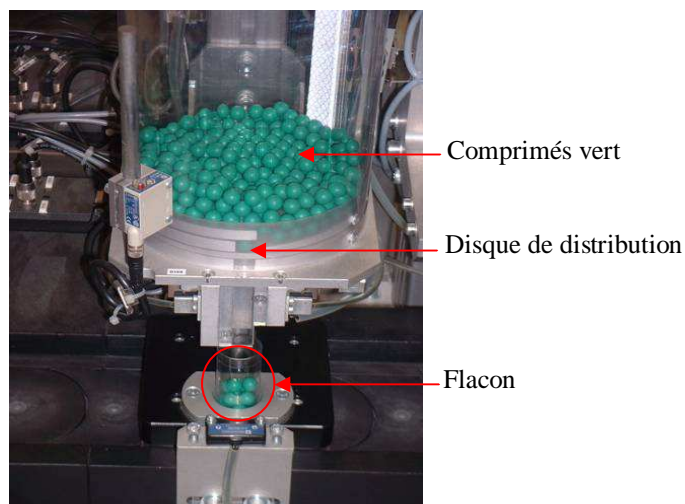


FIGURE 4.5 – Poste 3 : distribution de comprimés vert

4.2.4 Poste 4 : fermeture des petits flacons par bouchonnage

Ce poste comprend les mêmes composants que ceux du poste 2 mais destiné à la fermeture des petits flacons. Avant de commencer le bouchonnage des petits flacons, nous devons s'assurer qu'il existe une palette avec un petit flacon non bouchonné rempli de capsules et un bouchon. Ceci se fait à l'aide d'un capteur de présence au niveau du poste. La fermeture des flacons se fait à l'aide du manipulateur pneumatique. Il fallait d'abord descendre la tête du manipulateur afin d'aspirer le bouchon, puis remonter la tête et l'avancer au dessus du flacon, avant de redescendre et d'éjecter le bouchon pour fermer le flacon. Ce poste est représenté par la figure 4.6.

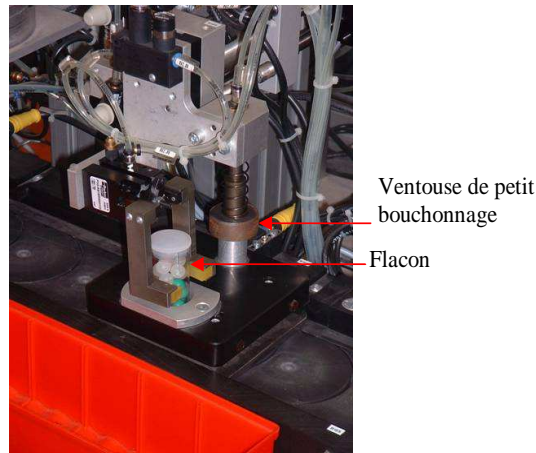


FIGURE 4.6 – Poste 4 : fermeture des petits flacons

4.2.5 Poste 5 : chargement des palettes

L'opération de chargements des palettes par des flacons et des bouchons est effectuée sur ce poste. Chaque palette est destinée à un seul type de flacons et doit contenir un seul flacon et un seul bouchon. Ce poste est représenté par la figure 4.7.



FIGURE 4.7 – Poste 4 : chargement des palettes

La machine PRODUCTIS est gérée par deux automates programmables "SCHNEIDER" : TSX Premium et TSX Micro. Les deux automates programmables sont en réseau unitelway¹. Un coupleur ethernet a été rajouté sur le TSX PREMIUM permettant de configurer l'adresse IP de l'automate pour communiquer entre la machine et le PC. Les données en provenance des deux APIs sont ainsi récupérées au moyen d'un serveur OPC à travers du protocole XIP (Xway sur IP) comme le montre la figure 4.8.

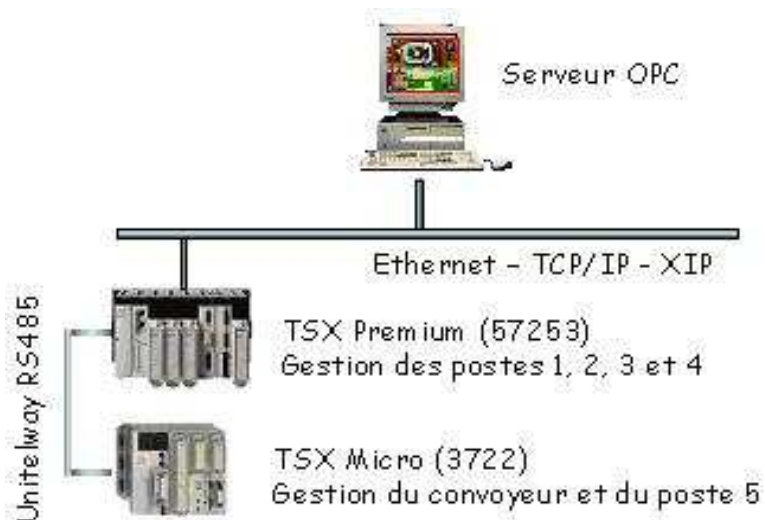


FIGURE 4.8 – Les automates reliés à la machine PRODUCTIS

1. Un protocole Unitelway étant essentiellement un protocole entre un API est éventuellement des PC

Notons que sur chaque station il existe un capteur de présence qui permet de détecter la présence de la palette sur la station. Aussi, le passage de la palette d'une station à une autre se fait après le verrouillage des barrières. Ces derniers sont considérés comme des actionneurs et les données d'identification liées aux capteurs et aux actionneurs équipant le système sont représentées par le tableau de la figure 1 de l'annexe.

Dans la section suivante, une modélisation boîte blanche du système à identifier est introduite. Ceci a pour raison de comparer les résultats de nos approches avec les résultats d'une modélisation boîte blanche afin de les valider.

4.3 Description et modélisation du système à identifier : approche boîte blanche

Le système est initialement au repos et dès l'activation du bouton de démarrage une palette arrive au niveau de la station de chargement. Cette dernière charge la palette avec un flacon et un bouchon et la libère, après le déverrouillage de la butée. Une fois libérée, la palette passe à l'interstation, qui se trouve entre la station de chargement et celle de distribution de comprimés verts. Elle attend que cette dernière soit libre pour y accéder. Au niveau de la station, le flacon reçoit un nombre fixé de comprimés verts. Ensuite, la palette, libérée après le levé de la butée, arrive au niveau de l'interstation qui se trouve entre la station de distribution de comprimés verts et celle de distribution de comprimés blancs. Si la station de distribution de comprimés blancs est libre alors, la palette peut y accéder afin que le flacon reçoive un nombre fixé de comprimés blancs. Une fois les comprimés blancs versés, la palette passe à l'interstation qui se trouve entre la station de distribution de comprimés blancs et de bouchonnage et elle attend que cette dernière soit libre. Lorsque la station de bouchonnage est libre, elle effectue les opérations de bouchonnage du flacon et la libération de la palette.

Notons qu'une station peut recevoir une seule palette à la fois et que nous avons fixé le nombre de palettes dans ce système à 4.

Partant de la description précédente du système, une modélisation de type boîte blanche permet de donner le RdP de la figure 4.9. Ce RdP est constitué des places P_1 , P_3 , P_5 et P_7 qui représentent respectivement la station de chargement, la station de distribution de comprimés verts, la station de distribution de comprimés blancs et celle de bouchonnage et évacuation. Les marquages initiaux de ces places sont nuls, ceci indique que les quatre stations sont initialement libre.

Il est constitué aussi des places P_2 , P_4 , P_6 et P_8 qui représentent les interstations. En effet, la place P_2 représente l'interstation qui se trouve entre la station de charge-

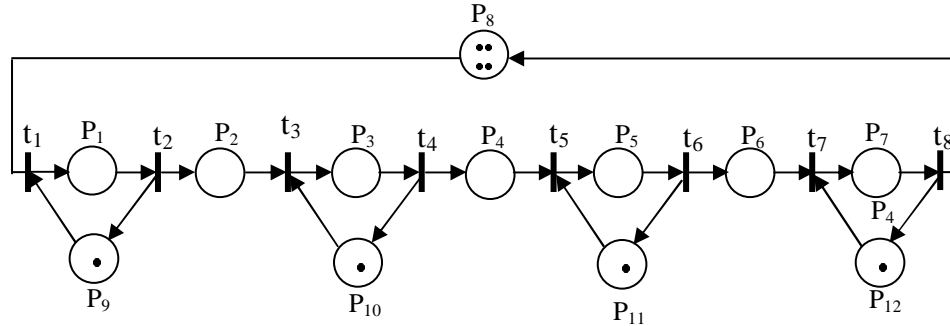


FIGURE 4.9 – Le modèle RdP du système obtenu après une modélisation boîte blanche

ment et celle de distribution de comprimés verts. La place P_4 représente l'interstation qui se trouve entre la station de distribution de comprimés verts et celle de distribution de comprimés blancs. La place P_6 représente l'interstation qui se trouve entre la station de distribution de comprimés blancs et celle de bouchonnage et évacuation. Enfin, la place P_8 représente l'interstation qui se trouve entre la station de bouchonnage et évacuation et celle de chargement.

Les places P_9 , P_{10} , P_{11} et P_{12} représentent l'état des différentes stations de la machine. L'existence d'une marque dans l'une de ces places indique que la station est libre et peut recevoir une palette.

Le RdP de la figure 4.9 est composé aussi de 8 transitions t_1 , t_2 , t_3 , t_4 , t_5 , t_6 , t_7 et t_8 . Ces transitions représentent les passages des palettes entre les stations et les interstations suite au verrouillage et déverrouillage des butées.

Dans ce qui suit nous appliquons nos méthodes d'identification proposées dans le chapitre 2 ([BME⁺06a], [BME⁺06b], [BME⁺07]), pour obtenir le modèle RdP de ce système. Notons que, ce travail a nécessité l'utilisation du langage de programmation C++ avec Cplex Cplex comme étant un solveur des programmes linéaires..

4.4 Application de la première méthode d'identification

En analysant les données d'identification liées aux 4 capteurs, qui se trouvent sur les quatres stations de la machine, nous remarquons qu'à partir de la ligne 28 jusqu'à la fin, on a des données répétitives. Donc, nous se contentons des données enregistrées du début jusqu'à la ligne 28 comme le représente la figure 4.10 avec : t_{reel} est le temps d'enregistrement des données, O_i avec $i = 1, \dots, 4$ sont les signaux de sortie associés

aux capteurs .

N	t_{reel}	O_1	O_2	O_3	O_4
1	0	0	0	0	0
2	1	1	0	0	0
3	2	1	0	0	0
4	3	0	0	0	0
5	4	1	0	0	0
6	5	1	1	0	0
7	6	1	1	0	0
8	7	1	0	0	0
9	8	0	0	0	0
10	9	1	0	0	0
11	10	1	1	0	0
12	11	1	1	0	0
13	12	1	1	1	0
14	13	1	1	1	0
15	14	1	1	0	0
16	15	1	0	0	0
17	16	0	0	0	0
18	17	1	0	0	0
19	18	1	1	0	0
20	19	1	1	0	0
21	20	1	1	1	0
22	21	1	1	1	0
23	22	1	1	1	1
24	23	1	1	1	1
25	24	1	1	1	0
26	25	1	1	0	0
27	26	1	0	0	0
28	27	0	0	0	0

FIGURE 4.10 – Les données d'identification liées aux capteurs

Néanmoins, en analysant ces données, nous remarquons qu'il y a des redondances qui n'apportent pas plus d'informations sur le système que nous allons négliger. Ainsi, nous obtenons un tableau de données finales d'identification liées aux capteurs comme les représente le tableau de la figure 4.11 où on a associé à chaque vecteur de sortie l'état mesurable correspondant. Ces données représentent seulement des informations partielles sur les différents états du système.

Les vecteurs de sortie enregistrés depuis les capteurs représentent les différents

N	t_reel	O ₁	O ₂	O ₃	O ₄	Etats
1	0	0	0	0	0	E ₀
2	1	1	0	0	0	E ₁
3	3	0	0	0	0	E ₀
4	4	1	0	0	0	E ₁
5	5	1	1	0	0	E ₂
6	7	1	0	0	0	E ₁
7	8	0	0	0	0	E ₀
8	9	1	0	0	0	E ₁
9	10	1	1	0	0	E ₂
10	12	1	1	1	0	E ₃
11	14	1	1	0	0	E ₂
12	15	1	0	0	0	E ₁
13	16	0	0	0	0	E ₀
14	17	1	0	0	0	E ₁
15	18	1	1	0	0	E ₂
16	20	1	1	1	0	E ₃
17	22	1	1	1	1	E ₄
18	24	1	1	1	0	E ₃
19	25	1	1	0	0	E ₂
20	26	1	0	0	0	E ₁
21	27	0	0	0	0	E ₀

FIGURE 4.11 – Les données finales d'identification liées aux capteurs

états mesurables atteignables par le système durant son fonctionnement. Ces états permettent de détecter tous les événements de ce système. En effet, cet exemple correspond au cas où les états sont partiellement mesurables (pas de capteur au niveau des interstations). En associant à chaque état E_i un marquage mesurable M_i , la séquence de marquages associée à la séquence d'états mesurables atteignables par le système à identifier est : $\mathcal{M} = M_0M_1M_0M_1M_2M_1M_0M_1M_2M_3M_2M_1M_0M_1M_2M_3M_4M_3M_2M_1M_0$ avec $M_0 = [0\ 0\ 0\ 0]^T$, $M_1 = [1\ 0\ 0\ 0]^T$, $M_2 = [1\ 1\ 0\ 0]^T$, $M_3 = [1\ 1\ 1\ 0]^T$, $M_4 = [1\ 1\ 1\ 1]^T$. En analysant cette séquence de marquage, nous pouvons déduire le nombre de places mesurables dans le modèle RdP à identifier. On trouve qu'il est égal à 4 ce qui est équivalent à la dimension du vecteur de marquage. Ces places peuvent être appelées p_1 , p_2 , p_3 et p_4 respectivement.

Le passage du système d'un état à un autre se fait après l'occurrence d'un événement. Donc, en utilisant les vecteurs de sortie mesurables, la séquence d'événements suivante est détectée : $\mathcal{E} = e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}$. Ainsi, en analysant les vecteurs de marquages associés aux vecteurs d'états en amont et en aval de chaque événement, dans la séquence d'événements, nous pouvons déterminer la transition associée à chaque événement. Prenant comme exemple les événements e_1, e_3, e_7 et e_{13} où le vecteur de marquage associé au vecteur d'états en amont de ces événements est $M_0 = [0 \ 0 \ 0 \ 0]^T$ et le vecteur de marquage associé au vecteur d'états en aval de ces événements est $M_1 = [1 \ 0 \ 0 \ 0]^T$. Donc, l'occurrence de ces événements provoque un changement dans l'état du système de $[0 \ 0 \ 0 \ 0]^T$ à $[1 \ 0 \ 0 \ 0]^T$ et ils peuvent être associés à la même transition t_1 . Ainsi, selon le même principe, nous associons aux événements e_2, e_6, e_{12} et e_{20} la transition t_2 , aux événements e_4, e_8 , et e_{14} la transition t_3 et aux événements e_5, e_{11} et e_{19} la transition t_4 , aux événements e_9 et e_{15} la transition t_5 , aux événements e_{10} et e_{18} la transition t_6 , à l'événement e_{16} la transition t_7 et à l'événement e_{17} la transition t_8 . Par conséquent, la séquence de transitions détectée depuis la séquence des vecteurs d'états est $\sigma = t_1, t_2, t_1, t_3, t_4, t_2, t_1, t_3, t_5, t_6, t_4, t_2, t_1, t_3, t_5, t_7, t_8, t_6, t_4, t_2$. Donc, l'ensemble de transitions du modèle RdP à identifier est $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$ avec $n = 8$ est le cardinal de cet ensemble.

En utilisant l'ensemble de transitions déduit T , l'ensemble de places P et la séquence de marquages mesurables \mathcal{M} , nous devons calculer la structure du RdP à identifier et le marquage initial des places non mesurables. Pour ce faire, nous définissons ce problème sous la forme d'un problème de programmation linéaire binaire et nous le résolvons en utilisant C++ et Cplex. Notons que, le nombre de places non mesurables dans le modèle à identifier est initialisé à 1 et il est incrémenté progressivement jusqu'à ce que l'on trouve une solution optimale. A chaque fois, nous comparons les comportements générés par le modèle identifié par ceux du système réel pour valider le modèle obtenu.

L'application de la procédure d'identification relative au théorème 2 du chapitre 2 et en considérant que le RdP à identifier possède une seule place non mesurable permet d'obtenir le modèle de la figure 4.12. Ce dernier est obtenu en satisfaisant la fonction objectif représentée par l'équation (2.8) du chapitre 2 et en vérifiant les contraintes (2.10), (2.12), (2.13), (2.14) et (2.24) du même chapitre.

Rappelons que, les contraintes de l'équation (2.10) portent respectivement sur : (1) la limitation du type de RdP à établir, (2) la vérification de la propriété de la connexité de ce RdP, (3) la vérification de l'équation de marquage et la satisfaction des conditions de franchissement d'une transition depuis le marquage initial. Ainsi, les contraintes décrites par les équations (2.11), (2.12), (2.13), (2.14) et (2.24) permettent d'interdire

4.4. Application de la première méthode d'identification

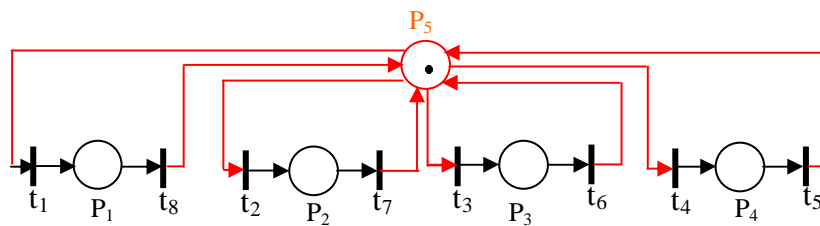
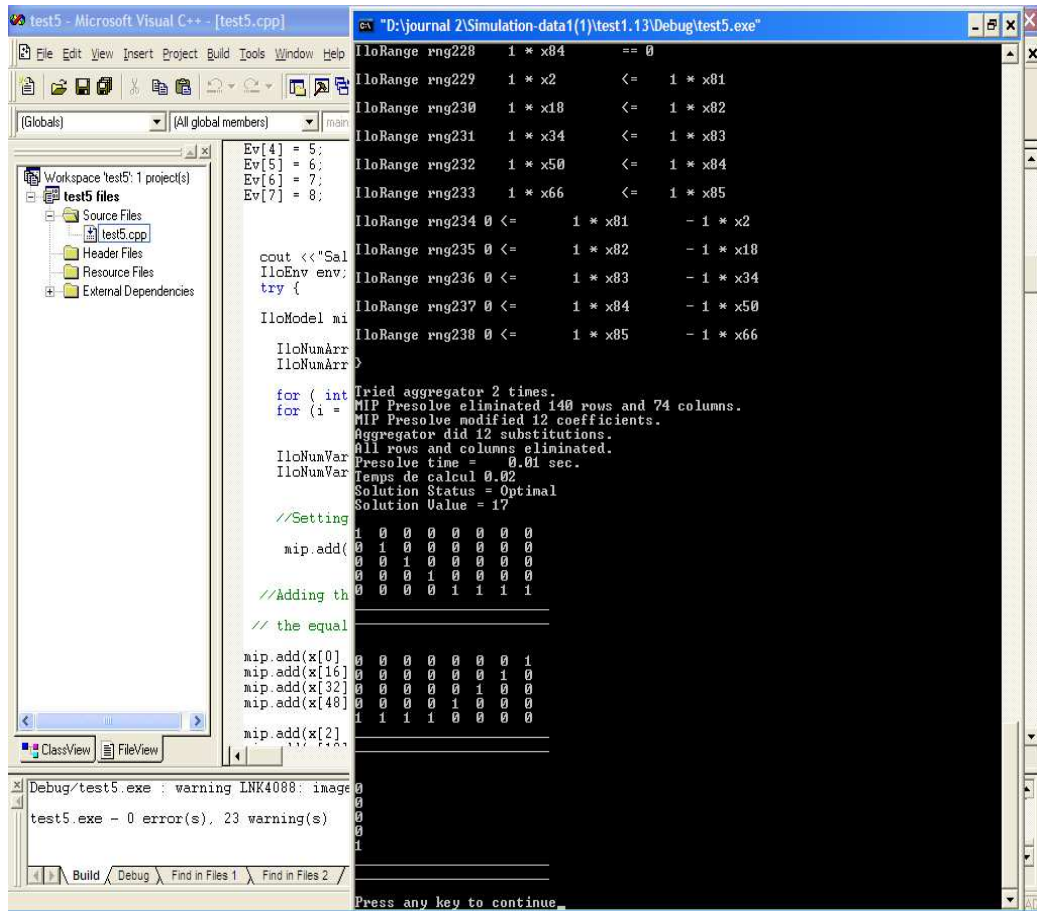


FIGURE 4.12 – Une solution optimale avec une place non mesurable

l'existence des transitions (places) puits (sources) et de caractériser l'existence d'un T-invariant dans le modèle RdP à identifier. Cependant, ce modèle RdP ne permet pas de générer tous les états atteignables par le système réel. Donc, nous ne pouvons pas le considérer comme modèle de ce système.

Notons que, la même solution est obtenue en vérifiant les contraintes (2.10), (2.11),

(2.12), (2.13) et (2.14) mais avec un temps de calcul plus grand.

En incrémentant le nombre de places non mesurables à 2 puis à 3, nous obtenons les RdP représentés par les figures 4.13 et 4.14. Ces deux modèles sont obtenus en vérifiant les contraintes (2.8), (2.10), (2.11), (2.12), (2.13) et (2.14). Cependant, ces modèles ne peuvent pas générer tous les comportements du système considéré. Donc, nous ne pouvons pas les considérer comme modèles de ce système.

Notons que, la vérification de la contrainte (2.24), dans ce cas, permet de donner les

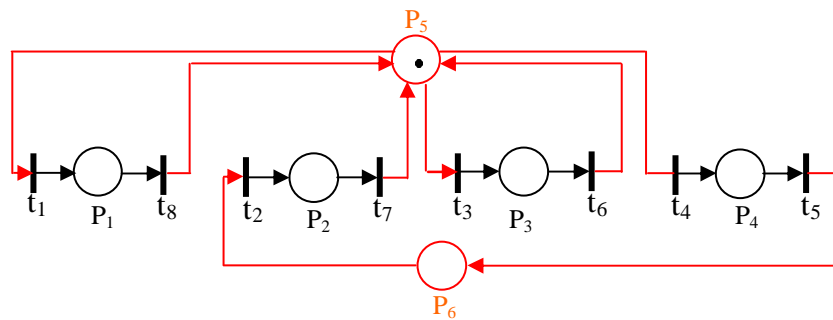


FIGURE 4.13 – Une solution optimale avec deux places non mesurables

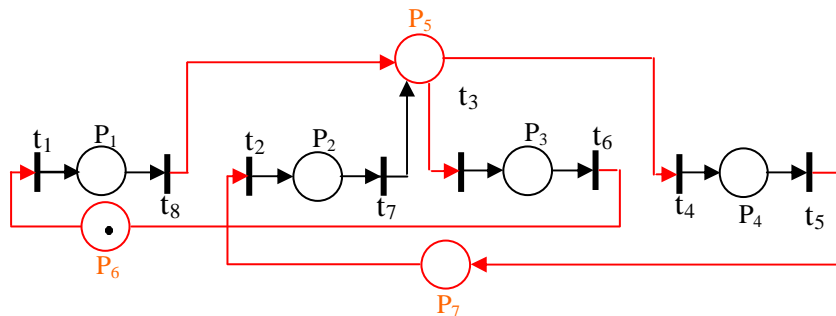


FIGURE 4.14 – Une solution optimale avec trois places non mesurables

mêmes solutions mais avec un temps de calcul plus grand et une contrainte de plus à vérifier.

En incrémentant le nombre de places non mesurables à 4, nous obtenons le modèle RdP de la figure 4.15. Ce dernier est établi en satisfaisant la fonction objectif représentée par l'équation (2.8) et en vérifiant les contraintes (b) : (2.10), (2.12), (2.13) et (2.14). Néanmoins, ce modèle permet de reproduire que quelques comportements du système réel donc, nous ne pouvons pas le considérer comme le bon modèle de ce système.

Notons que la vérification des contraintes (2.11) et (2.24), dans ce cas, permet d'avoir

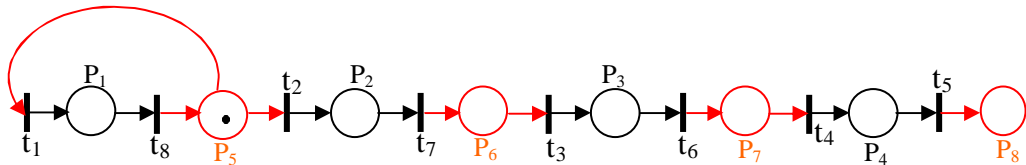


FIGURE 4.15 – Une solution optimale avec quatre places non mesurables

la même solution mais avec un temps de calcul plus grand que celui de la solution obtenue sans vérifié ces contraintes.

Avec cinq, six et sept places non mesurables nous avons obtenu des solutions optimales pour ce problème. Néanmoins, elles ne peuvent pas générer tous les comportements du système réel. Enfin, pour un modèle identifié avec huit places non mesurables, une solution optimale est obtenue comme montré sur la figure 4.16. Cette dernière est obtenue en satisfaisant (2.8) et en vérifiant les contraintes (c) : (2.10), (2.11), (2.12), (2.13), (2.14) et (2.24). L'analyse des comportements de ce RdP montre que ce dernier génère

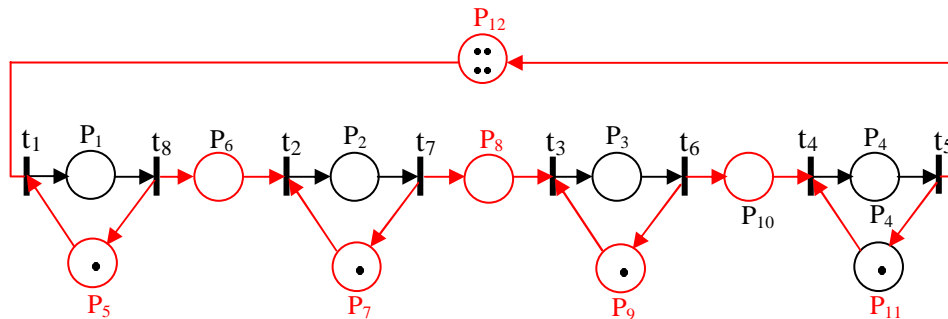


FIGURE 4.16 – Une solution optimale avec huit places non mesurables

exactement tous les comportements observés du système considéré. Notons, enfin, que ce modèle est similaire à celui de la figure 4.9 et qui a été obtenu en appliquant une approche de modélisation boîte blanche.

Le tableau (4.1) et la figure 4.17 présentent quelques résultats numériques de simulation. Ces résultats concernent les solutions obtenues pour chaque nombre de places non mesurables. Ils indiquent aussi, les contraintes vérifiées pour obtenir ces solutions et le temps de calcul nécessaire.

L'analyse de ces résultats montre que le temps de calcul nécessaire pour avoir des solutions optimales est court. Nous remarquons que, pour avoir une solution optimale avec une place non mesurable le temps de calcul est de 0.02 s. Pour un modèle identifié avec

Nombre de places mesurables	Nombre de places non mesurables	Contraintes	Temps de calcul (s)
4	1	(2.8), (2.10), (2.11), (2.12), (2.13), (2.14), (2.24)	0.02
4	2	(2.8), (2.10), (2.11), (2.12), (2.13), (2.14)	0.03
4	3	(2.8), (2.10), (2.11), (2.12), (2.13), (2.14)	0.03
4	4	(2.8), (2.10), (2.11), (2.13), (2.14)	0.04
		(2.8), (2.10), (2.12), (2.13), (2.14)	0.04
4	5	(2.8), (2.10), (2.11), (2.13), (2.14)	0.05
		(2.8), (2.10), (2.12), (2.13), (2.14)	0.05
4	6	(2.8), (2.10), (2.11), (2.13), (2.14)	0.06
4	7	(2.8), (2.10), (2.11), (2.13), (2.14), (2.24)	0.05
4	8	(2.8), (2.10), (2.11), (2.12), (2.13), (2.14), (2.24)	0.03

TABLE 4.1 – Résultats de simulation en appliquant la première méthode d'identification.

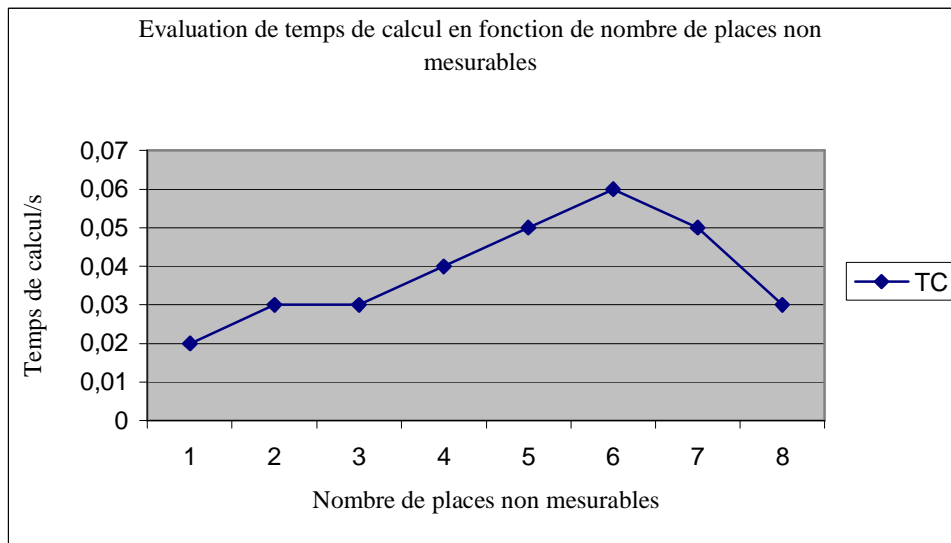


FIGURE 4.17 – Le temps de calcul nécessaire pour avoir une solution optimale en appliquant la première méthode d'identification

deux à trois places non mesurables le temps de calcul nécessaire est de 0.03s. Puis à partir de 4 jusqu'à 6 places non mesurables, le temps de calcul devient proportionnel au nombre de places non mesurables. Finalement, à partir de 7 places le temps de calcul baisse indiquant ainsi une convergence rapide vers une solution permettant de modé-

liser le système. On peut conclure que, plus que l'on est proche de l'optimum et avec un bon choix des contraintes, le temps de calcul nécessaire pour avoir une solution optimale devient raisonnable.

4.5 Application de la deuxième méthode d'identification

Cette méthode est basée sur l'exploitation des signaux d'entrée et de sortie mesurables liés aux capteurs et aux actionneurs équipant le système. En analysant les données enregistrées dans le tableau de la figure 1 de l'annexe, nous remarquons qu'à partir de la ligne 28 les mêmes informations qui se répètent. Donc, nous se contentons des données enregistrées depuis le début jusqu'à la ligne 28. Ces données sont représentées par le tableau de la figure 4.18 avec : t_{reel} est le temps d'enregistrement des données, O_i avec $i = 1, \dots, 4$ sont les signaux de sortie associés aux capteurs et I_i avec $i = 1, \dots, 4$ sont les signaux d'entrée associés aux actionneurs.

Ces données peuvent être représentées comme une succession d'états mesurables atteignables par ce système et d'événements mesurables franchies depuis ces états. Les différents états atteignables par ce système sont les suivants : $E_0 = [0 \ 0 \ 0 \ 0]^T$ représente l'état initial du système, $E_1 = [1 \ 0 \ 0 \ 0]^T$ indique la présence d'une palette sur la station de chargement, $E_2 = [1 \ 1 \ 0 \ 0]^T$ indique l'existence d'une palette sur la station de chargement et d'autre sur la station de distribution de premier type de médicaments, $E_3 = [1 \ 1 \ 1 \ 0]^T$ représente la présence d'une palette sur la station de chargement, une palette sur la station de distribution de premier type de médicaments et une autre sur la station de distribution du deuxième type de médicaments, et $E_4 = [1 \ 1 \ 1 \ 1]^T$, indique la présence d'une palette sur chaque station de l'équipement.

Une première analyse de ces données permet de conclure que le modèle RdP à identifier contient 4 places mesurables, ce qui équivaut à la dimension du vecteur d'état mesurable. Ces places sont appelées p_1, p_2, p_3 et p_4 respectivement. Elles représentent en réalité l'état de chaque station de l'équipement.

Ainsi, nous constatons que ce système permet de franchir 4 événements mesurables associés aux signaux d'entrée mesurables de ce système, notés e_1, e_2, e_3 et e_4 respectivement. L'événement e_1 est produit depuis l'état E_1 et conduit le système à l'état E_0 . L'événement e_2 est produit depuis E_2 et son occurrence conduit le système à l'état E_1 . L'occurrence de l'événement e_3 depuis l'état E_3 mène le système à l'état E_2 . L'événement e_4 est produit depuis E_4 et permet au système d'atteindre l'état E_3 . Ces événements seront associés aux transitions mesurables t_1, t_2, t_3 et t_4 respectivement,

N	t_reel	O ₁	O ₂	O ₃	O ₄	I ₁	I ₂	I ₃	I ₄
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0
3	2	1	0	0	0	1	0	0	0
4	3	0	0	0	0	0	0	0	0
5	4	1	0	0	0	0	0	0	0
6	5	1	1	0	0	0	0	0	0
7	6	1	1	0	0	0	1	0	0
8	7	1	0	0	0	1	0	0	0
9	8	0	0	0	0	0	0	0	0
10	9	1	0	0	0	0	0	0	0
11	10	1	1	0	0	0	0	0	0
12	11	1	1	0	0	0	0	0	0
13	12	1	1	1	0	0	0	0	0
14	13	1	1	1	0	0	0	1	0
15	14	1	1	0	0	0	1	0	0
16	15	1	0	0	0	1	0	0	0
17	16	0	0	0	0	0	0	0	0
18	17	1	0	0	0	0	0	0	0
19	18	1	1	0	0	0	0	0	0
20	19	1	1	0	0	0	0	0	0
21	20	1	1	1	0	0	0	0	0
22	21	1	1	1	0	0	0	0	0
23	22	1	1	1	1	0	0	0	0
24	23	1	1	1	1	0	0	0	1
25	24	1	1	1	0	0	0	1	0
26	25	1	1	0	0	0	1	0	0
27	26	1	0	0	0	1	0	0	0
28	27	0	0	0	0	0	0	0	0

FIGURE 4.18 – Les données d'identification finales liées aux capteurs et aux actionneurs

qui représentent les transitions du modèle RdP à identifier.

En analysant aussi les états mesurables atteignables par ce système et les événements produits depuis ces états nous remarquons que le système passe de E_0 à E_1 sans l'occurrence d'aucun événement mesurable. Donc, nous pouvons conclure que ce changement d'état est dû à l'occurrence d'un événement non mesurable que nous pouvons appeler e_5 . Cet événement sera associé à une transition non mesurable t_5 .

Ainsi, nous remarquons que le système change d'états de E_1 à E_2 sans l'occur-

rence d'aucun événement mesurable. Alors, nous pouvons dire que ce changement est dû à l'occurrence d'un événement non mesurable appelé e_6 . Ce dernier sera associé en terme de RdP à une transition non mesurable t_6 .

De plus, nous remarquons que le système passe de l'état E_2 à l'état E_3 sans la détection d'aucun événement mesurable. Donc, ce changement est dû à un événement non mesurable noté e_7 qui sera associé à une transition non mesurable t_7 .

Enfin, nous remarquons que le système change son état de E_3 à E_4 sans l'occurrence d'aucun événement mesurable. En effet, ce changement est dû à l'occurrence d'un événement non mesurable appelé e_8 qui sera associé à une transition non mesurable t_8 . Finalement, nous pouvons conclure que le modèle RdP à identifier est constitué de 4 transitions non mesurables t_5, t_6, t_7 et t_8 .

Dans cette méthode, le nombre de places non mesurables dans le modèle RdP à identifier est initialisé à 1 et il est incrémenté progressivement jusqu'à ce que l'on trouve une solution optimale. Ainsi, en utilisant les données d'identification représentées dans le tableau de la figure 4.18, le nombre de places (transitions) mesurables (non mesurable), nous définissons et nous résolvons un problème de programmation linéaire binaire pour obtenir le modèle RdP de ce système. Les solutions obtenues sont indiquées dans ce qui suit.

L'application de la procédure d'identification relative au théorème 5 du chapitre 2 et en considérant que le RdP à identifier possède une place non mesurable permet d'obtenir le modèle RdP de la figure 4.19. Ce dernier est obtenu en vérifiant les contraintes (2.8), (2.10), (2.12), (2.13) et (2.14).

Cependant, ce modèle ne peut pas générer tous les comportements du système réel. Ainsi, les différents états générés par ce RdP sont : $[0 \ 0 \ 0 \ 0 \ 1]^T$, $[1 \ 0 \ 0 \ 0 \ 0]^T$, $[0 \ 1 \ 0 \ 0 \ 0]^T$, $[0 \ 0 \ 1 \ 0 \ 0]^T$, $[0 \ 0 \ 0 \ 1 \ 0]^T$. Donc, il ne peut en aucun cas représenter le système considéré.

Notons que la vérification des contraintes (2.11) et (2.24), dans ce cas, permet d'avoir la même solution mais dans un temps de calcul plus grand et avec un nombre de contraintes de plus à vérifier.

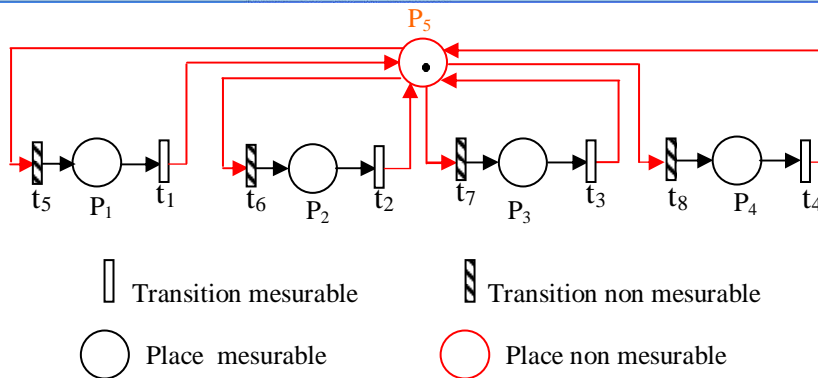
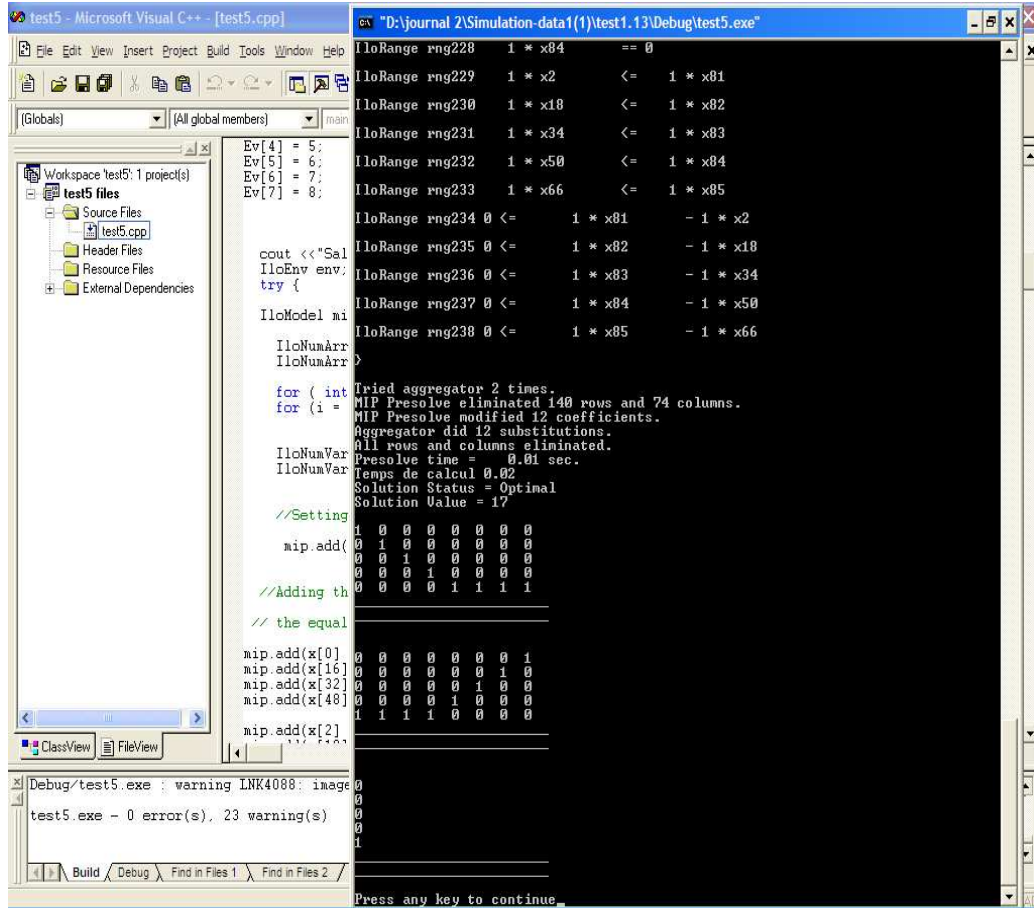


FIGURE 4.19 – Une solution optimale avec une place non mesurable

En incrémentant le nombre de places non mesurables à 2, nous obtenons le RdP de la figure 4.20. Ce dernier est établi en vérifiant les contraintes (2.8), (2.10), (2.12), (2.13) et (2.14).

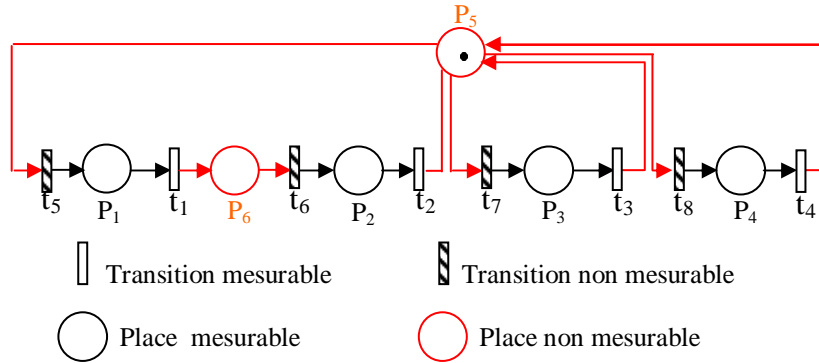


FIGURE 4.20 – Une solution optimale avec deux places non mesurables

Néanmoins, ce modèle ne peut pas atteindre tous les états atteignables par le système réel donc, nous ne pouvons pas le considérer comme modèle de ce système.

Notons que, la satisfaction des contraintes (2.11) et (2.24), dans ce cas, permet de donner la même solution mais dans un temps de calcul plus grand et avec un nombre de plus de contraintes à satisfaire.

L'incrémentación du nombre de places non mesurables à 3 permet de donner le modèle RdP de la figure 4.21. Ce dernier est établi en vérifiant les contraintes (2.8), (2.10), (2.12), (2.13), (2.14) et (2.24). Cependant, ce modèle ne peut pas être le bon modèle parce qu'il ne peut pas générer tous les états atteignables par le système considéré.

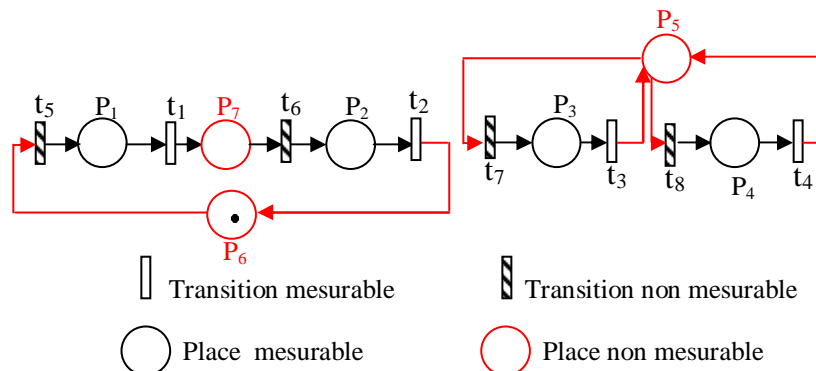


FIGURE 4.21 – Une solution optimale avec trois places non mesurables

Finalement, une solution optimale est obtenue pour quatre places non mesurables et elle est représentée par la figure 4.22. Notons que cette dernière est obtenue en vérifiant les contraintes (2.8), (2.10), (2.11), (2.12), (2.13), (2.14) et (2.24).

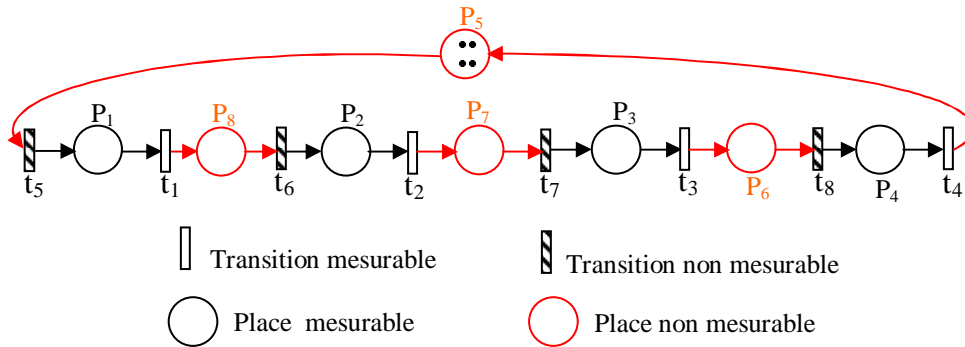


FIGURE 4.22 – Une solution optimale avec quatre places non mesurables

L'analyse des comportements de ce RdP montre qu'il génère tous les comportements observés du système. Cependant, il peut accéder à des états différents de ceux du système réel qui sont considérés comme des états interdits.

Le tableau (4.2) et la figure 4.23 présentent quelques résultats numériques de simulation. Ces résultats concernent les solutions obtenues pour chaque nombre de places non mesurables. Ils indiquent aussi, les contraintes vérifiées pour obtenir ces solutions et le temps de calcul nécessaire.

Nombre de places mesurables	Nombre de places non mesurables	Contraintes	Temps de calcul (s)
4	1	(2.8), (2.10), (2.12), (2.13), (2.14)	0.02
4	2	(2.8), (2.10), (2.11), (2.12), (2.13), (2.14)	0.03
4	3	(2.8), (2.10), (2.12), (2.13), (2.14), (2.24)	0.02
4	4	(2.8), (2.10), (2.11), (2.13), (2.14), (2.24)	0.02

TABLE 4.2 – Résultats de simulation en appliquant la deuxième méthode d'identification.

L'analyse de ces résultats montre que, pour avoir une solution optimale avec une et deux places non mesurables, le temps de calcul nécessaire augmente proportionnellement avec le nombre de places non mesurables. Puis, à partir de 3 places non mesurables, le temps de calcul baisse indiquant ainsi une convergence rapide vers une

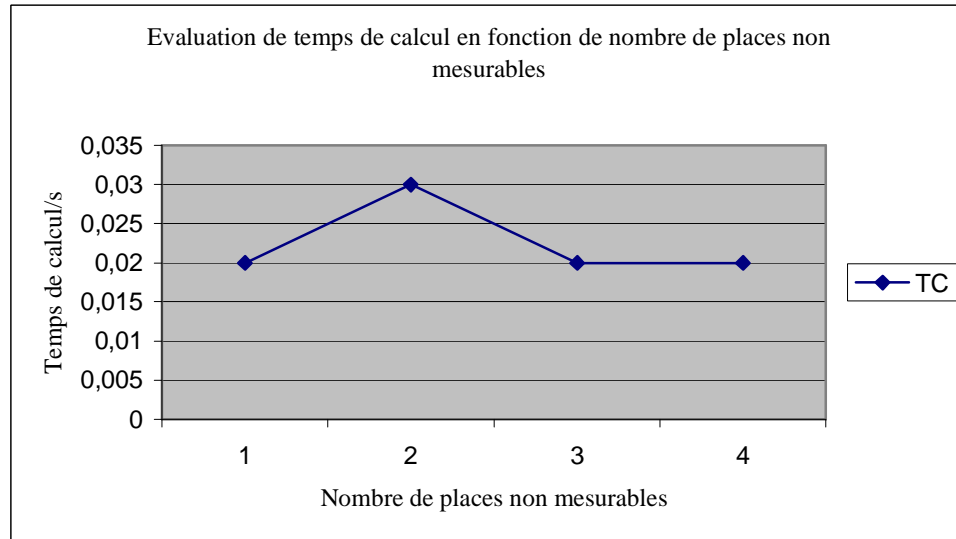


FIGURE 4.23 – Le temps de calcul nécessaire pour avoir une solution optimale en appliquant la deuxième méthode

solution permettant de modéliser le système considéré. Ceci permet de conclure que, malgré le nombre de paramètres à estimer le bon choix des contraintes joue un rôle important dans la convergence de la solution vers la valeur optimale dans un temps court. Ainsi, l'exploitation des signaux d'entrée et de sortie facilite la convergence de la méthode après un nombre d'itérations moins de celui de l'approche précédente.

Pour comparer les résultats obtenus en appliquant les deux approches d'identification, nous présentons le tableau (4.3).

Critères de comparaison	Approche 1	Approche 2
Temps de convergence	0.03s	0.02s
Convergence	Moyenne	Rapide
Données d'identifications	Etats	Etats/Evénements
Modèle	Complet	Complet + Comportements indésirables

TABLE 4.3 – Comparaison entre les deux approches d'identification.

Il est important de noter que, le modèle RdP élaboré en appliquant cette méthode d'identification peut, parfois, accéder à des états différents de ceux du système réel et qui sont considérés comme interdits. Pour inhiber l'atteignabilité de ces états nous appliquons nos algorithmes de contrôle développés dans le chapitre 3 ([BME⁺08a], [BME⁺08b], [BME⁺08c]).

4.5.1 Application du premier algorithme de synthèse d'un contrôleur RdP

Afin d'appliquer ces algorithmes, nous devons tout d'abord calculer le graphe de marquages du modèle RdP identifié. Un graphe de marquages partiel de ce RdP est représenté par la figure 4.24. Le graphe de marquages complet de ce RdP est représenté en mode texte par la figure 3 de l'annexe. Ainsi, les différents marquages atteignables par ce RdP sont représentés par le tableau de la figure 2 de l'annexe où les marquages indiqués en rouges sont ceux interdits.

L'application de la procédure de calcul des marquages interdits introduit dans le chapitre 3 (cf. §3.4.1) permet de donner l'ensemble de marquages interdits suivant : $M_{fr} = \{M_3, M_6, M_{11}, M_{14}, M_{19}, M_{22}, M_{69}, M_{89}, M_{111}, M_{103}, M_{132}, M_{154}, M_{176}, M_{244}, M_{264}, M_{281}, M_{294}, M_{302}, M_{304}, M_{326}, M_{328}, M_{329}\}$.

Ainsi, l'application de la procédure de calcul des transitions d'état interdites introduite dans le chapitre 3 (cf. §3.4.2) permet de donner l'ensemble suivant :

$$\Psi = \{(M_1 \xrightarrow{t_5} M_3), (M_3 \xrightarrow{t_5} M_6), (M_6 \xrightarrow{t_5} M_{11}), (M_{11} \xrightarrow{t_1} M_{14}), (M_{13} \xrightarrow{t_6} M_{22}), (M_{22} \xrightarrow{t_2} M_{32}), (M_{14} \xrightarrow{t_1} M_{19}), (M_{19} \xrightarrow{t_1} M_{27}), (M_{52} \xrightarrow{t_6} M_{69}), (M_{69} \xrightarrow{t_6} M_{89}), (M_{80} \xrightarrow{t_7} M_{103}), (M_{103} \xrightarrow{t_3} M_{126}), (M_{89} \xrightarrow{t_6} M_{111}), (M_{111} \xrightarrow{t_2} M_{132}), (M_{132} \xrightarrow{t_2} M_{154}), (M_{154} \xrightarrow{t_2} M_{177}), (M_{152} \xrightarrow{t_8} M_{176}), (M_{176} \xrightarrow{t_4} M_{37}), (M_{223} \xrightarrow{t_7} M_{244}), (M_{244} \xrightarrow{t_7} M_{264}), (M_{264} \xrightarrow{t_7} M_{281}), (M_{281} \xrightarrow{t_3} M_{294}), (M_{294} \xrightarrow{t_3} M_{304}), (M_{304} \xrightarrow{t_3} M_{312}), (M_{323} \xrightarrow{t_8} M_{326}), (M_{326} \xrightarrow{t_8} M_{328}), (M_{328} \xrightarrow{t_8} M_{329}), (M_{329} \xrightarrow{t_4} M_{302}), (M_{302} \xrightarrow{t_4} M_{176}), (M_{176} \xrightarrow{t_4} M_{37})\}.$$

Par ailleurs, les chemins d'influence de chaque transition interdite dans Ψ sont :

- Pour t_1 : $\pi_1^1 = t_1 M_{27}$, $\pi_1^2 = t_1 M_{19} t_1 M_{27}$, $\pi_1^3 = t_1 M_{14} t_1 M_{19} t_1 M_{27}$.
- Pour t_2 : $\pi_2^1 = t_2 M_{177}$, $\pi_2^2 = t_2 M_{154} t_2 M_{177}$, $\pi_2^3 = t_2 M_{132} t_2 M_{154} t_2 M_{177}$, $\pi_2^4 = t_2 M_{32}$.
- Pour t_3 : $\pi_3^1 = t_3 M_{312}$, $\pi_3^2 = t_3 M_{304} t_3 M_{312}$, $\pi_3^3 = t_3 M_{294} t_3 M_{304} t_3 M_{312}$, $\pi_3^4 = t_3 M_{126}$.
- Pour t_4 : $\pi_4^1 = t_4 M_{37}$, $\pi_4^2 = t_4 M_{176} t_4 M_{37}$, $\pi_4^3 = t_4 M_{302} t_4 M_{176} t_4 M_{37}$.
- Pour t_5 : $\pi_5^1 = t_5 M_{11} t_1 M_{14} t_1 M_{19} t_1 M_{27}$, $\pi_5^2 = t_5 M_6 t_5 M_{11} t_1 M_{14} t_1 M_{19} t_1 M_{27}$, $\pi_5^3 = t_5 M_3 t_5 M_6 t_5 M_{11} t_1 M_{14} t_1 M_{19} t_1 M_{27}$.
- Pour t_6 : $\pi_6^1 = t_6 M_{111} t_2 M_{132} t_2 M_{154} t_2 M_{177}$, $\pi_6^2 = t_6 M_{89} t_6 M_{111} t_2 M_{132} t_2 M_{154} t_2 M_{177}$, $\pi_6^3 = t_6 M_{69} t_6 M_{89} t_6 M_{111} t_2 M_{132} t_2 M_{154} t_2 M_{177}$, $\pi_6^4 = t_6 M_{22} t_2 M_{32}$.

4.5. Application de la deuxième méthode d'identification

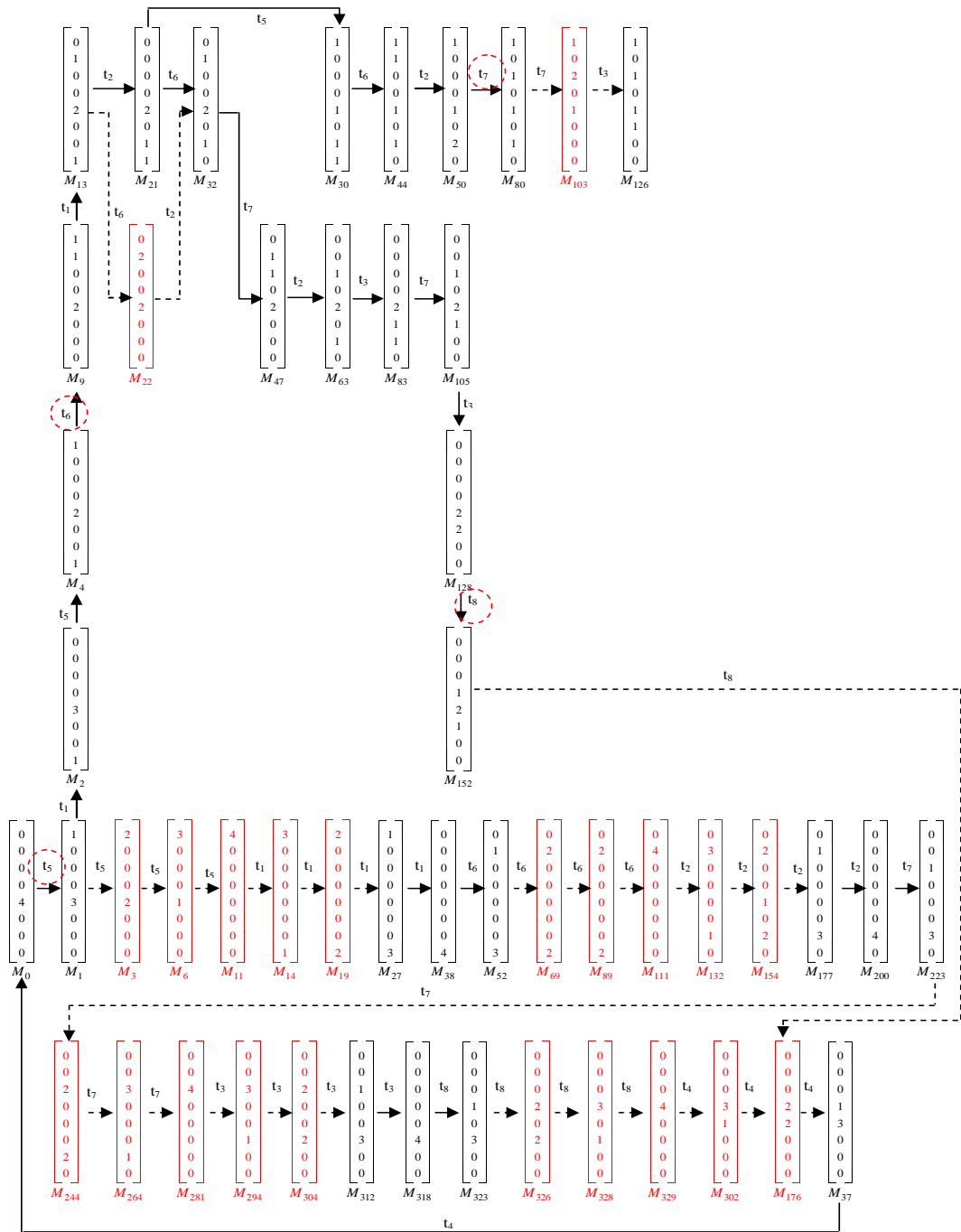


FIGURE 4.24 – Graphe de marquages partiel du modèle RdP identifié

○ Pour t_7 : $\pi_7^1 = t_7 M_{281} t_3 M_{294} t_3 M_{304} t_3 M_{312}$, $\pi_7^2 = t_7 M_{264} t_7 M_{281} t_3 M_{294} t_3 M_{304} t_3 M_{312}$, $\pi_7^3 = t_7 M_{244} t_7 M_{264} t_7 M_{281} t_3 M_{294} t_3 M_{304} t_3 M_{312}$, $\pi_7^4 = t_7 M_{103} t_3 M_{126}$.

○ Pour t_8 : $\pi_8^1 = t_8 M_{329} t_4 M_{302} t_4 M_{176} t_4 M_{37}$, $\pi_8^2 = t_8 M_{328} t_8 M_{329} t_4 M_{302} t_4 M_{176} t_4 M_{37}$, $\pi_8^3 = t_8 M_{326} t_8 M_{328} t_8 M_{329} t_4 M_{302} t_4 M_{176} t_4 M_{37}$, $\pi_8^4 = t_8 M_{176} t_4 M_{37}$.

Ainsi, la vérification de la relation d'inclusion entre ces chemins permet d'avoir l'ensemble final des chemins d'influence suivant :

$$\Pi' = \{ \pi_5^3 = t_5 M_3 t_5 M_6 t_5 M_{11} t_1 M_{14} t_1 M_{19} t_1 M_{27}, \pi_6^3 = t_6 M_{69} t_6 M_{89} t_6 M_{111} t_2 M_{132} t_2 M_{154} t_2 M_{177}, \pi_7^3 = t_7 M_{244} t_7 M_{264} t_7 M_{281} t_3 M_{294} t_3 M_{304} t_3 M_{312}, \pi_8^3 = t_8 M_{326} t_8 M_{328} t_8 M_{329} t_4 M_{302} t_4 M_{176} t_4 M_{37}, \pi_6^4 = t_6 M_{22} t_2 M_{32}, \pi_7^4 = t_7 M_{103} t_3 M_{126}, \pi_8^4 = t_8 M_{176} t_4 M_{37} \}.$$

Chaque chemin de l'ensemble Π' caractérise une place de contrôle à ajouter au modèle RdP identifié. En effet, π_5^3 permet de caractériser une place de contrôle p_{c1} en amont de t_5 et en aval de t_1 et de marquage initial égal à 1. Ce dernier est équivalent au nombre maximum de fois que la transition t_5 est autorisée de franchir depuis le marquage initial.

Par ailleurs, les chemins d'influence π_6^3 et π_6^4 caractérisent une autre place de contrôle p_{c2} à ajouter au modèle RdP. Cette place est en amont de t_6 et en aval de t_2 et elle est marquée aussi par un marquage initial $M_{c0}(p_{c2}) = 1$.

Ainsi, π_7^3 et π_7^4 caractérisent une autre place de contrôle p_{c3} à ajouter au modèle RdP. Cette place est en amont de t_7 et en aval de t_3 avec un marquage initial $M_{c0}(p_{c3}) = 1$.

De plus, π_8^3 et π_8^4 caractérisent une place de contrôle p_{c4} à ajouter au modèle RdP. Cette place est en amont de t_8 et en aval de t_4 avec un marquage initial $M_{c0}(p_{c4}) = 1$.

Finalement, le RdP identifié est affiné par l'ajout des places de contrôle p_{c1} , p_{c2} , p_{c3} et p_{c4} comme donné par la figure 4.25. Ce RdP génère exactement les comportements du système considéré. En comparant ce RdP avec celui obtenu par une modélisation boîte blanche représenté par la figure 4.9, on remarque qu'ils sont équivalents, ce qui nous permet de valider le RdP établi.

4.5.2 Application du deuxième algorithme de synthèse d'un contrôleur RdP

Pour appliquer cet algorithme, nous devons utiliser le graphe de marquages partiels du RdP identifié introduit dans la section (4.5.1). La première et la deuxième

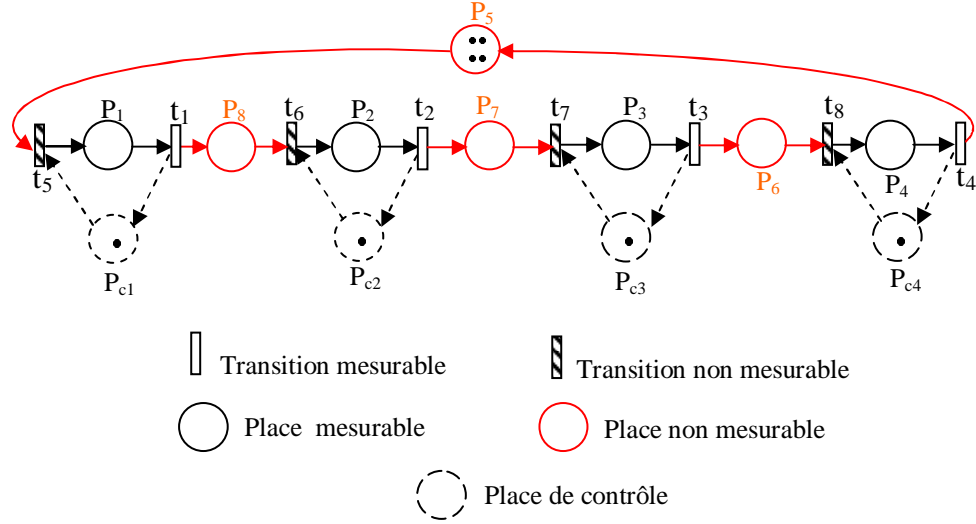


FIGURE 4.25 – Le RdP identifié final

étape de cet algorithme consiste à calculer l'ensemble de marquages interdits et l'ensemble de transitions d'états interdites comme indiqué dans l'algorithme précédent. Ensuite, en utilisant l'ensemble de transitions d'états interdites, nous calculons les différentes séquences de transitions interdites franchies depuis le marquage initial. Pour les déterminer, nous devons tout d'abord calculer S' l'ensemble de séquences de transitions légales franchies depuis le marquage initial. Ces séquences sont : $\sigma_1 = t_5$, $\sigma_2 = t_5 t_1 t_5 t_6 t_1$, $\sigma_3 = t_5 t_1 t_5 t_6 t_1 t_2 t_5 t_6 t_2 t_7$, $\sigma_4 = t_5 t_1 t_5 t_6 t_1 t_2 t_6 t_7 t_2 t_3 t_7 t_3 t_8$, donc $S' = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$.

L'application de la procédure de calcul des séquences de transitions interdites introduit dans le chapitre 3 (cf.§3.5.1) permet de donner les résultats suivants :

Les séquences de transitions interdites franchies depuis σ_1 sont :

$$\sigma_1^1 = t_5 t_5, \sigma_1^2 = t_5 t_5 t_5, \sigma_1^3 = t_5 t_5 t_5 t_5, \sigma_1^4 = t_5 t_5 t_5 t_5 t_1, \sigma_1^5 = t_5 t_5 t_5 t_5 t_1 t_1, \sigma_1^6 = t_5 t_5 t_5 t_5 t_1 t_1 t_1.$$

Les séquences de transitions interdites franchies depuis σ_2 sont : $\sigma_2^1 = t_5 t_1 t_5 t_6 t_1 t_6$, $\sigma_2^2 = t_5 t_1 t_5 t_6 t_1 t_6 t_2$.

Les séquences de transitions interdites franchies depuis σ_3 sont : $\sigma_3^1 = t_5 t_1 t_5 t_6 t_1 t_2 t_5 t_6 t_2 t_7 t_7$, $\sigma_3^2 = t_5 t_1 t_5 t_6 t_1 t_2 t_5 t_6 t_2 t_7 t_7 t_3$.

Les séquences de transitions interdites franchies depuis σ_4 sont : $\sigma_4^1 = t_5 t_1 t_5 t_6 t_1 t_2 t_6 t_7 t_2$

$$t_3t_7t_3t_8t_8, \sigma_4^2 = t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8t_8t_4.$$

Enfin, l'ensemble de toutes les séquences de transitions interdites est :

$$S = \{\sigma_1^1, \sigma_1^2, \sigma_1^3, \sigma_1^4, \sigma_1^5, \sigma_1^6, \sigma_2^1, \sigma_2^2, \sigma_3^1, \sigma_3^2, \sigma_4^1, \sigma_4^2\}.$$

Après avoir calculé S , nous le mettons à jour en calculant l'ensemble de préfixes de chaque séquence de transitions interdites dans cet ensemble et en vérifiant la relation d'inclusion entre ces ensembles comme suit :

$$\begin{aligned} Pref(\sigma_1^1) &= \{\varepsilon, t_5, t_5t_5\}, \\ Pref(\sigma_1^2) &= \{\varepsilon, t_5, t_5t_5, t_5t_5t_5\}, \\ Pref(\sigma_1^3) &= \{\varepsilon, t_5, t_5t_5, t_5t_5t_5, t_5t_5t_5t_5\}, \\ Pref(\sigma_1^4) &= \{\varepsilon, t_5, t_5t_5, t_5t_5t_5, t_5t_5t_5t_5, t_5t_5t_5t_5t_1\}, \\ Pref(\sigma_1^5) &= \{\varepsilon, t_5, t_5t_5, t_5t_5t_5, t_5t_5t_5t_5, t_5t_5t_5t_5t_1, t_5t_5t_5t_5t_1t_1\}, \\ Pref(\sigma_1^6) &= \{\varepsilon, t_5, t_5t_5, t_5t_5t_5, t_5t_5t_5t_5, t_5t_5t_5t_5t_1, t_5t_5t_5t_5t_1t_1, t_5t_5t_5t_5t_1t_1t_1\}. \end{aligned}$$

$$\begin{aligned} Pref(\sigma_2^1) &= \{\varepsilon, t_5, t_5t_1, t_5t_1t_5, t_5t_1t_5t_6, t_5t_1t_5t_6t_1, t_5t_1t_5t_6t_1t_6\}, \\ Pref(\sigma_2^2) &= \{\varepsilon, t_5, t_5t_1, t_5t_1t_5, t_5t_1t_5t_6, t_5t_1t_5t_6t_1, t_5t_1t_5t_6t_1t_6, t_5t_1t_5t_6t_1t_6t_2\}. \end{aligned}$$

$$\begin{aligned} Pref(\sigma_3^1) &= \{\varepsilon, t_5, t_5t_1, t_5t_1t_5, t_5t_1t_5t_6, t_5t_1t_5t_6t_1, t_5t_1t_5t_6t_1t_2, t_5t_1t_5t_6t_1t_2t_5, \\ & t_5t_1t_5t_6t_1t_2t_5t_6, t_5t_1t_5t_6t_1t_2t_5t_6t_2, t_5t_1t_5t_6t_1t_2t_5t_6t_2t_7, t_5t_1t_5t_6t_1t_2t_5t_6t_2t_7t_7\}. \\ Pref(\sigma_3^2) &= \{\varepsilon, t_5, t_5t_1, t_5t_1t_5, t_5t_1t_5t_6, t_5t_1t_5t_6t_1, t_5t_1t_5t_6t_1t_2, t_5t_1t_5t_6t_1t_2t_5, \\ & t_5t_1t_5t_6t_1t_2t_5t_6, t_5t_1t_5t_6t_1t_2t_5t_6t_2, t_5t_1t_5t_6t_1t_2t_5t_6t_2t_7, t_5t_1t_5t_6t_1t_2t_5t_6t_2t_7t_7, \\ & t_5t_1t_5t_6t_1t_2t_5t_6t_2t_7t_7t_3\}. \end{aligned}$$

$$\begin{aligned} Pref(\sigma_4^1) &= \{\varepsilon, t_5, t_5t_1, t_5t_1t_5, t_5t_1t_5t_6, t_5t_1t_5t_6t_1, t_5t_1t_5t_6t_1t_2, t_5t_1t_5t_6t_1t_2t_6, \\ & t_5t_1t_5t_6t_1t_2t_6, t_5t_1t_5t_6t_1t_2t_6t_7, t_5t_1t_5t_6t_1t_2t_6t_7t_2, t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3, \\ & t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8, t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8t_8\}. \\ Pref(\sigma_4^2) &= \{\varepsilon, t_5, t_5t_1, t_5t_1t_5, t_5t_1t_5t_6, t_5t_1t_5t_6t_1, t_5t_1t_5t_6t_1t_2, t_5t_1t_5t_6t_1t_2t_6, \\ & t_5t_1t_5t_6t_1t_2t_6, t_5t_1t_5t_6t_1t_2t_6t_7, t_5t_1t_5t_6t_1t_2t_6t_7t_2, t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3, \\ & t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8, t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8t_8, t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8t_8t_4\}. \end{aligned}$$

La vérification de la relation d'inclusion entre ces ensembles de préfixes montre que :

$$Pref(\sigma_1^1) \subset Pref(\sigma_1^2) \subset Pref(\sigma_1^3) \subset Pref(\sigma_1^4) \subset Pref(\sigma_1^5) \subset Pref(\sigma_1^6) \text{ donc : } S := S \setminus \{\sigma_1^1, \sigma_1^2, \sigma_1^3, \sigma_1^4, \sigma_1^5\}.$$

$$\text{Ainsi, nous remarquons que : } Pref(\sigma_2^1) \subset Pref(\sigma_2^2), Pref(\sigma_3^1) \subset Pref(\sigma_3^2), Pref(\sigma_4^1) \subset Pref(\sigma_4^2).$$

Enfin, l'ensemble final des séquences de transitions interdites est : $S = \{\sigma_1^6 = t_5t_5t_5t_5t_1t_1t_1, \sigma_2^2 = t_5t_1t_5t_6t_1t_6t_2, \sigma_3^2 = t_5t_1t_5t_6t_1t_2t_5t_6t_2t_7t_7t_3, \sigma_4^2 = t_5t_1t_5t_6t_1t_2t_6t_7t_2t_3t_7t_3t_8t_4\}$. Chacune de ces séquences caractérise une place de contrôle à ajouter au

modèle RdP identifié. Pour chacune de ces séquences nous déterminons ses sous séquences de transitions légales et interdites.

- Pour σ_1^6 : $\sigma_1^{6l} = t_5$ et $\sigma_1^{6in} = t_5 t_5 t_5 t_1 t_1 t_1$. Nous remarquons que la première transition de σ_1^{6in} est $^* \sigma_1^{6in} = t_5$ et la dernière transition est $\sigma_1^{6in*} = t_1$. Donc, cette séquence caractérise une place de contrôle p_{c1} en amont de t_5 et en aval de t_1 avec un marquage initial $M_{c0}(p_{c3}) = |t_5 \cap \sigma_1^{6l}| = 1$.
- Pour σ_2^2 : $\sigma_2^{2l} = t_5 t_1 t_5 t_6 t_1$ et $\sigma_2^{2in} = t_6 t_2$. Nous remarquons que $^* \sigma_2^{2in} = t_6$ et $\sigma_2^{2in*} = t_2$. Donc, cette séquence caractérise une place de contrôle p_{c2} en amont de t_6 et en aval de t_2 avec un marquage initial $M_{c0}(p_{c2}) = |t_6 \cap \sigma_2^{2l}| = 1$.
- Pour σ_3^2 : $\sigma_3^{2l} = t_5 t_1 t_5 t_6 t_1 t_2 t_5 t_6 t_2 t_7$ et $\sigma_3^{2in} = t_7 t_3$. Nous remarquons que $^* \sigma_3^{2in} = t_7$ et $\sigma_3^{2in*} = t_3$. Donc, cette séquence caractérise une place de contrôle p_{c3} en amont de t_7 et en aval de t_3 avec un marquage initial $M_{c0}(p_{c3}) = |t_7 \cap \sigma_3^{2l}| = 1$.
- Pour σ_4^2 : $\sigma_4^{2l} = t_5 t_1 t_5 t_6 t_1 t_2 t_6 t_7 t_2 t_3 t_7 t_3 t_8$ et $\sigma_4^{2in} = t_8 t_4$. Nous remarquons que $^* \sigma_4^{2in} = t_8$ et $\sigma_4^{2in*} = t_4$. Donc, cette séquence caractérise une place de contrôle p_{c4} en amont de t_8 et en aval de t_4 avec un marquage initial $M_{c0}(p_{c4}) = |t_8 \cap \sigma_4^{2l}| = 1$.

Finalement, l'application de cet algorithme permet de caractériser quatre places de contrôle p_{c1} , p_{c2} , p_{c3} et p_{c4} à ajouter au modèle RdP identifié comme ceux caractérisées en appliquant le premier algorithme de contrôle. Le RdP affiné par l'ajout de ces places de contrôle p_{c1} , p_{c2} , p_{c3} et p_{c4} est donné par la figure 4.25. Ce RdP génère exactement les comportements du système considéré. En comparant ce RdP avec celui obtenu par une modélisation boîte blanche représenté par la figure 4.9, on remarque qu'ils sont équivalents.

4.6 Conclusion

Dans ce chapitre, nous avons appliqué les algorithmes proposés dans le chapitre 2 et 3 sur une plateforme réel "PRODUCTIS". Notre objectif à travers cette application est d'obtenir le modèle comportemental de ce système en utilisant les données réelles récupérées durant son fonctionnement. Les résultats numériques obtenus montrent l'efficacité de nos algorithmes que ce soit pour le temps de calcul ou la convergence vers la solution optimale. Ils montrent aussi que le bon choix des contraintes joue un rôle très important dans la convergence vers l'optimal dans un temps très raisonnable et ceci malgré le nombre de paramètres à estimer.

D'après les différents tests effectués, nous pouvons constater que nos méthodes

d'identification proposées dans ce travail peuvent être appliquées sur des systèmes réels de grandes tailles. Cependant, lorsque le nombre de paramètres à estimer est assez grand par rapport aux paramètres mesurés, la convergence vers la solution optimale devient lente et l'obtention d'une solution optimale nécessite des informations complémentaires sur le système à identifier.

Conclusion générale

Effectuer un processus d'identification d'un Système à Evénements Discrets (SED) revient à établir un modèle capable de générer le comportement de ce système. Ce modèle est obtenu en utilisant les signaux d'entrée et/ou de sortie. Grâce à leurs propriétés mathématiques et graphiques, les réseaux de Petri (RdP) sont choisis comme formalisme de modélisation parfaitement adapté à la résolution du problème d'identification d'un SED.

Dans cette thèse, nous avons montré que le problème d'identification par RdP est relativement récent et que les approches d'identification à base de RdP utilisent des hypothèses restrictives sur le modèle à identifier et/ou sur les données d'identification. Ainsi, nous avons proposé deux approches qui visent à remédier à certains inconvénients.

La première approche considère que le vecteur d'état est composé de signaux de sortie et utilise les informations issues des capteurs pour établir un modèle du système. Dans ce cadre, les conditions nécessaires et suffisantes pour l'existence d'une solution unique sont synthétisées. Ensuite, le cas le plus général est considéré en formulant le problème d'identification sous la forme d'un problème d'optimisation combinatoire. Ce problème est, ensuite, résolu en utilisant la technique de la programmation linéaire binaire. L'analyse de la l'approche proposé montre qu'elle a l'avantage d'être simple, relative et constructive. Ainsi, le modèle identifié permet de reproduire exactement les comportements du système considéré. L'application de cette méthode sur un système réel montre aussi son efficacité en termes de temps de calcul et de l'optimalité de la solution. Néanmoins, bien que la complexité de l'approche proposé soit linéaire, la convergence vers une solution optimale est parfois difficile notamment lorsque le nombre de paramètres à estimer devient grand.

Pour améliorer la convergence vers la solution optimale, une deuxième approche d'identification d'une classe est proposée. Elle consiste à exploiter les signaux d'entrée et de sortie mesurables du système considéré. Cette deuxième approche considère que le modèle du système est composé d'une partie mesurable et d'une partie non mesurable. Elle propose d'exploiter les signaux d'E/S afin de déterminer des bornes sur le

nombre de places (transitions) nom mesurables. Une fois ces bornes sont obtenues, le problème d'identification est défini sous la forme d'un problème d'optimisation combinatoire et résolu par la technique de la programmation linéaire binaire. L'analyse de cette approche montre qu'elle est relative, constructive et d'une complexité linéaire. L'application de cette méthode sur un système réel montre aussi son efficacité en termes de temps de calcul et de convergence vers une solution optimale.

Bien que les méthodes proposées permettent d'identifier des modèles reproduisant les comportements observables dans la séquence exploitée pour l'identification, les modèles obtenus peuvent parfois générer des comportements non observés. Ces derniers sont considérés comme des comportements interdits que l'on doit interdire.

Pour ce faire, le problème d'états interdits a été traité comme un problème de transitions d'état interdites et deux algorithmes de synthèse d'un contrôleur RdP ont été proposés. Le premier algorithme est basé sur l'utilisation des chemins d'influence des transitions interdites. Quant au deuxième algorithme, il consiste à utiliser les séquences de transitions interdites pour caractériser les places de contrôle à ajouter au modèle RdP identifié. Comparativement, avec les travaux qui visent à résoudre ce problème, nos algorithmes possèdent l'avantage de ne nécessiter aucune spécification, et d'utiliser uniquement les états atteignables observés dans la séquence d'E/S, ce qui répond bien à un besoin d'identification.

En résumé, nous avons développé deux approches d'identification de SED capables de traiter une large classe de problèmes pour tout type de modèle RdP ordinaire pur. En effet, nos méthodes d'identification proposées dans ce travail peuvent être appliquées sur des systèmes réels de grande taille. Les résultats numériques obtenus montrent l'efficacité de nos approches que ce soit en termes du temps de calcul ou bien de convergence vers la solution optimale. Cependant, lorsque le nombre de paramètres à estimer est très grand, la convergence vers la solution optimale devient lente.

Ce travail de thèse est le premier qui porte sur l'identification des SED dans le laboratoire CReSTIC, de l'Université de Reims Champagne Ardenne. Il a permis d'explorer une piste de travail nouvelle et prometteuse et il nous a guidé vers des perspectives prometteuses. Parmi celles-ci, la proposition d'une borne inférieure sur le nombre d'arcs composant le modèle RdP. Cette borne sera mise à jour en utilisant une des méthodes propres au programmation linéaire en nombres entiers qui est la méthode de recherche arborescente par séparation et évaluation.

Par ailleurs, les exemples présentés montrent que le choix des contraintes est une étape cruciale permettant d'accélérer et de garantir la convergence vers une solution optimale. Ainsi, il serait très intéressant de proposer une méthodologie qui guide l'uti-

lisateur dans la formulation des "bonnes contraintes" lorsque l'on dispose d'un minimum de connaissances sur le système. D'autres travaux viseront d'exploiter les approches proposées pour l'identification et le diagnostic des Systèmes Dynamiques Hybrides.

Annexe

4.7 Les données d'identification

Ci-après, les données utilisées pour l'identification du système considéré sont introduites (cf figure (4.26)).

N	t_reel	O ₁	O ₂	O ₃	O ₄	I ₁	I ₂	I ₃	I ₄
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0
3	2	1	0	0	0	1	0	0	0
4	3	0	0	0	0	0	0	0	0
5	4	1	0	0	0	0	0	0	0
6	5	1	1	0	0	0	0	0	0
7	6	1	1	0	0	0	1	0	0
8	7	1	0	0	0	1	0	0	0
9	8	0	0	0	0	0	0	0	0
10	9	1	0	0	0	0	0	0	0
11	10	1	1	0	0	0	0	0	0
12	11	1	1	0	0	0	0	0	0
13	12	1	1	1	0	0	0	0	0
14	13	1	1	1	0	0	0	1	0
15	14	1	1	0	0	0	1	0	0
16	15	1	0	0	0	1	0	0	0
17	16	0	0	0	0	0	0	0	0
18	17	1	0	0	0	0	0	0	0
19	18	1	1	0	0	0	0	0	0
20	19	1	1	0	0	0	0	0	0
21	20	1	1	1	0	0	0	0	0
22	21	1	1	1	0	0	0	0	0
23	22	1	1	1	1	0	0	0	0
24	23	1	1	1	1	0	0	0	1

25	24	1	1	1	0	0	0	1	0
26	25	1	1	0	0	0	1	0	0
27	26	1	0	0	0	1	0	0	0
28	27	0	0	0	0	0	0	0	0
29	28	1	0	0	0	0	0	0	0
30	29	1	1	0	0	0	0	0	0
31	30	1	1	0	0	0	0	0	0
32	31	1	1	1	0	0	0	0	0
33	32	1	1	1	0	0	0	0	0
34	33	1	1	1	1	0	0	0	0
35	34	1	1	1	1	0	0	0	1
36	35	1	1	1	0	0	0	1	0
37	36	1	1	0	0	0	1	0	0
38	37	1	0	0	0	1	0	0	0
39	38	0	0	0	0	0	0	0	0
40	39	1	0	0	0	0	0	0	0
41	40	1	1	0	0	0	0	0	0
42	41	1	1	0	0	0	0	0	0
43	42	1	1	1	0	0	0	0	0
44	43	1	1	1	0	0	0	0	0
45	44	1	1	1	1	0	0	0	0
46	45	1	1	1	1	0	0	0	1
47	46	1	1	1	0	0	0	1	0
48	47	1	1	0	0	0	1	0	0
49	48	1	0	0	0	1	0	0	0
50	49	0	0	0	0	0	0	0	0
51	50	1	0	0	0	0	0	0	0
52	51	1	1	0	0	0	0	0	0
53	52	1	1	0	0	0	0	0	0
54	53	1	1	1	0	0	0	0	0
55	54	1	1	1	0	0	0	0	0
56	55	1	1	1	1	0	0	0	0
57	56	1	1	1	1	0	0	0	1
58	57	1	1	1	0	0	0	1	0
59	58	1	1	0	0	0	1	0	0
60	59	1	0	0	0	1	0	0	0
61	60	0	0	0	0	0	0	0	0
62	61	1	0	0	0	0	0	0	0
63	62	1	1	0	0	0	0	0	0
64	63	1	1	0	0	0	0	0	0
65	64	1	1	1	0	0	0	0	0

66	65	1	1	1	0	0	0	0	0
67	66	1	1	1	1	0	0	0	0
68	67	1	1	1	1	0	0	0	1
69	68	1	1	1	0	0	0	1	0
70	69	1	1	0	0	0	1	0	0
71	70	1	0	0	0	1	0	0	0
72	71	0	0	0	0	0	0	0	0
73	72	1	0	0	0	0	0	0	0
74	73	1	1	0	0	0	0	0	0
75	74	1	1	0	0	0	0	0	0
76	75	1	1	1	0	0	0	0	0
77	76	1	1	1	0	0	0	0	0
78	77	1	1	1	1	0	0	0	0
79	78	1	1	1	1	0	0	0	1
80	79	1	1	1	0	0	0	1	0
81	80	1	1	0	0	0	1	0	0
82	81	1	0	0	0	1	0	0	0
83	82	0	0	0	0	0	0	0	0
84	83	1	0	0	0	0	0	0	0
85	84	1	1	0	0	0	0	0	0
86	85	1	1	0	0	0	0	0	0
87	86	1	1	1	0	0	0	0	0
88	87	1	1	1	0	0	0	0	0
89	88	1	1	1	1	0	0	0	0
90	89	1	1	1	1	0	0	0	1
91	90	1	1	1	0	0	0	1	0
92	91	1	1	0	0	0	1	0	0
93	92	1	0	0	0	1	0	0	0
94	93	0	0	0	0	0	0	0	0
95	94	1	0	0	0	0	0	0	0
96	95	1	1	0	0	0	0	0	0
97	96	1	1	0	0	0	0	0	0
98	97	1	1	1	0	0	0	0	0
99	98	1	1	1	0	0	0	0	0
100	99	1	1	1	1	0	0	0	0
101	100	1	1	1	1	0	0	0	1

FIGURE 4.26 – Les données d'identification reliées aux capteurs et aux actionneurs

Ces données représentent les signaux d'entrée et de sortie liés aux actionneurs et aux capteurs équipant le système.

4.8 Les marquages atteignables par le RdP identifié et son graphe de marquages

Les marquages atteignables par le RdP identifié sont ceux de la figure (4.27) où les marquages indiqués en rouge sont des marquages interdits.

M[p1,p2,p3,p4,p5,p6,p7,p8]		
M0 = [0,0,0,0,4,0,0,0]	M1 = [1,0,0,0,3,0,0,0]	M2 = [0,0,0,0,3,0,0,1]
M3 = [2,0,0,0,2,0,0,0]	M4 = [1,0,0,0,2,0,0,1]	M5 = [0,1,0,0,3,0,0,0]
M6 = [3,0,0,0,1,0,0,0]	M7 = [0,0,0,0,2,0,0,2]	M8 = [2,0,0,0,1,0,0,1]
M9 = [1,1,0,0,2,0,0,0]	M10 = [0,0,0,0,3,0,1,0]	M11 = [4,0,0,0,0,0,0,0]
M12 = [1,0,0,0,1,0,0,2]	M13 = [0,1,0,0,2,0,0,1]	M14 = [3,0,0,0,0,0,0,1]
M15 = [2,1,0,0,1,0,0,0]	M16 = [1,0,0,0,2,0,1,0]	M17 = [0,0,1,0,3,0,0,0]
M18 = [0,0,0,0,1,0,0,3]	M19 = [2,0,0,0,0,0,0,2]	M20 = [1,1,0,0,1,0,0,1]
M21 = [0,0,0,0,2,0,1,1]	M22 = [0,2,0,0,2,0,0,0]	M23 = [3,1,0,0,0,0,0,0]
M24 = [2,0,0,0,1,0,1,0]	M25 = [1,0,1,0,2,0,0,0]	M26 = [0,0,0,0,3,1,0,0]
M27 = [1,0,0,0,0,0,0,3]	M28 = [0,1,0,0,1,0,0,2]	M29 = [2,1,0,0,0,0,0,1]
M30 = [1,0,0,0,1,0,1,1]	M31 = [1,2,0,0,1,0,0,0]	M32 = [0,1,0,0,2,0,1,0]
M33 = [0,0,1,0,2,0,0,1]	M34 = [3,0,0,0,0,0,1,0]	M35 = [2,0,1,0,1,0,0,0]
M36 = [1,0,0,0,2,1,0,0]	M37 = [0,0,0,1,3,0,0,0]	M38 = [0,0,0,0,0,0,0,4]
M39 = [1,1,0,0,0,0,0,2]	M40 = [0,0,0,0,1,0,1,2]	M41 = [0,2,0,0,1,0,0,1]
M42 = [2,0,0,0,0,0,1,1]	M43 = [2,2,0,0,0,0,0,0]	M44 = [1,1,0,0,1,0,1,0]
M45 = [1,0,1,0,1,0,0,1]	M46 = [0,0,0,0,2,0,2,0]	M47 = [0,1,1,0,2,0,0,0]
M48 = [0,0,0,0,2,1,0,1]	M49 = [3,0,1,0,0,0,0,0]	M50 = [2,0,0,0,1,1,0,0]
M51 = [1,0,0,1,2,0,0,0]	M52 = [0,1,0,0,0,0,0,3]	M53 = [1,0,0,0,0,0,1,2]
M54 = [1,2,0,0,0,0,0,1]	M55 = [0,1,0,0,1,0,1,1]	M56 = [0,0,1,0,1,0,0,2]
M57 = [0,3,0,0,1,0,0,0]	M58 = [2,1,0,0,0,0,1,0]	M59 = [2,0,1,0,0,0,0,1]
M60 = [1,0,0,0,1,0,2,0]	M61 = [1,1,1,0,1,0,0,0]	M62 = [1,0,0,0,1,1,0,1]
M63 = [0,0,1,0,2,0,1,0]	M64 = [0,1,0,0,2,1,0,0]	M65 = [0,0,0,1,2,0,0,1]
M66 = [3,0,0,0,0,1,0,0]	M67 = [2,0,0,1,1,0,0,0]	M68 = [0,0,0,0,0,0,1,3]
M69 = [0,2,0,0,0,0,0,2]	M70 = [1,1,0,0,0,0,1,1]	M71 = [1,0,1,0,0,0,0,2]
M72 = [1,3,0,0,0,0,0,0]	M73 = [0,0,0,0,1,0,2,1]	M74 = [0,2,0,0,1,0,1,0]
M75 = [0,1,1,0,1,0,0,1]	M76 = [0,0,0,0,1,1,0,2]	M77 = [2,0,0,0,0,0,2,0]
M78 = [2,1,1,0,0,0,0,0]	M79 = [2,0,0,0,0,1,0,1]	M80 = [1,0,1,0,1,0,1,0]

4.8. Les marquages atteignables par le RdP identifié et son graphe de marquages

M81 = [1,1,0,0,1,1,0,0]	M82 = [1,0,0,1,1,0,0,1]	M83 = [0,0,0,0,2,1,1,0]
M84 = [0,0,2,0,2,0,0,0]	M85 = [0,1,0,1,2,0,0,0]	M86 = [3,0,0,1,0,0,0,0]
M87 = [0,1,0,0,0,0,1,2]	M88 = [0,0,1,0,0,0,0,3]	M89 = [0,3,0,0,0,0,0,1]
M90 = [1,0,0,0,0,0,2,1]	M91 = [1,2,0,0,0,0,1,0]	M92 = [1,1,1,0,0,0,0,1]
M93 = [1,0,0,0,0,1,0,2]	M94 = [0,1,0,0,1,0,2,0]	M95 = [0,0,1,0,1,0,1,1]
M96 = [0,2,1,0,1,0,0,0]	M97 = [0,1,0,0,1,1,0,1]	M98 = [0,0,0,1,1,0,0,2]
M99 = [2,0,1,0,0,0,1,0]	M100 = [2,1,0,0,0,1,0,0]	M101 = [2,0,0,1,0,0,0,1]
M102 = [1,0,0,0,1,1,1,0]	M103 = [1,0,2,0,1,0,0,0]	M104 = [1,1,0,1,1,0,0,0]
M105 = [0,0,1,0,2,1,0,0]	M106 = [0,0,0,1,2,0,1,0]	M107 = [0,0,0,0,0,0,2,2]
M108 = [0,2,0,0,0,0,1,1]	M109 = [0,1,1,0,0,0,0,2]	M110 = [0,0,0,0,0,1,0,3]
M111 = [0,4,0,0,0,0,0,0]	M112 = [1,1,0,0,0,0,2,0]	M113 = [1,0,1,0,0,0,1,1]
M114 = [1,2,1,0,0,0,0,0]	M115 = [1,1,0,0,0,1,0,1]	M116 = [1,0,0,1,0,0,0,2]
M117 = [0,0,0,0,1,0,3,0]	M118 = [0,1,1,0,1,0,1,0]	M119 = [0,0,0,0,1,1,1,1]
M120 = [0,0,2,0,1,0,0,1]	M121 = [0,2,0,0,1,1,0,0]	M122 = [0,1,0,1,1,0,0,1]
M123 = [2,0,0,0,0,1,1,0]	M124 = [2,0,2,0,0,0,0,0]	M125 = [2,1,0,1,0,0,0,0]
M126 = [1,0,1,0,1,1,0,0]	M127 = [1,0,0,1,1,0,1,0]	M128 = [0,0,0,0,2,2,0,0]
M129 = [0,0,1,1,2,0,0,0]	M130 = [0,1,0,0,0,0,2,1]	M131 = [0,0,1,0,0,0,1,2]
M132 = [0,3,0,0,0,0,1,0]	M133 = [0,2,1,0,0,0,0,1]	M134 = [0,1,0,0,0,1,0,2]
M135 = [0,0,0,1,0,0,0,3]	M136 = [1,0,0,0,0,0,3,0]	M137 = [1,1,1,0,0,0,1,0]
M138 = [1,0,0,0,0,1,1,1]	M139 = [1,0,2,0,0,0,0,1]	M140 = [1,2,0,0,0,1,0,0]
M141 = [1,1,0,1,0,0,0,1]	M142 = [0,0,1,0,1,0,2,0]	M143 = [0,1,0,0,1,1,1,0]
M144 = [0,1,2,0,1,0,0,0]	M145 = [0,0,1,0,1,1,0,1]	M146 = [0,0,0,1,1,0,1,1]
M147 = [0,2,0,1,1,0,0,0]	M148 = [2,0,1,0,0,1,0,0]	M149 = [2,0,0,1,0,0,1,0]
M150 = [1,0,0,0,1,2,0,0]	M151 = [1,0,1,1,1,0,0,0]	M152 = [0,0,0,1,2,1,0,0]
M153 = [0,0,0,0,0,0,3,1]	M154 = [0,2,0,0,0,0,2,0]	M155 = [0,1,1,0,0,0,1,1]
M156 = [0,0,0,0,0,1,1,2]	M157 = [0,0,2,0,0,0,0,2]	M158 = [0,3,1,0,0,0,0,0]
M159 = [0,2,0,0,0,1,0,1]	M160 = [0,1,0,1,0,0,0,2]	M161 = [1,0,1,0,0,0,2,0]
M162 = [1,1,0,0,0,1,1,0]	M163 = [1,1,2,0,0,0,0,0]	M164 = [1,0,1,0,0,1,0,1]
M165 = [1,0,0,1,0,0,1,1]	M166 = [1,2,0,1,0,0,0,0]	M167 = [0,0,0,0,1,1,2,0]
M168 = [0,0,2,0,1,0,1,0]	M169 = [0,1,1,0,1,1,0,0]	M170 = [0,1,0,1,1,0,1,0]
M171 = [0,0,0,0,1,2,0,1]	M172 = [0,0,1,1,1,0,0,1]	M173 = [2,0,0,0,0,2,0,0]
M174 = [2,0,1,1,0,0,0,0]	M175 = [1,0,0,1,1,1,0,0]	M176 = [0,0,0,2,2,0,0,0]
M177 = [0,1,0,0,0,0,3,0]	M178 = [0,0,1,0,0,0,2,1]	M179 = [0,2,1,0,0,0,1,0]
M180 = [0,1,0,0,0,1,1,1]	M181 = [0,1,2,0,0,0,0,1]	M182 = [0,0,1,0,0,1,0,2]
M183 = [0,0,0,1,0,0,1,2]	M184 = [0,3,0,0,0,1,0,0]	M185 = [0,2,0,1,0,0,0,1]
M186 = [1,0,0,0,0,1,2,0]	M187 = [1,0,2,0,0,0,1,0]	M188 = [1,1,1,0,0,1,0,0]

M189 = [1,1,0,1,0,0,1,0]	M190 = [1,0,0,0,0,2,0,1]	M191 = [1,0,1,1,0,0,0,1]
M192 = [0,0,1,0,1,1,1,0]	M193 = [0,0,0,1,1,0,2,0]	M194 = [0,0,3,0,1,0,0,0]
M195 = [0,1,0,0,1,2,0,0]	M196 = [0,1,1,1,1,0,0,0]	M197 = [0,0,0,1,1,1,0,1]
M198 = [2,0,0,1,0,1,0,0]	M199 = [1,0,0,2,1,0,0,0]	M200 = [0,0,0,0,0,0,4,0]
M201 = [0,1,1,0,0,0,2,0]	M202 = [0,0,0,0,0,1,2,1]	M203 = [0,0,2,0,0,0,1,1]
M204 = [0,2,0,0,0,1,1,0]	M205 = [0,2,2,0,0,0,0,0]	M206 = [0,1,1,0,0,1,0,1]
M207 = [0,1,0,1,0,0,1,1]	M208 = [0,0,0,0,0,2,0,2]	M209 = [0,0,1,1,0,0,0,2]
M210 = [0,3,0,1,0,0,0,0]	M211 = [1,0,1,0,0,1,1,0]	M212 = [1,0,0,1,0,0,2,0]
M213 = [1,0,3,0,0,0,0,0]	M214 = [1,1,0,0,0,2,0,0]	M215 = [1,1,1,1,0,0,0,0]
M216 = [1,0,0,1,0,1,0,1]	M217 = [0,0,0,0,1,2,1,0]	M218 = [0,0,2,0,1,1,0,0]
M219 = [0,0,1,1,1,0,1,0]	M220 = [0,1,0,1,1,1,0,0]	M221 = [0,0,0,2,1,0,0,1]
M222 = [2,0,0,2,0,0,0,0]	M223 = [0,0,1,0,0,0,3,0]	M224 = [0,1,0,0,0,1,2,0]
M225 = [0,1,2,0,0,0,1,0]	M226 = [0,0,1,0,0,1,1,1]	M227 = [0,0,0,1,0,0,2,1]
M228 = [0,0,3,0,0,0,0,1]	M229 = [0,2,1,0,0,1,0,0]	M230 = [0,2,0,1,0,0,1,0]
M231 = [0,1,0,0,0,2,0,1]	M232 = [0,1,1,1,0,0,0,1]	M233 = [0,0,0,1,0,1,0,2]
M234 = [1,0,0,0,0,2,1,0]	M235 = [1,0,2,0,0,1,0,0]	M236 = [1,0,1,1,0,0,1,0]
M237 = [1,1,0,1,0,1,0,0]	M238 = [1,0,0,2,0,0,0,1]	M239 = [0,0,1,0,1,2,0,0]
M240 = [0,0,0,1,1,1,1,0]	M241 = [0,0,2,1,1,0,0,0]	M242 = [0,1,0,2,1,0,0,0]
M243 = [0,0,0,0,0,1,3,0]	M244 = [0,0,2,0,0,0,2,0]	M245 = [0,1,1,0,0,1,1,0]
M246 = [0,1,0,1,0,0,2,0]	M247 = [0,1,3,0,0,0,0,0]	M248 = [0,0,0,0,0,2,1,1]
M249 = [0,0,2,0,0,1,0,1]	M250 = [0,0,1,1,0,0,1,1]	M251 = [0,2,0,0,0,2,0,0]
M252 = [0,2,1,1,0,0,0,0]	M253 = [0,1,0,1,0,1,0,1]	M254 = [0,0,0,2,0,0,0,2]
M255 = [1,0,1,0,0,2,0,0]	M256 = [1,0,0,1,0,1,1,0]	M257 = [1,0,2,1,0,0,0,0]
M258 = [1,1,0,2,0,0,0,0]	M259 = [0,0,0,0,1,3,0,0]	M260 = [0,0,1,1,1,1,0,0]
M261 = [0,0,0,2,1,0,1,0]	M262 = [0,0,1,0,0,1,2,0]	M263 = [0,0,0,1,0,0,3,0]
M264 = [0,0,3,0,0,0,1,0]	M265 = [0,1,0,0,0,2,1,0]	M266 = [0,1,2,0,0,1,0,0]
M267 = [0,1,1,1,0,0,1,0]	M268 = [0,0,1,0,0,2,0,1]	M269 = [0,0,0,1,0,1,1,1]
M270 = [0,0,2,1,0,0,0,1]	M271 = [0,2,0,1,0,1,0,0]	M272 = [0,1,0,2,0,0,0,1]
M273 = [1,0,0,0,0,3,0,0]	M274 = [1,0,1,1,0,1,0,0]	M275 = [1,0,0,2,0,0,1,0]
M276 = [0,0,0,1,1,2,0,0]	M277 = [0,0,1,2,1,0,0,0]	M278 = [0,0,0,0,0,2,2,0]
M279 = [0,0,2,0,0,1,1,0]	M280 = [0,0,1,1,0,0,2,0]	M281 = [0,0,4,0,0,0,0,0]
M282 = [0,1,1,0,0,2,0,0]	M283 = [0,1,0,1,0,1,1,0]	M284 = [0,1,2,1,0,0,0,0]
M285 = [0,0,0,0,0,3,0,1]	M286 = [0,0,1,1,0,1,0,1]	M287 = [0,0,0,2,0,0,1,1]
M288 = [0,2,0,2,0,0,0,0]	M289 = [1,0,0,1,0,2,0,0]	M290 = [1,0,1,2,0,0,0,0]
M291 = [0,0,0,2,1,1,0,0]	M292 = [0,0,1,0,0,2,1,0]	M293 = [0,0,0,1,0,1,2,0]
M294 = [0,0,3,0,0,1,0,0]	M295 = [0,0,2,1,0,0,1,0]	M296 = [0,1,0,0,0,3,0,0]

4.8. Les marquages atteignables par le RdP identifié et son graphe de marquages

M297 = [0,1,1,1,0,1,0,0]	M298 = [0,1,0,2,0,0,1,0]	M299 = [0,0,0,1,0,2,0,1]
M300 = [0,0,1,2,0,0,0,1]	M301 = [1,0,0,2,0,1,0,0]	M302 = [0,0,0,3,1,0,0,0]
M303 = [0,0,0,0,0,3,1,0]	M304 = [0,0,2,0,0,2,0,0]	M305 = [0,0,1,1,0,1,1,0]
M306 = [0,0,0,2,0,0,2,0]	M307 = [0,0,3,1,0,0,0,0]	M308 = [0,1,0,1,0,2,0,0]
M309 = [0,1,1,2,0,0,0,0]	M310 = [0,0,0,2,0,1,0,1]	M311 = [1,0,0,3,0,0,0,0]
M312 = [0,0,1,0,0,3,0,0]	M313 = [0,0,0,1,0,2,1,0]	M314 = [0,0,2,1,0,1,0,0]
M315 = [0,0,1,2,0,0,1,0]	M316 = [0,1,0,2,0,1,0,0]	M317 = [0,0,0,3,0,0,0,1]
M318 = [0,0,0,0,0,4,0,0]	M319 = [0,0,1,1,0,2,0,0]	M320 = [0,0,0,2,0,1,1,0]
M321 = [0,0,2,2,0,0,0,0]	M322 = [0,1,0,3,0,0,0,0]	M323 = [0,0,0,1,0,3,0,0]
M324 = [0,0,1,2,0,1,0,0]	M325 = [0,0,0,3,0,0,1,0]	M326 = [0,0,0,2,0,2,0,0]
M327 = [0,0,1,3,0,0,0,0]	M328 = [0,0,0,3,0,1,0,0]	M329 = [0,0,0,4,0,0,0,0]

FIGURE 4.27 – Les marquages atteignables par le RdP identifié

Ainsi, le graphe de marquages du modèle RdP identifié en format texte est représenté par le tableau de la figure (4.28).

From	Fired	To	From	Fired	To	From	Fired	To
M0	t5	M1	M1	t1	M2	M1	t5	M3
M3	t5	M6	M4	t1	M7	M4	t5	M8
M6	t1	M8	M6	t5	M11	M7	t5	M12
M8	t6	M15	M9	t1	M13	M9	t2	M16
M11	t1	M14	M12	t1	M18	M12	t5	M19
M13	t6	M22	M14	t1	M19	M14	t6	M23
M16	t1	M21	M16	t5	M24	M16	t7	M25
M18	t6	M28	M19	t1	M27	M19	t6	M29
M20	t6	M31	M21	t5	M30	M21	t6	M32
M23	t1	M29	M23	t2	M34	M24	t1	M30
M25	t3	M36	M25	t5	M35	M26	t5	M36
M28	t2	M40	M28	t5	M39	M28	t6	M41
M30	t1	M40	M30	t5	M42	M30	t6	M44
M31	t5	M43	M32	t2	M46	M32	t5	M44
M33	t6	M47	M34	t1	M42	M34	t7	M49
M36	t1	M48	M36	t5	M50	M36	t8	M51

M39	t1	M52	M39	t2	M53	M39	t6	M54
M41	t2	M55	M41	t5	M54	M41	t6	M57
M43	t1	M54	M43	t2	M58	M44	t1	M55
M45	t1	M56	M45	t3	M62	M45	t5	M59
M47	t2	M63	M47	t3	M64	M47	t5	M61
M49	t1	M59	M49	t3	M66	M50	t1	M62
M51	t4	M1	M51	t5	M67	M52	t2	M68
M53	t7	M71	M54	t1	M69	M54	t2	M70
M55	t6	M74	M55	t7	M75	M56	t3	M76
M57	t5	M72	M58	t1	M70	M58	t2	M77
M59	t6	M78	M60	t1	M73	M60	t5	M77
M61	t3	M81	M61	t5	M78	M62	t1	M76
M63	t3	M83	M63	t5	M80	M63	t7	M84
M65	t4	M2	M65	t5	M82	M65	t6	M85
M67	t4	M3	M67	t5	M86	M68	t6	M87
M70	t1	M87	M70	t2	M90	M70	t6	M91
M71	t6	M92	M72	t1	M89	M72	t2	M91
M74	t2	M94	M74	t5	M91	M74	t7	M96
M75	t6	M96	M76	t5	M93	M76	t6	M97
M78	t1	M92	M78	t2	M99	M78	t3	M100
M80	t1	M95	M80	t3	M102	M80	t5	M99
M81	t5	M100	M81	t8	M104	M82	t1	M98
M83	t5	M102	M83	t7	M105	M83	t8	M106
M85	t4	M5	M85	t5	M104	M86	t1	M101
M87	t7	M109	M88	t3	M110	M88	t6	M109
M90	t6	M112	M90	t7	M113	M91	t1	M108
M92	t2	M113	M92	t3	M115	M92	t6	M114
M94	t2	M117	M94	t5	M112	M94	t7	M118
M95	t7	M120	M96	t2	M118	M96	t3	M121
M97	t6	M121	M97	t8	M122	M98	t4	M7
M99	t3	M123	M99	t7	M124	M100	t1	M115
M101	t4	M8	M101	t6	M125	M102	t1	M119
M103	t1	M120	M103	t3	M126	M103	t5	M124
M104	t5	M125	M105	t3	M128	M105	t5	M126
M106	t7	M129	M107	t6	M130	M107	t7	M131
M109	t2	M131	M109	t3	M134	M109	t6	M133

4.8. Les marquages atteignables par le RdP identifié et son graphe de marquages

M112	t1	M130	M112	t2	M136	M112	t7	M137
M113	t7	M139	M114	t1	M133	M114	t2	M137
M115	t6	M140	M115	t8	M141	M116	t1	M135
M117	t7	M142	M118	t2	M142	M118	t3	M143
M119	t6	M143	M119	t7	M145	M119	t8	M146
M121	t2	M143	M121	t5	M140	M121	t8	M147
M122	t6	M147	M123	t1	M138	M123	t7	M148
M125	t1	M141	M125	t2	M149	M125	t4	M15
M126	t8	M151	M127	t1	M146	M127	t4	M16
M128	t8	M152	M129	t3	M152	M129	t4	M17
M130	t7	M155	M131	t3	M156	M131	t6	M155
M133	t2	M155	M133	t3	M159	M133	t6	M158
M135	t4	M18	M135	t6	M160	M136	t1	M153
M137	t3	M162	M137	t7	M163	M138	t1	M156
M139	t1	M157	M139	t3	M164	M139	t6	M163
M141	t1	M160	M141	t2	M165	M141	t4	M20
M142	t7	M168	M143	t2	M167	M143	t5	M162
M144	t3	M169	M144	t5	M163	M145	t3	M171
M146	t4	M21	M146	t5	M165	M146	t6	M170
M147	t5	M166	M148	t1	M164	M148	t3	M173
M149	t7	M174	M150	t1	M171	M150	t5	M173
M151	t4	M25	M151	t5	M174	M152	t4	M26
M153	t7	M178	M154	t2	M177	M154	t7	M179
M155	t7	M181	M156	t6	M180	M156	t7	M182
M158	t2	M179	M158	t3	M184	M159	t2	M180
M160	t4	M28	M160	t6	M185	M161	t1	M178
M162	t2	M186	M162	t7	M188	M162	t8	M189
M164	t1	M182	M164	t3	M190	M164	t6	M188
M165	t6	M189	M165	t7	M191	M166	t1	M185
M167	t7	M192	M167	t8	M193	M168	t3	M192
M169	t3	M195	M169	t5	M188	M169	t8	M196
M170	t7	M196	M171	t5	M190	M171	t6	M195
M172	t5	M191	M172	t6	M196	M173	t1	M190
M174	t4	M35	M175	t1	M197	M175	t4	M36
M176	t5	M199	M177	t2	M200	M177	t7	M201
M179	t2	M201	M179	t3	M204	M179	t7	M205

M180	t8	M207	M181	t2	M203	M181	t3	M206
M182	t8	M209	M183	t4	M40	M183	t6	M207
M185	t2	M207	M185	t4	M41	M185	t6	M210
M187	t1	M203	M187	t3	M211	M187	t7	M213
M188	t8	M215	M189	t1	M207	M189	t2	M212
M190	t6	M214	M190	t8	M216	M191	t1	M209
M192	t3	M217	M192	t5	M211	M192	t7	M218
M193	t7	M219	M194	t3	M218	M194	t5	M213
M196	t2	M219	M196	t3	M220	M196	t4	M47
M197	t6	M220	M197	t8	M221	M198	t1	M216
M199	t4	M51	M199	t5	M222	M200	t7	M223
M202	t6	M224	M202	t7	M226	M202	t8	M227
M204	t2	M224	M204	t7	M229	M204	t8	M230
M206	t3	M231	M206	t6	M229	M206	t8	M232
M207	t7	M232	M208	t6	M231	M208	t8	M233
M210	t2	M230	M210	t4	M57	M211	t1	M226
M212	t1	M227	M212	t4	M60	M212	t7	M236
M214	t2	M234	M214	t8	M237	M215	t1	M232
M216	t1	M233	M216	t4	M62	M216	t6	M237
M217	t8	M240	M218	t3	M239	M218	t5	M235
M219	t5	M236	M219	t7	M241	M220	t2	M240
M221	t4	M65	M221	t5	M238	M221	t6	M242
M223	t7	M244	M224	t2	M243	M224	t7	M245
M225	t7	M247	M226	t3	M248	M226	t6	M245
M227	t6	M246	M227	t7	M250	M228	t3	M249
M229	t8	M252	M230	t2	M246	M230	t4	M74
M231	t8	M253	M232	t2	M250	M232	t3	M253
M233	t6	M253	M233	t8	M254	M234	t1	M248
M235	t3	M255	M235	t8	M257	M236	t1	M250
M237	t1	M253	M237	t2	M256	M237	t4	M81
M238	t6	M258	M239	t3	M259	M239	t5	M255
M240	t7	M260	M240	t8	M261	M241	t3	M260
M242	t4	M85	M242	t5	M258	M243	t7	M262
M245	t2	M262	M245	t3	M265	M245	t7	M266
M246	t7	M267	M247	t2	M264	M247	t3	M266
M249	t3	M268	M249	t6	M266	M249	t8	M270

4.8. Les marquages atteignables par le RdP identifié et son graphe de marquages

M250	t7	M270	M251	t2	M265	M251	t8	M271
M253	t2	M269	M253	t4	M97	M253	t6	M271
M255	t1	M268	M255	t3	M273	M255	t8	M274
M256	t8	M275	M257	t1	M270	M257	t3	M274
M258	t4	M104	M259	t5	M273	M259	t8	M276
M260	t8	M277	M261	t4	M106	M261	t5	M275
M262	t8	M280	M263	t4	M117	M263	t7	M280
M265	t7	M282	M265	t8	M283	M266	t2	M279
M267	t3	M283	M267	t4	M118	M267	t7	M284
M269	t4	M119	M269	t6	M283	M269	t7	M286
M270	t6	M284	M271	t2	M283	M271	t4	M121
M272	t6	M288	M273	t1	M285	M273	t8	M289
M274	t8	M290	M275	t1	M287	M275	t4	M127
M276	t8	M291	M277	t3	M291	M277	t4	M129
M279	t3	M292	M279	t7	M294	M279	t8	M295
M281	t3	M294	M282	t2	M292	M282	t3	M296
M283	t7	M297	M283	t8	M298	M284	t2	M295
M285	t8	M299	M286	t3	M299	M286	t4	M145
M287	t6	M298	M287	t7	M300	M288	t2	M298
M289	t8	M301	M290	t1	M300	M290	t3	M301
M291	t8	M302	M292	t3	M303	M292	t7	M304
M293	t8	M306	M294	t3	M304	M294	t8	M307
M296	t2	M303	M296	t8	M308	M297	t2	M305
M298	t2	M306	M298	t4	M170	M298	t7	M309
M300	t3	M310	M300	t4	M172	M300	t6	M309
M302	t4	M176	M302	t5	M311	M303	t7	M312
M305	t3	M313	M305	t4	M192	M305	t7	M314
M307	t3	M314	M307	t4	M194	M308	t2	M313
M309	t3	M316	M309	t4	M196	M310	t4	M197
M311	t4	M199	M312	t3	M318	M312	t8	M319
M314	t3	M319	M314	t4	M218	M314	t8	M321
M316	t2	M320	M316	t4	M220	M316	t8	M322
M319	t3	M323	M319	t4	M239	M319	t8	M324
M321	t3	M324	M321	t4	M241	M322	t2	M325
M324	t3	M326	M324	t4	M260	M324	t8	M327
M326	t8	M328	M327	t3	M328	M327	t4	M277

M2	t5	M4	M2	t6	M5	M3	t1	M4
M4	t6	M9	M5	t2	M10	M5	t5	M9
M7	t6	M13	M8	t1	M12	M8	t5	M14
M9	t5	M15	M10	t5	M16	M10	t7	M17
M12	t6	M20	M13	t2	M21	M13	t5	M20
M15	t1	M20	M15	t2	M24	M15	t5	M23
M17	t3	M26	M17	t5	M25	M18	t5	M27
M20	t1	M28	M20	t2	M30	M20	t5	M29
M21	t7	M33	M22	t2	M32	M22	t5	M31
M24	t5	M34	M24	t7	M35	M25	t1	M33
M26	t8	M37	M27	t1	M38	M27	t6	M39
M29	t1	M39	M29	t2	M42	M29	t6	M43
M30	t7	M45	M31	t1	M41	M31	t2	M44
M32	t7	M47	M33	t3	M48	M33	t5	M45
M35	t1	M45	M35	t3	M50	M35	t5	M49
M37	t4	M0	M37	t5	M51	M38	t6	M52
M40	t5	M53	M40	t6	M55	M40	t7	M56
M42	t1	M53	M42	t6	M58	M42	t7	M59
M44	t2	M60	M44	t5	M58	M44	t7	M61
M45	t6	M61	M46	t5	M60	M46	t7	M63
M48	t5	M62	M48	t6	M64	M48	t8	M65
M50	t5	M66	M50	t8	M67	M51	t1	M65
M52	t6	M69	M53	t1	M68	M53	t6	M70
M54	t6	M72	M55	t2	M73	M55	t5	M70
M56	t5	M71	M56	t6	M75	M57	t2	M74
M58	t7	M78	M59	t1	M71	M59	t3	M79
M60	t7	M80	M61	t1	M75	M61	t2	M80
M62	t5	M79	M62	t6	M81	M62	t8	M82
M64	t2	M83	M64	t5	M81	M64	t8	M85
M66	t1	M79	M66	t8	M86	M67	t1	M82
M68	t7	M88	M69	t2	M87	M69	t6	M89
M70	t7	M92	M71	t1	M88	M71	t3	M93
M73	t5	M90	M73	t6	M94	M73	t7	M95
M75	t2	M95	M75	t3	M97	M75	t5	M92
M76	t8	M98	M77	t1	M90	M77	t7	M99
M79	t1	M93	M79	t6	M100	M79	t8	M101

4.8. Les marquages atteignables par le RdP identifié et son graphe de marquages

M80	t7	M103	M81	t1	M97	M81	t2	M102
M82	t4	M4	M82	t5	M101	M82	t6	M104
M84	t3	M105	M84	t5	M103	M85	t2	M106
M86	t4	M6	M87	t2	M107	M87	t6	M108
M89	t2	M108	M89	t6	M111	M90	t1	M107
M91	t2	M112	M91	t7	M114	M92	t1	M109
M93	t1	M110	M93	t6	M115	M93	t8	M116
M95	t3	M119	M95	t5	M113	M95	t6	M118
M96	t5	M114	M97	t2	M119	M97	t5	M115
M98	t5	M116	M98	t6	M122	M99	t1	M113
M100	t2	M123	M100	t8	M125	M101	t1	M116
M102	t5	M123	M102	t7	M126	M102	t8	M127
M104	t1	M122	M104	t2	M127	M104	t4	M9
M105	t8	M129	M106	t4	M10	M106	t5	M127
M108	t2	M130	M108	t6	M132	M108	t7	M133
M110	t6	M134	M110	t8	M135	M111	t2	M132
M113	t1	M131	M113	t3	M138	M113	t6	M137
M114	t3	M140	M115	t1	M134	M115	t2	M138
M116	t4	M12	M116	t6	M141	M117	t5	M136
M118	t5	M137	M118	t7	M144	M119	t5	M138
M120	t3	M145	M120	t5	M139	M120	t6	M144
M122	t2	M146	M122	t4	M13	M122	t5	M141
M123	t8	M149	M124	t1	M139	M124	t3	M148
M126	t1	M145	M126	t3	M150	M126	t5	M148
M127	t5	M149	M127	t7	M151	M128	t5	M150
M129	t5	M151	M130	t2	M153	M130	t6	M154
M131	t7	M157	M132	t2	M154	M132	t7	M158
M134	t2	M156	M134	t6	M159	M134	t8	M160
M136	t7	M161	M137	t1	M155	M137	t2	M161
M138	t6	M162	M138	t7	M164	M138	t8	M165
M140	t1	M159	M140	t2	M162	M140	t8	M166
M141	t6	M166	M142	t3	M167	M142	t5	M161
M143	t7	M169	M143	t8	M170	M144	t2	M168
M145	t5	M164	M145	t6	M169	M145	t8	M172
M146	t7	M172	M147	t2	M170	M147	t4	M22
M148	t8	M174	M149	t1	M165	M149	t4	M24

M150	t8	M175	M151	t1	M172	M151	t3	M175
M152	t5	M175	M152	t8	M176	M153	t6	M177
M155	t2	M178	M155	t3	M180	M155	t6	M179
M156	t8	M183	M157	t3	M182	M157	t6	M181
M159	t6	M184	M159	t8	M185	M160	t2	M183
M161	t3	M186	M161	t7	M187	M162	t1	M180
M163	t1	M181	M163	t2	M187	M163	t3	M188
M164	t8	M191	M165	t1	M183	M165	t4	M30
M166	t2	M189	M166	t4	M31	M167	t5	M186
M168	t5	M187	M168	t7	M194	M169	t2	M192
M170	t2	M193	M170	t4	M32	M170	t5	M189
M171	t8	M197	M172	t3	M197	M172	t4	M33
M173	t8	M198	M174	t1	M191	M174	t3	M198
M175	t5	M198	M175	t8	M199	M176	t4	M37
M178	t3	M202	M178	t6	M201	M178	t7	M203
M180	t2	M202	M180	t6	M204	M180	t7	M206
M181	t6	M205	M182	t3	M208	M182	t6	M206
M183	t7	M209	M184	t2	M204	M184	t8	M210
M186	t1	M202	M186	t7	M211	M186	t8	M212
M188	t1	M206	M188	t2	M211	M188	t3	M214
M189	t4	M44	M189	t7	M215	M190	t1	M208
M191	t3	M216	M191	t4	M45	M191	t6	M215
M192	t8	M219	M193	t4	M46	M193	t5	M212
M195	t2	M217	M195	t5	M214	M195	t8	M220
M196	t5	M215	M197	t4	M48	M197	t5	M216
M198	t4	M50	M198	t8	M222	M199	t1	M221
M201	t2	M223	M201	t3	M224	M201	t7	M225
M203	t3	M226	M203	t6	M225	M203	t7	M228
M205	t2	M225	M205	t3	M229	M206	t2	M226
M207	t2	M227	M207	t4	M55	M207	t6	M230
M209	t3	M233	M209	t4	M56	M209	t6	M232
M211	t3	M234	M211	t7	M235	M211	t8	M236
M213	t1	M228	M213	t3	M235	M214	t1	M231
M215	t2	M236	M215	t3	M237	M215	t4	M61
M216	t8	M238	M217	t5	M234	M217	t7	M239
M218	t8	M241	M219	t3	M240	M219	t4	M63

4.8. Les marquages atteignables par le RdP identifié et son graphe de marquages

M220	t4	M64	M220	t5	M237	M220	t8	M242
M222	t1	M238	M222	t4	M67	M223	t3	M243
M224	t8	M246	M225	t2	M244	M225	t3	M245
M226	t7	M249	M226	t8	M250	M227	t4	M73
M228	t6	M247	M229	t2	M245	M229	t3	M251
M230	t7	M252	M231	t2	M248	M231	t6	M251
M232	t4	M75	M232	t6	M252	M233	t4	M76
M234	t7	M255	M234	t8	M256	M235	t1	M249
M236	t3	M256	M236	t4	M80	M236	t7	M257
M237	t8	M258	M238	t1	M254	M238	t4	M82
M239	t8	M260	M240	t4	M83	M240	t5	M256
M241	t4	M84	M241	t5	M257	M242	t2	M261
M243	t8	M263	M244	t3	M262	M244	t7	M264
M245	t8	M267	M246	t2	M263	M246	t4	M94
M248	t6	M265	M248	t7	M268	M248	t8	M269
M250	t3	M269	M250	t4	M95	M250	t6	M267
M252	t2	M267	M252	t3	M271	M252	t4	M96
M253	t8	M272	M254	t4	M98	M254	t6	M272
M256	t1	M269	M256	t4	M102	M256	t7	M274
M257	t4	M103	M258	t1	M272	M258	t2	M275
M260	t3	M276	M260	t4	M105	M260	t5	M274
M261	t7	M277	M262	t3	M278	M262	t7	M279
M264	t3	M279	M264	t7	M281	M265	t2	M278
M266	t3	M282	M266	t8	M284	M267	t2	M280
M268	t3	M285	M268	t6	M282	M268	t8	M286
M269	t8	M287	M270	t3	M286	M270	t4	M120
M271	t8	M288	M272	t2	M287	M272	t4	M122
M274	t1	M286	M274	t3	M289	M274	t4	M126
M275	t7	M290	M276	t4	M128	M276	t5	M289
M277	t5	M290	M278	t7	M292	M278	t8	M293
M280	t3	M293	M280	t4	M142	M280	t7	M295
M282	t8	M297	M283	t2	M293	M283	t4	M143
M284	t3	M297	M284	t4	M144	M285	t6	M296
M286	t6	M297	M286	t8	M300	M287	t4	M146
M288	t4	M147	M289	t1	M299	M289	t4	M150
M290	t4	M151	M291	t4	M152	M291	t5	M301

M292	t8	M305	M293	t4	M167	M293	t7	M305
M295	t3	M305	M295	t4	M168	M295	t7	M307
M297	t3	M308	M297	t4	M169	M297	t8	M309
M299	t4	M171	M299	t6	M308	M299	t8	M310
M301	t1	M310	M301	t4	M175	M301	t8	M311
M303	t8	M313	M304	t3	M312	M304	t8	M314
M305	t8	M315	M306	t4	M193	M306	t7	M315
M308	t4	M195	M308	t8	M316	M309	t2	M315
M310	t6	M316	M310	t8	M317	M311	t1	M317
M313	t4	M217	M313	t7	M319	M313	t8	M320
M315	t3	M320	M315	t4	M219	M315	t7	M321
M317	t4	M221	M317	t6	M322	M318	t8	M323
M320	t4	M240	M320	t7	M324	M320	t8	M325
M322	t4	M242	M323	t4	M259	M323	t8	M326
M325	t4	M261	M325	t7	M327	M326	t4	M276
M328	t4	M291	M328	t8	M329	M329	t4	M302

FIGURE 4.28 – Le graphe de marquages du modèle RdP identifié en format texte

Table des figures

1.1	Un exemple d'un SED	12
1.2	Automate à états finis	12
1.3	Automate autonome non déterministe avec fonction de sortie	13
1.4	Un RdP marqué modilisant le système de l'exemple 1	15
1.5	Franchissement d'une transition	16
1.6	Classification des méthodes d'identification d'un SED	22
1.7	Automate initial	27
1.8	L'AANDS final	27
1.9	L'arbre de préfixes final de toutes les séquences d'événements	29
1.10	L'automate initial qui génère l'ensemble des séquences d'événements Q	30
1.11	L'automate final A_f qui génère l'ensemble des séquences d'événements Q	30
1.12	Le RdP marqué qui génère les séquences d'événements de l'ensemble Q	31
1.13	Les réseaux de Petri synthétisés	33
1.14	Le réseau de Petri affiné	34
1.15	Le système à identifier et sa spécification	36
1.16	Le modèle identifié	37
1.17	Le RdP obtenu	39
1.18	Le modèle RdP identifié unique	41
1.19	Le modèle RdP identifié 1	42
1.20	Le modèle RdP identifié 2	42
2.1	Schéma de séparation et évaluation du système de l'exemple 2	51
2.2	L'architecture du système à identifier	52
2.3	Système à deux réservoirs	53
2.4	Le modèle RdP identifié	61
2.5	Le modèle identifié 1 avec une place non mesurable	65
2.6	Le modèle identifié 2 avec une place non mesurable	65
2.7	Un modèle RdP identifié avec deux places non mesurables	66
2.8	La solution optimale	66
2.9	Solution réalisable pour $\bar{n} = 4$ et une place non mesurable	69

2.10	Solution réalisable pour $\bar{n} = 4$ et deux places non mesurables	70
2.11	Le modèle identifié optimal	79
3.1	Problème de transitions d'état interdites	85
3.2	Problème d'états interdits	86
3.3	Le modèle RdP à contrôler avec une <i>CGEM</i>	89
3.4	Le modèle RdP contrôlé correspondant au modèle de la figure 3.3	90
3.5	Exemple d'un RdP à contrôler	92
3.6	Le modèle RdP contrôlé correspondant au modèle de la figure 3.5	93
3.7	Un exemple de problèmes d'états interdits	95
3.8	Une solution du problème de blocage avec une place de contrôle	96
3.9	Un chemin d'influence π_{α_i} d'une transition t_{α_i}	100
3.10	Le modèle RdP à contrôler	105
3.11	Le graphe de marquages du modèle RdP à contrôler	105
3.12	Le modèle RdP contrôlé en appliquant l'algorithme 3	107
3.13	marquages légaux et marquages interdits	108
3.14	Le modèle RdP contrôlé en appliquant l'algorithme 4	113
4.1	La machine PRODUCTIS	116
4.2	Les postes de la machine	117
4.3	Poste 1 : distribution de comprimés blancs	118
4.4	Poste 2 : fermeture des gros flacons par bouchonnage	119
4.5	Poste 3 : distribution de comprimés vert	119
4.6	Poste 4 : fermeture des petits flacons	120
4.7	Poste 4 : chargement des palettes	121
4.8	Les automates reliés à la machine PRODUCTIS	121
4.9	Le modèle RdP du système obtenu après une modélisation boîte blanche	123
4.10	Les données d'identification liées aux capteurs	124
4.11	Les données finales d'identification liées aux capteurs	125
4.12	Une solution optimale avec une place non mesurable	127
4.13	Une solution optimale avec deux places non mesurables	128
4.14	Une solution optimale avec trois places non mesurables	128
4.15	Une solution optimale avec quatre places non mesurables	129
4.16	Une solution optimale avec huit places non mesurables	129
4.17	Le temps de calcul nécessaire pour avoir une solution optimale en appliquant la première méthode d'identification	130
4.18	Les données d'identification finales liées aux capteurs et aux actionneurs	132
4.19	Une solution optimale avec une place non mesurable	134
4.20	Une solution optimale avec deux places non mesurables	135
4.21	Une solution optimale avec trois places non mesurables	135
4.22	Une solution optimale avec quatre places non mesurables	136

4.23	Le temps de calcul nécessaire pour avoir une solution optimale en appliquant la deuxième méthode	137
4.24	Graphe de marquages partiel du modèle RdP identifié	139
4.25	Le RdP identifié final	141
4.26	Les données d'identification reliées aux capteurs et aux actionneurs .	151
4.27	Les marquages atteignables par le RdP identifié	155
4.28	Le graphe de marquages du modèle RdP identifié en format texte . . .	164

Bibliographie

- [ABT02] L. Aguirre, O. Begovich, et A. R. Treviño. Observability in interpreted Petri nets using sequence invariants. 4 :3602–3607, 2002.
- [AIE03] R. T. Antonio, R. R. Israel, et L. M. Ernesto. Observability of discrete event systems modeled by interpreted Petri nets. *IEEE transactions on robotics and automation*, 19 :557–565, 2003.
- [All05] G. Allaire. *Analyse numérique et optimisation : une introduction à la modélisation mathématique et à la simulation numérique*. ellipses, 2005.
- [AR05] Z. Achour et N. Rezg. Time floating general mutual exclusion constraints in partial observable discrete event systems. In *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration*, pages 75–79, 2005.
- [AR07] Z. Achour et N. Rezg. Time floating general mutual exclusion constraints (TFGMEC). *Studies in Informatics and Control*, 16 :57–66, 2007.
- [ARX04] Z. Achour, N. Rezg, et X. Xie. Supervisory control of partially observable marked graphs. *IEEE Transactions on Automatic control*, 49 :2007–2011, 2004.
- [ARX05] Z. Achour, N. Rezg, et X. Xie. On the existence of Petri net controller for discrete event systems under partial observation. In *Proceedings of 16th IFAC World Congress on Automatic Control*, pages 385–390, 2005.
- [BBB95] R. K. Boel, L. Ben, et V. V. Breusegem. On forbidden state problem for class of controlled Petri nets. *IEEE Transactions on Automatic Control*, 40 :1717–1731, 1995.
- [BCG06] F. Basile, P. Chiacchio, et A. Giua. Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places. 42 :995–1004, 2006.

- [BF72] A. W. Biermann et J. A. Feldman. On the synthesis of finite-state machines from samples of their behavior. *IEEE Transactions on Computers*, 21 :592–597, 1972.
- [BME⁺06a] R. Bekrar, N. Messai, N. Essounbouli, A. Hamzaoui, et B. Riera. Identification of discrete event systems using ordinary Petri nets. In *Proceedings of the Workshop on Advanced Control and Diagnosis (IAR-ACD'06)*, CDROM papier N 20, 2006.
- [BME⁺06b] R. Bekrar, N. Messai, N. Essounbouli, A. Hamzaoui, et B. Riera. Off-line identification for class of discrete event systems using safe Petri nets. In *Proceedings of the 3rd IFAC workshop on Discrete Event System Design (DESDes'06)*, pages 221–226, 2006.
- [BME⁺07] R. Bekrar, N. Messai, N. Essounbouli, A. Hamzaoui, et B. Riera. Identification des systèmes à événements discrets par réseaux de Petri. In *Proceedings of JD-JN-MACS 2007*, CDROM papier N 60, 2007.
- [BME⁺08a] R. Bekrar, N. Messai, N. Essounbouli, et A. Hamzaoui. Synthèse d'un contrôleur RdP pour le problème d'états interdits. In *Proceedings de la Conférence Internationale Francophone d'Automatique*, CDROM Papier N 226, 2008.
- [BME⁺08b] R. Bekrar, N. Messai, N. Essounbouli, A. Hamzaoui, et B. Riera. Control synthesis approach for des modelled by Petri nets. In *Proceedings of 17th IFAC World Congress*, pages 3580–3585, 2008.
- [BME⁺08c] R. Bekrar, N. Messai, N. Essounbouli, A. Hamzaoui, et B. Riera. Synthesis method of a PN controller using forbidden transitions sequences. In *Proceedings of International Conference on Informatics in Control, Automation and Robotics*, pages 149–154, 2008.
- [Boo67] T. L. Booth. *Sequential Machines and Automata Theory*. John Wiley and Sons, Inc. New York, London, Sidney, 1967.
- [BPG07] F. Basile, P. Chiacchio, et A. Giua. An optimisation approach to Petri net monitor design. *IEEE Transaction on Automation and Control*, 52 :306–311, 2007.
- [BY04] T. Bourdeaud'huy et P. Yim. Synthèse de réseaux de Petri à partir d'exigences. In *MOSIM'04, Conférence Francophone de Modélisation et Simulation*, page 413–420, 2004.

- [CGS06a] M. P. Cabasino, A. Giua, et C. Seatzu. Computational complexity analysis of a Petri net identification procedure. In *Proceeding of 2006 International Symposium on Nonlinear Theory and its Applications*, 2006.
- [CGS06b] M. P. Cabasino, A. Giua, et C. Seatzu. Identification of deterministic Petri nets. In *Proceeding of the 8th International Workshop on Discrete Event Systems*, pages 325–331, 2006.
- [CGS07] M. P. Cabasino, A. Giua, et C. Seatzu. Identification of Petri nets from knowledge of their language. *Discrete Event Dynamic Systems*, 17 :447–474, 2007.
- [CL99] C. G. Cassandras et S. Lafortune. *Introduction to discrete event systems*. kluwer academic publishers, 1999. ISBN 0-7923-8609-4.
- [DA99] R. David et H. Alla. *Du Grafset aux réseaux de Petri*. Hermes, 1999. ISBN 0-7923-8609-4.
- [DFM06] M. Dotoli, M. P. Fanti, et A. M. Mangini. An optimisation approach for identification of Petri nets. In *Proceeding of 8th International Workshop on Discrete Event systems*, pages 332–337, 2006.
- [DFM07] M. Dotoli, M. P. Fanti, et A. M. Mangini. Real time identification of discrete event systems by Petri nets. In *Proceeding of the 1st IFAC Workshop on dependable control of discrete systems (DCDS'07)*, 2007.
- [DFM08a] M. Dotoli, M. P. Fanti, et A. M. Mangini. Real time identification of discrete event systems using Petri nets. *Automatica*, 44 :1209–1219, 2008.
- [DFM⁺08b] M. Dotoli, M. P. Fanti, A. M. Mangini, et W. Ukovich. On-line identification of Petri nets with unobservable transitions. In *Proceedings of the 9th International Workshop on Discrete Event Systems*, pages 449–454, 2008.
- [ELL06] S. O. Elmahdi, E. Leclercq, et D. Lefebvre. Synthèse de modèles par réseaux de Petri pour le diagnostic des systèmes à événements discrets. In *Proceeding de 15^{me} congrès de Maîtrise des Risques et de Sûreté de Fonctionnement*, 2006.
- [GDS92] A. Giua, F. Dicesare, et M. Silva. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 934–939, 1992.

- [GDS93] A. Giua, F. DiCesare, et M. Silva. Petri net supervisors for generalised mutual exclusion constraints. In *Proceeding of the 12th IFAC world Congress*, pages 267–270, 1993.
- [Gha02] A. Ghaffari. *Les réseaux de Petri pour la synthèse de contrôle des systèmes à événements discrets*. PhD thesis, Université de Metz, 2002.
- [Giu96] A. Giua. Petri net techniques for supervisory control of discrete event systems. In *Proceeding of the First International Workshop on Manufacturing and Petri Nets*, pages 1–30, 1996.
- [Gou02] G. Gousset. Identification de systèmes logiques séquentiels par observation et analyse énumérative. Master’s thesis, Université de Nancy 1 et ENS de cachan, 2002.
- [GRX01a] A. Ghaffari, N. Rezg, et X. L. Xie. Conception du superviseur optimal vivant à l’aide de la théorie des régions. In *Proceedings du colloque francophone sur la modélisation des systèmes réactifs*, 2001.
- [GRX01b] A. Ghaffari, N. Rezg, et X. L. Xie. Live and maximally permissive controller synthesis using the theory of regions. In *Proceedings of the symposium on the supervisory control of discrete event system*, 2001.
- [GRX01c] A. Ghaffari, N. Rezg, et X. L. Xie. State feedback control of discrete event system using marked graphs. In *Proceedings of 40 IEEE Conference on decision and control*, pages 4998–5003, 2001.
- [GRX02a] A. Ghaffari, N. Rezg, et X. Xie. Algebraic and geometric characterization of Petri net controllers using the theory of regions. In *Proceedings of 6th Workop on Discrete Event Systems*, pages 219– 224, 2002.
- [GRX02b] A. Ghaffari, N. Rezg, et X. Xie. Net transformation and theory of regions for optimal control of Petri nets. In *Proceedings of 15th IFAC World Congress on Automatic Control*, pages 443–448, 2002.
- [GRX02c] A. Ghaffari, N. Rezg, et X. Xie. Theory of regions based synthesis of the maximally permissive and non blocking supervisory control. *Journal européen des systèmes automatisés*, 36 :919–930, 2002.
- [GRX03a] A. Ghaffari, N. Rezg, et X. Xie. Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE transactions on robotics and automation*, 19 :137–142, 2003.

- [GRX03b] A. Ghaffari, N. Rezg, et X. Xie. Feedback control logic for forbidden-state problems of marked graphs : Application to a real manufacturing system. *IEEE Transactions on Automatic Control*, 48 :18–29, 2003.
- [GS01] A. Giua et C. Seatzu. Supervisory control of railway networks with Petri nets. In *Proceeding of the 40th IEEE International Conference on Decision and Control*, pages 5004–5009, 2001.
- [GS05] A. Giua et C. Seatzu. Identification of free-labeled Petri nets via integer programming. In *Proceeding of the 44th IEEE conference on Decision and Control, and the european Control Conference*, pages 7639–7644, 2005.
- [GS07] A. Giua et C. Seatzu. A system theory view of Petri nets. *Advances in Control Theory and Applications Series : Lecture Notes in Control and Information Sciences*, 353, 2007.
- [HG06] M. Gilson et T. Bastogne H. Garnier. Identification de modèles paramétriques à temps continu. méthodes, outil logiciel et avantages. In *Proceeding de JIME'2006 Journées Identification et Modélisation Expérimentale*, 2006.
- [HGZ96] L. E. Holloway, X. Guan, et L. Zhang. A generalization of state avoidance policies for controlled Petri nets. *IEEE Transactions on Automatic Control*, 41 :804–816, 1996.
- [Hir92] K. Hiraishi. Construction of a class of safe Petri nets by presenting firing sequences. *Lecture Notes in Computer Science*, 616 :244–262, 1992. ISBN :3-540-55676-1.
- [HK90] L. E. Holloway et B. H. Krogh. synthesis of feedback control logic for a class of controlled Petri nets. *IEEE Transactions on Automatic Control*, 35 :514–523, 1990.
- [HMU01] J. E. Hopcroft, R. Motwani, et J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley, second edition, 2001.
- [IA05] M. Iordache et P. Antsaklis. A survey on the supervision of Petri nets. In *Proceeding of DES Workshop PN 2005*, 2005.
- [Ior03] M. V. Iordache. *Methods for the supervisory control of concurrent systems based on Petri Net abstractions*. PhD thesis, Graduate School of the University of Notre Dame(Indiana), 2003.

- [Ke171] J. Kella. Sequential machine identification. *IEEE Transactions on Computers*, 20 :332–338, 1971.
- [KG95] R. Kumar et V. K. Garg. Optimal supervisory control of discrete event dynamical systems. *SIAM Journal of Control and Optimization*, 33 :419–439, 1995.
- [Kle05] S. Klein. *Identification of Discrete Event Systems for Fault Detection Purposes*. PhD thesis, Ecole Normale Supérieure (ENS) de Cachan (France) et Université de Kaiserslautern (Allemagne), 2005.
- [KLL05] S. Klein, L. Litz, et J. J. Lesage. Fault detection of discrete event systems using an identification approach. In *Proceeding of 16th IFAC World Congress, CDROM paper N 02643*, 2005.
- [KR03] M. Kaloper et P. Rudnicki. Minimization of finite state machines. *Journal of formalized mathematics*, 6, 2003.
- [Lee06] E. J. Lee. *Reconfiguration dynamique de la commande d'un système manufacturier : approche par la synthèse de la commande*. PhD thesis, École Centrale de Lille, 2006.
- [Lef05] D. Lefebvre. Sensor selection for immediate diagnosis of faulty behaviours prediction. In *Proceedings of the Workshop on Advanced Control and Diagnosis (IAR-ACD'05)*, pages 127–132, 2005.
- [Lju87] L. Ljung. *System identification : Theory for the user*. Prentice-Hall, Englewood Cliffs, N. J, 1987. ISBN 0-13-881640.
- [LPS03] P. Lacomme, C. Prins, et M. Sevaux. *Algorithmes de graphes*, pages 21–42. Edition Eyrolles, 2003. ISBN 2-212-11385-4.
- [LTD06] E. J. Lee, A. Toguyeni, et N. Dangoumauv. A Petri net based approach for the synthesis of parts' controllers for reconfigurable manufacturing systems. In *Proceeding of SICE-ICASE International Joint Conference*, pages 5567–5572, 2006.
- [MM01] M. E. Meda et E. L. Mellado. A passive method for on-line identification of discrete event systems. In *Proceedings of the 40th IEEE Conference on Decision and Control(CDC'01)*, volume 5, pages 4990–4995, 2001.
- [MM02] M. E. Meda et E. L. Mellado. Incremental synthesis of Petri net models for identification of discrete event systems. In *Proceedings of the 41th IEEE Conference on Decision and Control(CDC'02)*, volume 1, pages 805–810, 2002.

- [MM03] M. E. Meda et E. L. Mellado. Required event sequences for identification of discrete event systems. In *Proceedings of the 42th IEEE Conference on Decision and Control(CDC'03)*, volume 4, pages 3778–3783, 2003.
- [MM05] M. E. Meda et E. L. Mellado. Identification of concurrent discrete event systems using Petri nets. In *Proceeding of the 17th IMACS World Congress*, page 11Ú15, 2005.
- [MRM98] M. E. Meda, A. Ramirez, et A. Malo. Identification in discrete event systems. In *Proceedings of the IEEE International Conference Systems, Man and Cybernetics*, pages 740–745, 1998.
- [MTM00] M. E. Meda, A. R. Trevino, et E. L. Mellado. Asymptotic identification of discrete event systems. In *Proceedings of the 39th IEEE Conference on Decision and Control(CDC'00)*, volume 3, pages 2266–2271, 2000.
- [Mur89] T. Murata. Petri nets : properties, analysis and applications. *Proceedings of the IEEE*, 4 :541–580, 1989.
- [Pin06] J. E. Pin. *Automates finis*. Encyclopédie de l'informatique et des systèmes d'information, 2006.
- [PX95] J. M. Proth et X. Xie. *Les réseaux de Petri pour la conception et la gestion des systèmes de production*. Manuels Informatiques Masson, 1995.
- [RW87a] P. J. Ramadge et W. M. Wonham. Modular feedback logic for discrete event systems. *SIAM Journal of Control and Optimization*, 25 :1202–1218, 1987.
- [RW87b] P. J. Ramadge et W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25 :206–230, 1987.
- [RW89] P. J. Ramadge et W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE ; Special issue on Dynamics of Discrete Event Systems*, 77 :81–98, 1989.
- [Sre06] R. S. Sreenivas. On minimal representations of Petri net languages. *IEEE Transactions on Automatic Control*, 51 :799– 804, 2006.
- [Sud96] T. A. Sudkamp. *Languages and Machines : An Introduction to the Theory of Computer Science*. Addison Wesley, second edition, 1996.
- [Vee81] L. P. J. Veelenturf. An automata theoretical approach to developing learning neural networks. *Cybernetics and Systems*, 12 :179–202, 1981.

- [YML⁺96] K. Yamalidou, J. Moody, M. Lemmon, et P. Antsaklis. Feedback control of Petri nets based on place invariants. *Automatica*, 32 :15–28, 1996.
- [Zay01] J. Zaytoon. *Système dynamique hybrides*. Hermes Science Publications, 2001. ISBN 2-7462-0247-6.

