

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8Mai 1945 – Guelma
Faculté des Sciences et de la Technologie
Département de Génie Electrotechnique et Automatique

682



**Mémoire de fin d'étude
pour l'obtention du diplôme de Master Académique**

Domaine : Sciences et Techniques
Filière : Automatique et Informatique Industrielle
Spécialité : Commande et Diagnostic des Systèmes Industriels



**Modélisation et Simulation des Systèmes Dynamiques
Hybrides par Réseaux de Petri Hybrides**

Présenté par :

Baali Abderrahmen

Sous la direction de :

Madame : *Boubidi*

JUIN 2011

11/2688



A tous ceux qui aiment le savoir et n'ont
cessé de l'avoir.

A mes très chers parents

A mon frère et à mes deux sœurs

A toute ma famille.

Remerciements

Je remercie tout d'abord Allah le tout puissant de m'avoir aidé à achever ce travail présenté dans ce mémoire.

L'achèvement de ce document n'a été possible sans l'aide et le soutien de plusieurs personnes qui ont contribué de près ou de loin à mon mémoire.

Je tiens tout d'abord à remercier mon encadreur Madame Boubidi, pour la confiance qu'elle m'a accordé, pour son esprit scientifique, pour le temps qu'elle m'a consacré.

Mes remerciements profonds à tous mes amis et collègues, en particulier ceux et celles qui m'ont apporté un soutien moral et une amitié inoubliable et précieuse.

En fin, je remercie l'université 08 mai 1945 de Guelma et tous ses enseignants pour les cinq années d'accueil pleines de plaisir, bonheur et savoir que j'ai passé de ma vie.

SOMMAIRE

<i>Introduction Générale.....</i>	<i>1</i>
<i>Chapitre 1 : Les systèmes dynamiques hybrides.....</i>	<i>4</i>
1.1. <i>Introduction.....</i>	<i>5</i>
1.2. <i>Classification des systèmes dynamiques.....</i>	<i>5</i>
1.2.1. <i>Les systèmes dynamique continus.....</i>	<i>5</i>
1.2.2. <i>Les systèmes dynamiques à événement discrets.....</i>	<i>6</i>
1.2.3. <i>Les systèmes dynamiques hybrides.....</i>	<i>7</i>
1.2.4. <i>Les systèmes dynamiques positifs.....</i>	<i>7</i>
1.3. <i>Les systèmes dynamique hybrides (SDH).....</i>	<i>8</i>
1.3.1. <i>Structure d'un SDH.....</i>	<i>8</i>
1.3.2. <i>Exemples illustratifs.....</i>	<i>9</i>
1.3.3. <i>Les difficultés liées aux SDH.....</i>	<i>10</i>
1.4. <i>Un modèle, pour quoi faire ?.....</i>	<i>10</i>
1.5. <i>Modèles hybrides.....</i>	<i>11</i>
1.5.1. <i>Introduction.....</i>	<i>11</i>
1.5.2. <i>Automates hybrides.....</i>	<i>12</i>
1.5.3. <i>Réseaux de Petri hybrides.....</i>	<i>12</i>
1.5.4. <i>Réseaux de Petri mixtes.....</i>	<i>12</i>
1.6. <i>Pour quoi simuler ?.....</i>	<i>13</i>
1.7. <i>Simulation des SDH.....</i>	<i>13</i>
1.8. <i>Conclusion.....</i>	<i>14</i>
<i>Chapitre 2 : Les Réseaux de Petri.....</i>	<i>15</i>
2.1. <i>Introduction.....</i>	<i>16</i>
2.2. <i>Outil de modélisation.....</i>	<i>16</i>
2.3. <i>Réseaux de Petri discrets.....</i>	<i>17</i>
2.3.1. <i>Notions de base.....</i>	<i>17</i>

2.3.1.1. Définition.....	17
2.3.1.2. Places, transitions et arcs.....	17
2.3.1.3. Marquage.....	17
2.3.1.4. Franchissement d'une transition.....	18
2.3.1.5. Séquence de franchissement.....	19
2.3.1.6. Marquages accessible.....	19
2.3.1.7. Graphe de marquages.....	20
2.3.2. RdP Particuliers.....	20
2.3.2.1. Graphe d'état.....	20
2.3.2.2. Graphe d'événement.....	21
2.3.2.3. RdP sans conflit.....	21
2.3.2.4. RdP généralisés.....	22
2.3.2.5. RdP à capacités.....	22
2.3.2.6. RdP à priorités.....	23
2.3.3. Algèbre Linéaire.....	23
2.3.3.1. Notation et définitions.....	23
2.3.3.2. Equation fondamentale ou équation d'état.....	24
2.3.4. RdP temporisés.....	25
2.3.4.1. Réseaux de Petri P-temporisés.....	25
2.3.4.2. Réseaux de Petri T-temporisés.....	26
2.3.4.3. RdP T-temporels.....	27
2.4. Modélisation par RdP.....	27
2.5. Les RdP du discrets aux continus.....	29
2.6. Réseaux de Petri continues.....	30
2.6.1. Introduction.....	30
2.6.2. Les Réseaux de Petri Continu à vitesses Constantes.....	30
2.6.2.1. Définition et notations.....	30
2.6.2.2. Principe de fonctionnement.....	31
2.6.3. Les autres modèles continus.....	32
2.6.3.1. Le RdP continu à vitesses variables.....	32
2.6.3.2. Le RdP continu asymptotique.....	32
2.7. Réseaux de Petri hybrides (RdPH).....	32
2.7.1. Introduction.....	32
2.7.2. L'influence de la partie discrète sur la partie continue.....	33

2.7.3. L'influence de la partie continue sur la partie discrète.....	34
2.7.4. Partage de ressource.....	34
2.7.5. Transformation de marques.....	34
2.7.6. Test à zéro et un arc de poids O^+	36
2.8. Conclusion.....	37

Chapitre 3 : Modélisation et simulation des SDH par RdPH.....38

3.1. Introduction.....	39
3.2. Présentation de l'application SIRPHYCO.....	39
3.3. Architecture de l'application SIRPHYCO.....	39
3.4. Fonctionnement de l'application SIRPHYCO.....	40
3.5. Exemples de simulation.....	40
3.5.1. Exemple 1 : Système de recyclage d'eau.....	41
3.5.2. Exemple 2 : Machine produisant des pièces par lots.....	43
3.5.3. Exemple 3 : Circuit électrique (Le hacheur abaisseur).....	45
3.4. Conclusion.....	47

Conclusion générale.....48

Bibliographie.....50

Introduction Générale

Les innovations technologiques ont eu un impact considérable sur l'apparition des processus dynamiques ayant une nature hétérogène mélangeant le continu et le discret. Ces progrès technologiques ont un impact similaire dans le domaine de la recherche scientifique où on constate un intérêt particulier pour l'étude de ces systèmes dits "hybrides". De tels systèmes sont caractérisés par l'interaction des parties continues régies par des équations différentielles et des parties discrètes, décrites par des machines à états finis ou des réseaux de Petri.

En automatique, les systèmes physiques sont souvent représentés par un modèle dynamique continu ou par un modèle à événements discrets. La nature de chaque modèle est définie selon les variables utilisées pour décrire l'état du système et la variable caractérisant le temps. Il est important dans de nombreux cas d'utiliser l'une de ces deux catégories de modèles. Cependant la majorité des systèmes complexes réalistes mélangeant le continu et le discret ne peuvent pas être classés ni dans la catégorie "système continu" ni dans la catégorie "*système discret*". Il est nécessaire alors d'utiliser des modèles hybrides permettant la prise en compte à la fois des variables continues et des variables discrètes ainsi que l'interaction entre elles. Plusieurs modélisations hybrides ont été développées dans la littérature. Elles peuvent être classées en trois catégories principales :

- ❖ Les approches basées sur une extension de techniques de modélisation de systèmes continus comme le formalisme de bond-graph mixte.
- ❖ Les approches basées sur l'extension de modèles de systèmes à événements discrets parmi lesquels nous pouvons citer les réseaux de Petri hybrides.
- ❖ Les approches mixtes, combinant à la fois le continu et le discret dans une structure unique.

A nos jours, il existe dans la littérature principalement deux formalismes permettant la modélisation de l'aspect hybride : les *automates hybrides* ou bien par les *réseaux de Petri hybrides (RdPH)*. D'une manière informelle et générale, un *automate hybride* est l'association d'un automate d'états finis et un ensemble d'équations dynamiques continues pilotées par ce dernier. Les équations modélisant le comportement continu à un instant donné dépendent de l'état de l'automate, ce dernier pouvant évoluer en fonction des valeurs des grandeurs continues. Toutefois, l'utilisation de ce formalisme est limitée par la taille du modèle qu'il engendre (explosion combinatoire du nombre d'états du graphe). D'où le besoin de disposer de mécanismes de structurations plus puissants comme les *réseaux de Petri (RdP)*. Ces derniers sont largement utilisés dans la modélisation des systèmes à événements discrets et dans les études de sûreté de fonctionnement des systèmes dynamiques. Ils se caractérisent par une évolution asynchrone dans laquelle les transitions des composantes parallèles sont franchies les unes après les autres, et par une représentation explicite des synchronisations et des mécanismes d'allocation de ressources. Ces caractéristiques sont très intéressantes pour modéliser les aspects événementiels des systèmes hybrides.

L'objectif de ce mémoire est modélisé et simulé des systèmes dynamiques hybrides. Pour atteindre cet objectif, le mémoire est réparti en trois chapitres comme suit :

Chapitre 1

Dans le premier chapitre, nous avons présenté la notion des systèmes dynamiques hybrides allant de la classification des systèmes dynamiques à la structure de ces derniers. Par la suite, nous avons présenté deux exemples illustratifs. Enfin nous avons terminé par les outils utilisés pour la modélisation des Systèmes Dynamiques Hybrides.

Chapitre 2

Ce chapitre est consacré à la présentation de l'outil de modélisation des Systèmes Dynamiques Hybrides. Il y a de nombreux modèles pouvant prendre en compte la modélisation des Systèmes Dynamiques Hybrides. Les deux principaux sont les RdP hybride et les automates hybrides. Nous avons retenu les RdP hybride à cause de leur puissance de modélisation et de leur capacité à fournir une simulation pilotée par des événements.

Chapitre 3

Dans ce dernier chapitre on a basé sur la modélisation et la simulation des Systèmes Dynamiques Hybrides par réseaux de Petri hybrides et dans cette partie, nous avons choisi trois exemples des SDH pour la simulation. Enfin nous avons terminé par une conclusion générale.

Chapitre 1

Les Systèmes Dynamiques Hybrides

1.1. Introduction

Les systèmes dynamiques hybrides sont des systèmes comportant un système continu interagissant avec et/ou supervisé par un système à événements discrets. Un système se caractérise par la nature de ses variables d'états, qui peuvent être continues ou discrètes. Le premier type est défini sur l'ensemble des réels \mathcal{R} , alors que le second prend ses valeurs dans un ensemble dénombrable (entiers, booléens,...). Pour définir un système hybride, nous allons d'abord introduire les notions de système continu et de système à événement discret.

1.2. Classification des systèmes dynamiques

Suivant la nature des variables d'état et de la variable d'état indépendante qui est le temps, on peut classer les systèmes dynamiques comme suit :

1.2.1. Les systèmes dynamiques continus

Dans cette catégorie la variable temps est toujours présente, elle peut être continue ou discrète. Le comportement de ce type de système est décrit par des variables d'état continues, le temps lui peut être continu, le système est dit à temps continu (*figure. 1.1.a*), ou discret, c'est le cas des systèmes discrétisés, dont l'exemple le plus répandu sont les systèmes échantillonnés (*figure. 1.1.b*). La représentation de l'évolution de ces grandeurs, traduisant le comportement dynamique du procédé, est alors une représentation mathématique à l'aide de modèles continus, tels que les équations différentielles, les équations récurrentes, les équations aux différences, les fonctions de transfert, ou encore les équations d'état.

Un système continu est formellement défini comme suit :

Définition 1.1 (Système Dynamique Continu) : Un système dynamique continu est un triplet (X, X_0, f) tel que :

- ❖ $X \subseteq \mathcal{R}^n$ est l'espace d'état continu ;
- ❖ $X_0 \subseteq X$ est l'espace d'état initial du système ;
- ❖ $f : X \rightarrow \mathcal{R}^n$ est le champ de vecteur continu ;

Le comportement d'un système continu est le plus souvent décrit par une équation différentielle de la forme

$$\dot{x} = f(x)$$

Où $x \in X$ est l'état du système, le comportement d'un système continu est caractérisé par la solution de l'équation différentielle $\dot{x} = f(x)$ à partir de l'état initial $x_0 \in X_0$.

Un système continu est dit linéaire s'il est modélisé par une équation différentielle de la forme

$$\dot{x} = A \cdot x$$

Où $\mathcal{R}^{n \times n}$ est une matrice constante.

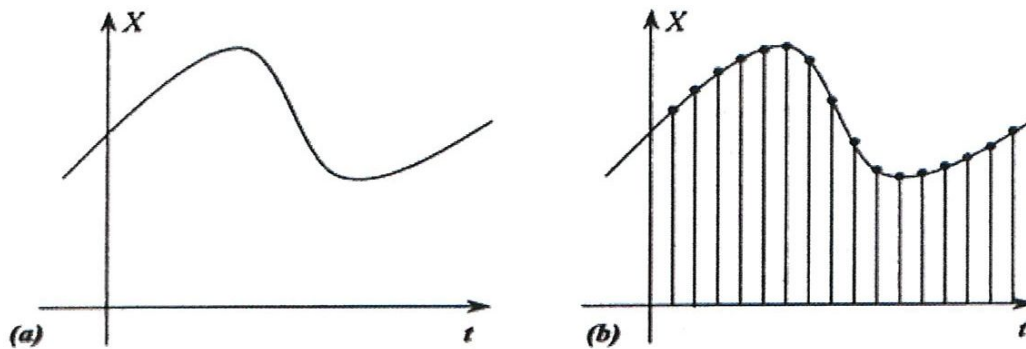


Figure. 1.1. Système dynamique continu, (a) Système à temps continu, (b) Système échantillonné.

1.2.2. Les systèmes dynamiques à événement discrets

Un système à événement discrets (SED) est décrit par des variables d'état discrètes, qui prennent leurs valeurs dans un ensemble dénombrable. Ce type de système peut être non temporisé (autonome) ou temporisé. Dans le cas d'un SED autonome, la variable temps est symbolique, c'est-à-dire qu'elle est utilisée pour définir une chronologie entre les événements. Dans le cas des SED temporisés, le temps peut être continu ou discret (figure. 1.2), à chaque événement est rattaché l'instant de son occurrence qui prend ses valeurs dans \mathcal{R} , cas d'un automate temporisé par exemple. Les SED peuvent être modélisés par les automates à état, les réseaux de Petri, les chaînes de Markov, l'algèbre $(\text{Max}, +)$, ...etc.

Un système continu est formellement défini comme suit :

Définition 1.2 (Système Dynamique à Événements Discrets) : Un système dynamique continu est un triplet (Q, Q_0, δ) tel que :

- ❖ Q est un ensemble fini ou infini d'états ;
- ❖ $Q_0 \subseteq Q$ est l'ensemble des états initiaux ;
- ❖ $\delta : Q \times Q$ est la fonction de transition entre états, elle détermine le comportement du système ;

Notons que le comportement d'un même système physique peut être décrit par des variables d'états continues ou discrètes, suivant le phénomène auxquels on s'intéresse.

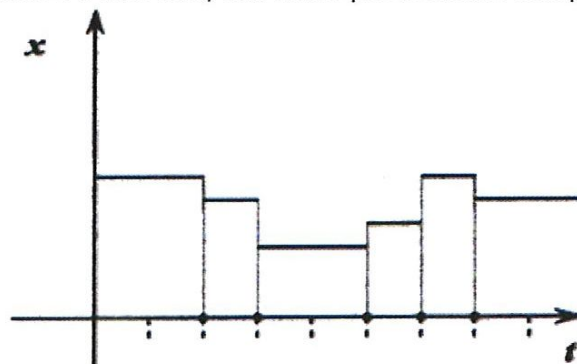


Figure. 1.2. Système à événements discrets.

1.2.3. Les systèmes dynamiques hybrides

Dans certains cas et pour des objectifs spécifiques, il est possible de choisir toutes les variables d'état dans l'une des deux catégories, pour la description d'un système physique donnée, mais ce n'est pas toujours le cas. En effet, la plupart des systèmes physiques ne peuvent pas être classés dans l'une des deux catégories précédentes homogènes des systèmes dynamiques, et les variables d'états auxquelles on s'intéresse contiennent des variables discrètes et d'autres continues. Dans ce cas les systèmes sont dits systèmes dynamiques hybrides, ce sont des systèmes hétérogènes caractérisés par l'interaction d'une dynamique de nature discrète et une autre de nature événementielle. L'essor de ces systèmes est relativement nouveau, il est une combinaison d'un système à événement continu et un système à événements discrets, en insistant sur l'importance des interfaces qui assurent les connections entre les dynamiques continue et discrète, comme le montre la *Figure. 1.3.*

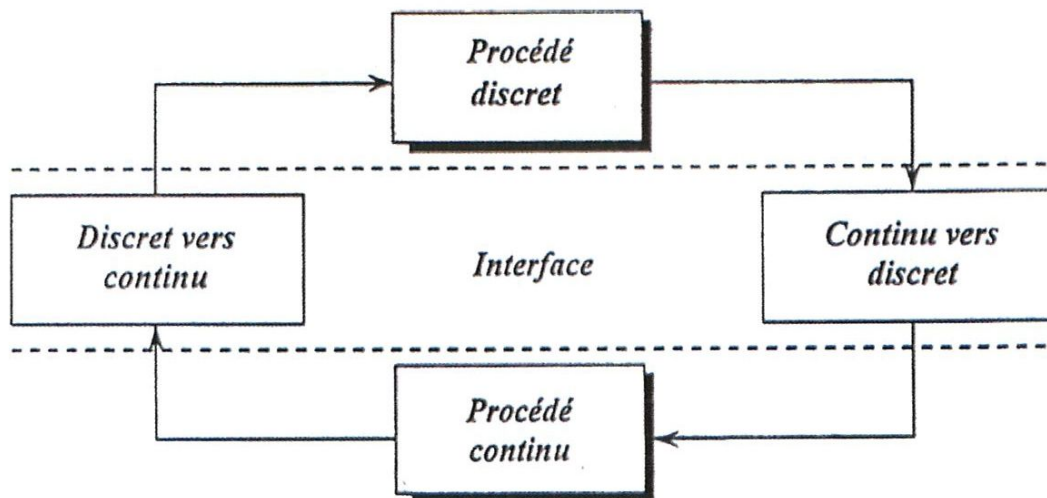


Figure. 1.3. Système dynamique hybride.

La recherche sur les systèmes dynamiques hybrides s'articule autour des trois axes complémentaires suivant :

- ❖ *La modélisation* : Formaliser des modèles précis qui peuvent décrire le comportement riche et complexe des SDH.
- ❖ *L'analyse* : développer des outils de simulation, de validation et de vérification des SDH.
- ❖ *La commande* : effectuer la synthèse d'un contrôleur discret (ou hybride) conformément à certains objectifs de performances et de sûreté de fonctionnement du procédé hybride commandé.

1.2.4. Les systèmes dynamiques positifs

Les systèmes positifs sont une autre classe des systèmes dynamique relative aux variables d'état. Cette terminologie a été présentée pour la première fois par *David Luenberger* dans son livre « *Introduction to dynamic Systems* ». Un système dynamique est dit positif si toutes ses variables d'état sont toujours positives. Cette positivité est souvent

une conséquence de la nature du phénomène. Des exemples de tels phénomènes peuvent être issus de disciplines aussi différentes telles que l'économie (les variables descriptives peuvent être des prix de marchandises, des quantités de stocks), les sciences sociales (nombres d'individus, des taux de satisfaction, de consommation), la physique (problèmes de réservoirs d'eau, systèmes d'irrigation), la chimie (réacteurs chimiques, colonnes à distiller), les télécommunications (quantité d'informations à faire transiter par un réseau).

1.3. Les systèmes dynamiques hybrides (SDH)

1.3.1. Structure d'un SDH

Un SDH est un système dynamique composé de l'interaction d'une partie événementielle et d'une partie continue. Les deux composantes continues et discrètes d'un SDH sont interconnectées avec une loi qui orchestre cette interconnexion (*figure. 1.4*).

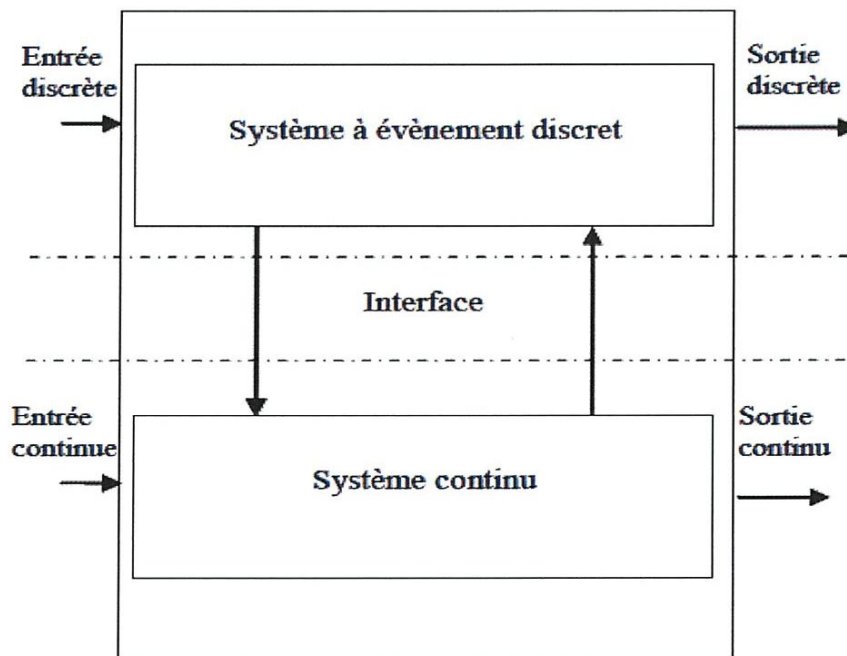


Figure. 1.4. Structure d'un SDH.

- ❖ La partie « Système à Événements Discrets » (SED) représente l'état discret du SDH. Son évolution est gouvernée par l'occurrence d'événements tels que l'opération d'usinage sur une pièce, l'instruction traitée par un processeur, un interrupteur ouvert ou fermé D'habitude, les SED sont classiquement représentés par l'algèbre de Boole combinée à des formalismes états transitions décrits par les modèles usuels tels que les automates à états finis ou bien les réseaux de Petri.
- ❖ La partie continue symbolise l'environnement physique dans lequel évolue le système. Elle est constituée d'une infinité de valeurs décrivant le comportement d'un phénomène physique tel que la température d'une pièce ou d'un objet, la vitesse d'un mobile, le niveau dans un réservoir...etc. L'étude de ces systèmes fait appel à des outils mathématiques capables de représenter la dynamique continue: équations différentielles, inclusions différentielles, méthodes d'état à forme matricielle....etc.

- ❖ L'interface ou bien l'interaction exprime la relation liant la partie discrète à la partie continue et vice versa. Le rôle de cette partie est de fournir des informations à la partie discrète respectivement à la partie continue sur l'évolution continue ou discrète. Sous l'effet d'un évènement interne ou bien d'un évènement externe, le système peut changer de configuration ou de mode.

De ce fait, les interactions entre les deux modèles se font par l'intermédiaire des évènements. Au niveau de la partie discrète, un évènement correspond à un franchissement de transition. Alors qu'au niveau des systèmes continus, il s'agit d'un dépassement de seuil d'une variable continue. Une transition d'un mode vers un autre mode a lieu lorsque certaines conditions logiques sont vérifiées.

1.3.2. Exemples illustratifs

Comme mentionné précédemment, un système est qualifié d'hybride s'il implique des processus continus et des phénomènes événementiels. Par extension, nous pouvons qualifier d'hybrides des systèmes physiques dont certaines grandeurs varient très rapidement par rapport aux autres. Une modélisation hybride pour cette catégorie de systèmes physiques est envisageable et donne souvent de meilleurs résultats par rapport à une modélisation événementielle. Nous allons présenter ici deux exemples de systèmes hybrides, le premier est un système de bacs impliquant un flux de liquide (continu) et le second est un système manufacturier traitant un flot de produits (discret approché par une description continue).

Exemple. 1.1 : Soit le système de bacs schématisé en (figure. 1.5). Ce système comporte deux bacs qui sont vidés en permanence à un débit de 5 et 7 litres/seconde respectivement. Les bacs sont aussi alimentés à tour de rôle, avec une vanne dont le débit est 12 litres/seconde. Cette dernière a deux positions, quand elle est en position A, elle alimente le bac1 et elle alimente le bac 2 si elle est en position B. Pour commuter entre les positions A et B la vanne a besoin de 0.5 seconde. Pendant lesquelles, la vanne se comporte comme si elle est à son ancienne position.

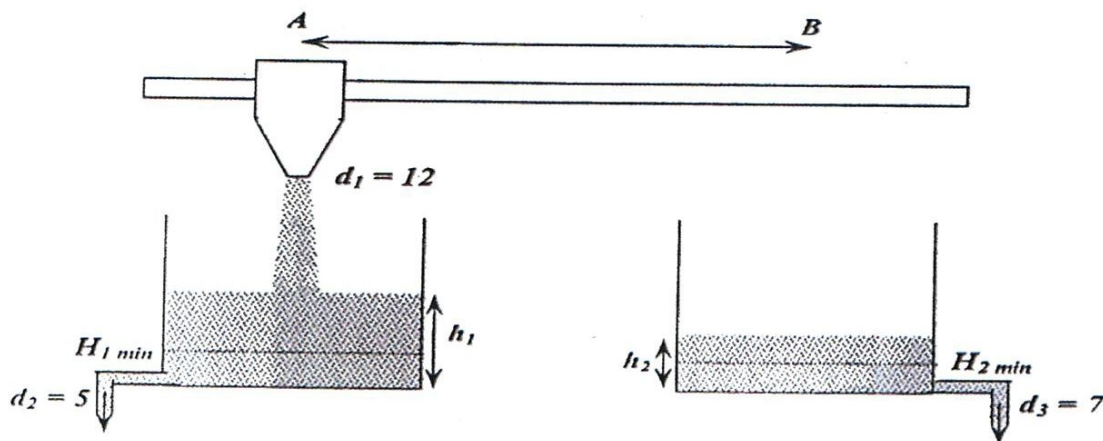


Figure. 1.5. Système de bacs.

Exemple. 1.2 : La figure. 1.6. représente un système manufacturier comportant 3 machines et 2 stocks tampons. Ce système est utilisé pour satisfaire une demande périodique, dont la période est de 20 unités de temps. Les machines 1 et 2 restent opératoires en permanence, tandis que la machine 3 peut être arrêtée pour la régulation du taux de fabrication. Les actions de mise en arrêt et en fonctionnement de la machine 3

prennent 5 u.t. les machines ont des taux de fabrication de 10, 7, et 22 pièces/unité de temps, respectivement. Dans ce système le flux des pièces est supposé être un processus continu, tandis que l'état de la machine 3 ainsi que l'état de la demande sont des variables discrètes.

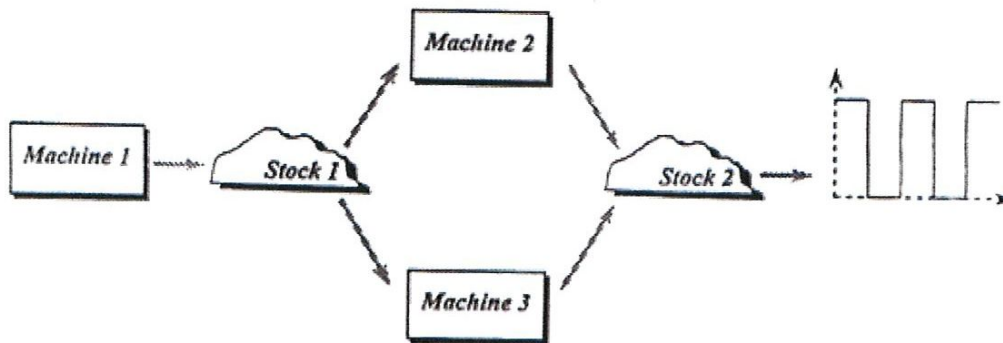


Figure. 1.6. Système manufacturier.

1.3.3. Les difficultés liées aux SDH

Van der Schaft et Schumacher énumèrent dans « *The complementary formalism for hybrid systems* » les principaux comportements types qu'il est possible de rencontrer en étudiant la trajectoire d'un système dynamique hybride :

- ❖ Le système arrive dans un état où il n'y a plus de trajectoires continues définies et où il n'y a pas de transitions vers un autre état. Le système est dit « *bloqué* » ;
- ❖ Les durées des trajectoires continues (i.e., le temps pendant lequel le système évolue entre deux sauts/commutations) deviennent de plus en plus petites. Le système est dit « *Zeno* » ;
- ❖ L'évolution du système rencontre plusieurs événements simultanés. Nous avons plusieurs changements d'état au même instant.
- ❖ Le système commute indéfiniment entre deux états, il est alors appelé « *livelock* » ;
- ❖ L'ensemble des trajectoires est vaste et nous n'avons donc pas une solution unique ;
- ❖ Un état du système commute sur lui-même indéfiniment ;
- ❖ La partie continue tend vers l'infini dans un temps fini.

1.4. Un modèle, pour quoi faire ?

L'utilisation de modèles prend une part de plus en plus importante dans les projets technologiques menés par les ingénieurs, que ce soit pour la définition de systèmes, leur conception, leur réalisation et même leur intégration (mise en œuvre), leur maintenance et leur exploitation.

Par définition, un modèle est une représentation d'un système : il peut ainsi permettre de décrire de façon non équivoque un système technologique. De sa définition à son exploitation, un système technologique va être appréhendé par différentes personnes

(ingénieurs, etc..) sous des points de vue différents (conception, réalisation, maintenance, exploitation, etc.). Le passage d'une personne à l'autre doit alors se faire sans perte d'information, ce que permet l'utilisation d'un modèle. Ce problème est suffisamment central pour qu'un énorme effort soit consacré au niveau de la définition et de la mise en œuvre de normes et de standards industriels. Par exemple, la modélisation du fonctionnement demandé à un automate industriel est possible par l'utilisation du Grafcet, qui est un outil de modélisation dérivant des Réseaux de Petri.

Lors de la conception d'un nouveau système technologique, le cahier des charges exprime le comportement attendu du système. La question fondamentale est de garantir que le système qui a été conçu (sur le papier) remplit bien le cahier des charges (satisfaisant). L'approche traditionnelle de la conception d'un système consiste à le concevoir sur le papier, à le réaliser et à faire des expériences sur le système correspondant à des scénarios types afin de vérifier si son comportement est satisfaisant et s'il est nécessaire de l'améliorer. Cette approche pose plusieurs problèmes. Dans certains cas, il est impossible de réaliser des expériences sur le système (par exemple, systèmes spatiaux). Parfois, cela peut être dangereux si des informations manquent sur son comportement possible.

De façon plus courante, une telle approche est longue et coûteuse, ce que permettent de moins en moins les contraintes économiques. Une alternative est de définir le système technologique à l'aide d'un modèle. En plus de l'avantage vu précédemment, pour un scénario donné, un modèle permet en général de calculer numériquement les valeurs des variables d'état et de sortie, ce qui est plus court et plus économique que de les mesurer au cours d'une expérience. Cela est désigné par le terme de simulation.

La simulation permet dans une certaine mesure de tester si le comportement du système est satisfaisant et de mettre en évidence certains problèmes. Elle ne remplace pas complètement l'expérimentation qu'il est nécessaire d'effectuer si à l'issue de la simulation le comportement du système semble satisfaisant. Le problème de la simulation est que, même si le modèle est bon, on ne peut tester qu'un nombre limité de scénarios. Si le choix des scénarios types n'est pas pertinent par rapport à l'ensemble des scénarios auxquels sera confronté le système durant son existence, il est difficile de prévoir si le comportement du système sera satisfaisant dans tous les cas. Une alternative est de parfois pouvoir garantir qu'à partir d'une propriété du modèle, le modèle étant supposé bon, le comportement du système sera (ou non) satisfaisant pour une famille de scénarios. On parle d'analyse. L'analyse d'un système repose sur l'étude des propriétés mathématiques de son modèle.

1.5. Modèles hybrides

1.5.1. Introduction

C'est une étape nécessaire à toute étude qui ne se réduit pas à l'expérimentation. Sa qualité, mais aussi sa lisibilité et ses possibilités d'utilisation, sont essentielles. Nous présentons dans la suite trois outils utilisés pour la modélisation des SDH grâce, notamment, à leur aspect graphique. Il s'agit des automates hybrides, des réseaux de Petri hybrides et des réseaux de Petri mixtes.

1.5.2. Automates hybrides

Un automate hybride se présente, fondamentalement, comme un automate à état fini avec des équations différentielles associées à ses états discrets. Ainsi, l'état global d'un automate hybride, à un instant donné, est défini par une paire (q, X) , q représentant la situation (état discret) et X la valeur du vecteur d'état (au sens du continu). Cet état global se modifie pour deux raisons :

- ❖ Le franchissement d'une transition discrète, qui change brusquement la situation et souvent alors l'évolution de l'état continu, voire directement la valeur de cet état (saut). Ce franchissement se produit sur occurrence d'un événement approprié et/ou si une condition devient vraie ;
- ❖ L'évolution temporelle qui affecte X suivant l'équation différentielle associée à la situation courante. Cette situation reste inchangée.

L'avantage de cette représentation est sa simplicité. Les évolutions possibles d'un système hybride, elle sera à la base de l'analyse en vue d'établir des propriétés formelles. À chaque instant, un seul état discret est actif, donc il n'y a qu'un seul jeu d'équations (un seul modèle continu). Le caractère hybride se marque par le fait qu'un événement discret peut entraîner le changement d'état, donc la commutation du jeu d'équations, mais l'atteinte d'une valeur seuil sur une variable continue peut aussi entraîner un changement d'état discret.

1.5.3. Réseaux de Petri hybrides

Les réseaux de Petri ont été largement utilisés pour représenter des systèmes à événements discrets. Une des difficultés que soulève l'exploitation des réseaux de Petri est l'augmentation rapide de la complexité du modèle, induite par la possibilité d'avoir un nombre quelconque de jetons dans les places. Cela a conduit à introduire des réseaux de Petri continus (RdPC) où le marquage devient un nombre réel positif. Des RdPC, on est passé aux réseaux de Petri continus temporisés, puis aux réseaux de Petri hybrides (RdPH), aux réseaux de Petri lots et à d'autres extensions.

1.5.4. Réseaux de Petri mixtes

Ce modèle, dont la structure est basée sur les RdP discrets, intègre la dynamique continue de la manière suivante :

- ❖ Une place peut être associée à une équation ou à un jeu d'équations différentielles définissant le comportement continu. Il n'y a alors qu'une seule marque dans cette place, le comportement du modèle devant, dans chaque situation, être défini de manière unique ;
- ❖ Une transition peut avoir une condition de franchissement dépendant des variables continues ; elle peut entraîner la création d'un événement, sans répercussion directe sur les conditions de franchissement ;
- ❖ un jeu d'équations continues peut aussi être associé à un marquage où au moins deux places sont actives.
- ❖ les places peuvent par ailleurs correspondre seulement à des opérations discrètes.

1.6. Pour quoi simuler ?

À l'heure actuelle, les recherches concernant les méthodes et outils formels relatifs à l'analyse du comportement des systèmes hybrides et à la synthèse de leurs lois de commande en sont encore à leur début. La simulation reste donc un passage obligé lorsque l'on envisage :

- ❖ D'aider à la conception ou au dimensionnement d'une installation et de valider cette conception;
- ❖ D'optimiser des procédés existants;
- ❖ D'ajuster la recette de fabrication aux contraintes de production (disponibilité des ressources humaines et du procédé, respect des contraintes temporelles, vérification des objectifs de qualité);
- ❖ De valider le modèle élaboré (dans un but de prévision, par exemple) pour une installation existante;
- ❖ De valider la commande conçue pour une installation;

La simulation a ainsi pour objectif de détecter les erreurs dans les modèles élaborés avant d'implanter la commande correspondante. Il ne faut cependant pas perdre de vue qu'une simulation n'est qu'une représentation de la réalité et non la réalité elle-même... avec toutes les précautions que cela induit lors des premiers essais de fonctionnement !

Peu importe la modélisation choisie pour décrire le comportement dynamique de la partie continue ou celle utilisée pour décrire la partie discrète, les tâches inhérentes à la simulation d'un système hybride seront toujours les mêmes :

- ❖ Simuler le fonctionnement de la partie discrète;
- ❖ En déduire le ou les modèles actifs pour la partie continue;
- ❖ Résoudre les équations différentielles ou aux différences;

1.7. Simulation des SDH

Les difficultés d'analyse et de synthèse des systèmes dynamiques hybrides font de leur simulation un outil d'étude privilégié. Les logiciels de simulation se sont développés depuis une trentaine d'années, initialement pour traiter des processus particuliers : physique nucléaire, convertisseurs statiques. Ces logiciels sont en évolution constante, ils tendent à simuler des systèmes généraux et sont passés aujourd'hui du domaine de la recherche aux applications industrielles. Les programmes de simulation sont très variés, ils dépendent souvent des modèles utilisés pour représenter le système dynamique hybride (Réseaux de Petri hybride, automate hybride,...etc.), de leur objectif (orienté vers une classe d'applications ou au contraire à vocation générale), de la manière dont ils ont été construits (à partir d'un logiciel orienté vers la simulation des systèmes discrets, ou d'un logiciel destiné à simuler des processus continus, ou encore directement pensés pour simuler un SDH).

De nombreux outils graphiques (logiciels) ont été développés pour modéliser les réseaux de Petri. Nous nous sommes concentrés sur le logiciel *SIRPHYCO* (SIMulateur de Réseau de Petri HYbrides et COntinus), c'est un Logiciel de simulation de réseaux de Petri

hybrides et continus, il est facile à utiliser, sa conception de la simulation est facile, rapide et simple c'est pour ça qu'on l'a choisi pour faire notre simulation.

1.8. Conclusion

Nous avons organisé ce chapitre dans le but de donner une vue sur le principe de la modélisation et la simulation des systèmes hybrides. En effet, nous avons commencé par la description des SDH en donnant les définitions de base, suivis de la structure des SDH. Nous avons consacré la suite aux différentes approches de modélisation utilisées dans le développement du modèle de ces systèmes. Dans ce volet, nous avons fait un tour d'horizon sur les automates hybrides, réseaux de petri mixtes et nous nous sommes concentrés sur les réseaux de Petri hybride. Ce dernier est l'outil exploité dans l'élaboration du modèle des SDH dans notre travail.

Enfin, ce chapitre présente une vue générale sur les systèmes dynamiques hybrides. Bien entendu, cet état de l'art n'est pas complet, mais nous avons essayé d'être plus au moins objectifs dans le contenu. Grâce à cette étude, nous avons retenu les réseaux de Petri hybride comme outil de modélisation des SDH. L'objectif du prochain chapitre est de présenter les réseaux de petri discrets et leurs extensions aux Réseaux de Petri Continus et arrivons aux Réseaux de Petri Hybrides.

Chapitre 2

Les Réseaux de Petri

2.1. Introduction

Les Réseaux de Petri ont été inventés par *Carl Adam Petri* au début des années soixante. Des travaux ultérieurs ont permis de développer les Réseaux de Petri comme un outil de modélisation des systèmes à variables d'entrée, de sortie et d'état discrets. C'est aussi un outil de modélisation pour les systèmes à variables logiques puisque ceux-ci sont un cas particulier des systèmes à variables discrètes.

Un caractère très intéressant est que les modèles RdP sont sous la forme d'une représentation mathématique graphique. Ce point est important car le fait d'écrire sous forme graphique un modèle plutôt que sous forme d'équations peut permettre de le rendre lisible par des personnes dont la formation scientifique n'est pas forcément poussée. Le modèle RdP a d'ailleurs donné naissance au Grafcet, un langage de spécification et de programmation d'automates industriels. Dans le langage Grafcet, le caractère graphique est ici fondamental car ce langage doit être accessible à la fois à l'industriel, l'ingénieur, le technicien et l'ouvrier car c'est un outil graphique. De plus, l'un des avantages des RdP, par rapport aux autres formalismes du même type, est qu'ils reposent sur des fondements théoriques permettant de vérifier les propriétés générales d'un modèle (vérifier que le modèle est vivant, sans blocage, borné, etc.) ainsi que l'accessibilité de certains marquages. Les méthodes de recherche de propriétés dans les réseaux de Petri sont basées non seulement sur l'élaboration du graphe des marquages accessibles comme le cas des automates, mais aussi sur l'algèbre linéaire (calcul des invariants de places et de transitions).

Dans un premier instant, les Réseaux de Petri ont été largement utilisés pour modéliser les systèmes discrets (systèmes de calcul, de production, de communication...etc.), car le marquage d'une place peut correspondre, soit à l'état booléen d'un dispositif (par exemple, une ressource est disponible ou pas), soit à un nombre entier (par exemple le nombre de pièces dans un stock).

Dans ce contexte, nous commençons en premier temps par les notions de base des RdP discrets, suivis d'une représentation des RdP continus. Enfin, nous terminerons par l'outil exploité dans l'élaboration du modèle des SDH dans notre travail les RdP hybrides.

2.2. Outil de modélisation

Le couplage d'un modèle en RdP avec une représentation d'activité continue a été étudié par plusieurs auteurs qui ont modifié la théorie de base en ajoutant de nouvelles places et de nouvelles transitions destinées à représenter l'évolution linéaire ou exponentielle d'une variable continue. L'objectif initial du RdP continu était de trouver une représentation réduite d'un système à événement discret lorsque le nombre de jetons devenait trop important et de raisonner en termes de flux. Ils ont étendu cette approche à des systèmes caractérisés par des phénomènes d'accumulation de pièces et régis par des lois de comportement des places dont les transitions sont contrôlées par le calcul d'une équation d'état. Dans les travaux de Alla et David, ont été développés à partir des RdP continus, les *RdP hybrides*. Ces réseaux contiennent des places et des transitions discrètes (représentées graphiquement avec un contour simple) et continues (représentées avec un contour double). Ainsi, la partie continue permet de modéliser les flux continue et la partie discrète les fonctionnements logiques (vanne ouvert/fermée par exemple).

2.3. Réseaux de Petri discrets

2.3.1. Notions de base

2.3.1.1. Définition

Un réseau de Pétri est un moyen de:

- ❖ Modélisation du comportement des systèmes dynamiques à événements discrets.
- ❖ Description des relations existantes entre des conditions et des évènements.

2.3.1.2. Places, transitions et arcs

Un RdP est composé de places, transitions et arcs, la place (P) est représentée par un cercle, la transition (T) par un trait et l'arc étiqueté par une valeur (ou un poids), il relie soit une place à une transition ou une transition à une place (Figure. 2.1.).



Figure. 2.1. Un Réseau de Petri (RdP).

2.3.1.3. Marquage

Chaque place contient un nombre entier positif ou nul de marques (ou jetons). Le marquage M définit l'état du système décrit par le réseau à un instant donné. C'est un vecteur colonne de dimension de nombre de places dans le réseau. Le $i^{ème}$ élément du vecteur correspond au nombre de jetons contenus dans la place P_i (Figure. 2.2.).

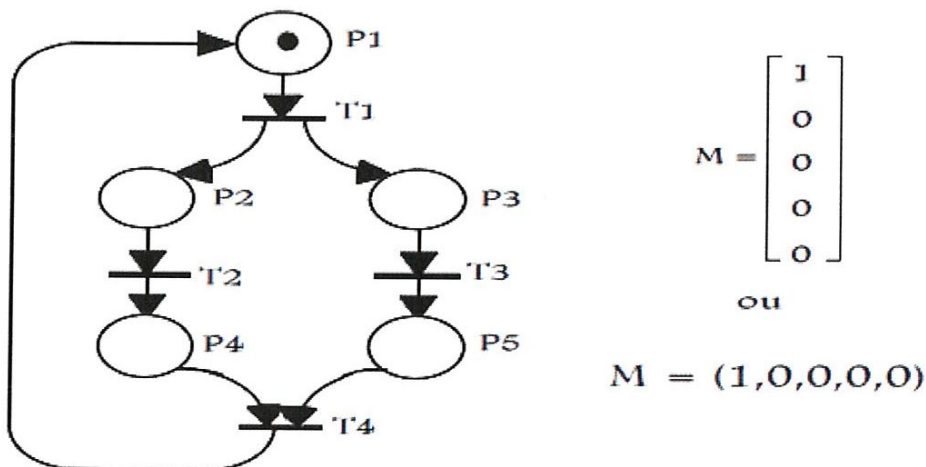


Figure. 2.2. Marquage.

2.3.1.4. *Franchissement d'une transition*

Le franchissement consiste à retirer un jeton de chacune des places d'entrée et à rajouter un jeton à chacune des places de sortie de la même transition, prenons l'exemple 1 (Figure. 2.3.a.), Le franchissement de T1 consiste à enlever un jeton de P1 et un jeton de P2 et à rajouter un jeton dans P3 et un jeton dans P4.



Figure. 2.3.a. – Exemple 1 : Franchissement d'une transition.

Prenons l'exemple 2, la transition T1 ne peut pas être franchie car la place P2 ne contient aucun jeton (Figure. 2.3.b.).

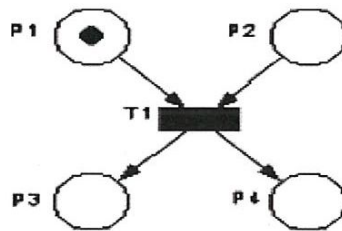


Figure. 2.3.b. – Exemple 2 : Transition non Franchissable.

Une transition sans place d'entrée est toujours franchissable : c'est une transition source. Le franchissement d'une transition source consiste à rajouter un jeton à chacune de ces places de sortie (Figure. 2.3.c.).

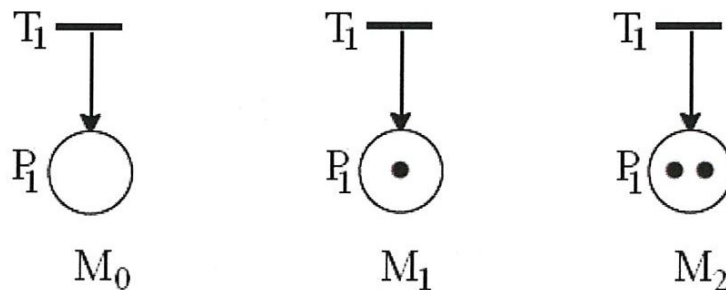


Figure. 2.3.c. – Exemple 3 : transition source.

Une transition sans place de sortie est une transition puits. Le franchissement d'une transition puits consiste à retirer un jeton de chacune de ses places d'entrée (Figure. 2.3.d.).

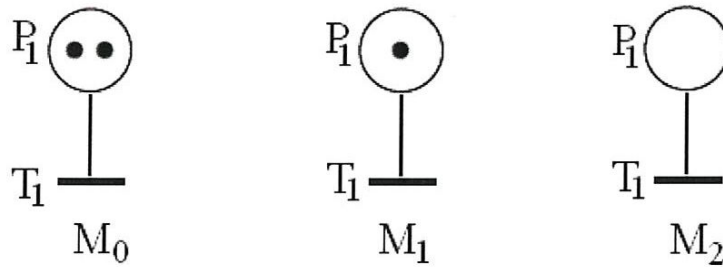


Figure. 2.3.d. – Exemple 4 : transition puits.

2.3.1.5. Séquence de franchissement

Une séquence de franchissement s est une suite de transitions $T_1 T_2 \dots T_k$ qui peuvent être franchies successivement à partir d'un marquage donné. Une seule transition peut être franchie à la fois (Figure. 2.4.).

On note : $M_i [s \rightarrow M_j$ ou $M_i [s > M_j$: à partir du marquage M_i , le franchissement de la séquence s aboutit au marquage M_j .

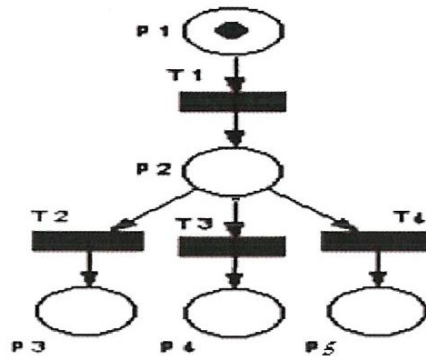


Figure. 2.4. Séquence de franchissement.

$T_1 T_2$, $T_1 T_3$ et $T_1 T_4$ sont deux séquences de franchissement:

$$M_0 [T_1 T_2 \rightarrow M_1, M_0 [T_1 T_3 \rightarrow M_2 \text{ et } M_0 [T_1 T_4 \rightarrow M_3 \text{ avec } M_1 = [00100]^T, M_2 = [00010]^T \text{ et } M_3 = [00001]^T$$

2.3.1.6. Marquages accessibles

L'ensemble des marquages accessibles est l'ensemble des marquages M_i qui peuvent être atteint par le franchissement d'une séquence s à partir du marquage initial M_0 . On le note $*M_0$ (Figure. 2.4.).

$$*M_0 = \{M_i \text{ tel que } M_i [s \rightarrow M_j \}$$

$$*M_0 = \{ M_0 \ M_1 \ M_2 \ M_3 \ M_4 \} \text{ avec } M_0 = [10000]^T ; M_1 = [01000]^T ; M_2 = [00100]^T ;$$

$$M_3 = [00010]^T \text{ et } M_4 = [00001]^T$$

2.3.1.7. Graphe de marquages

On utilise le graphe de marquages quand le nombre de marquages accessibles est fini.

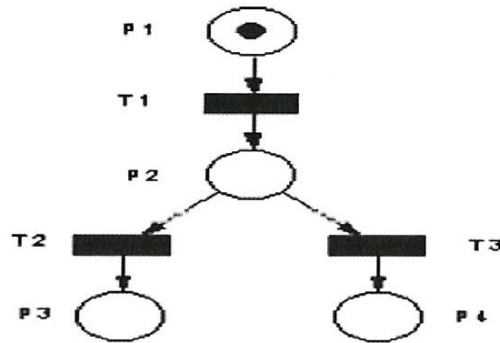


Figure. 2.5.a. graphe de marquages.

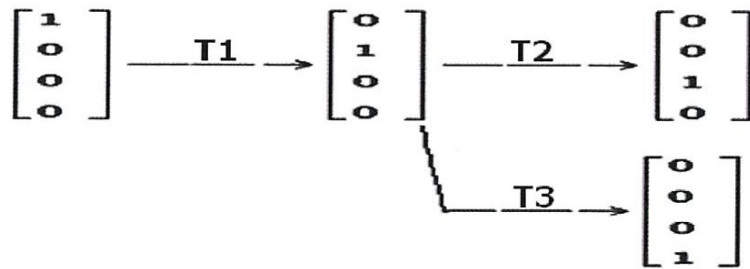


Figure. 2.5.b. Graphe de marquages correspondant.

2.3.2. RdP Particuliers

2.3.2.1. Graphe d'état

Un réseau de Petri non marqué est un graphe d'état si et seulement si toute transition a exactement une seule place d'entrée et une seule place de sortie (Figure. 2.6.).

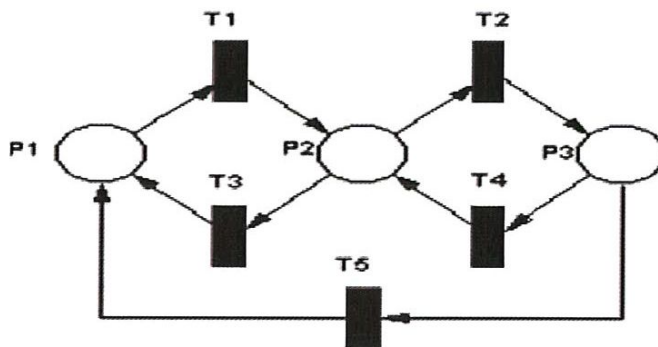


Figure. 2.6. graphe d'état.

Chacune des transitions T_1 , T_2 , T_3 , T_4 et T_5 possède une seule place d'entrée et une seule place de sortie.

2.3.2.2. Graphe d'événement

Un RdP est un graphe d'événement si et seulement si, chaque place possède exactement une seule transition d'entrée et une seule transition de sortie (Figure. 2.7.a.).

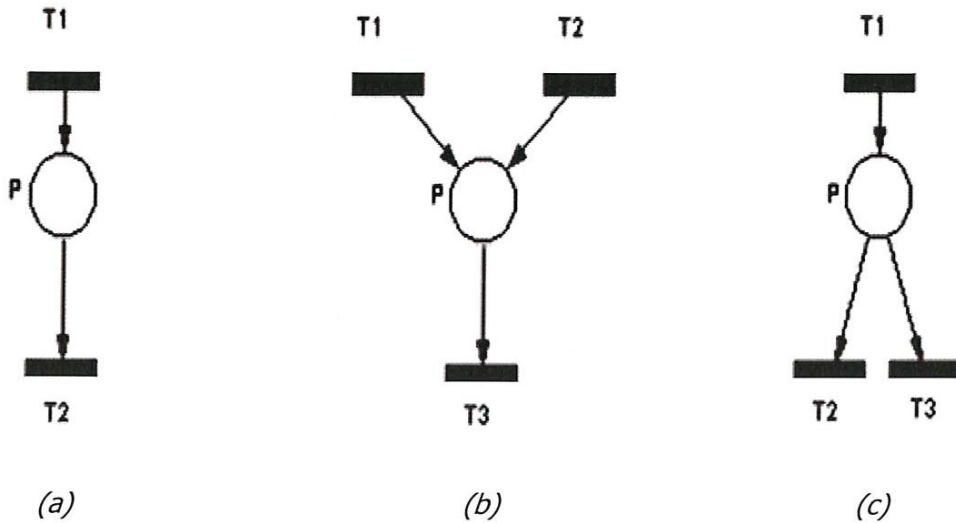


Figure. 2.7. (a) Graphe d'événements, (b) et (c) non un graphe d'événements.

2.3.2.3. RdP sans conflit

Un RdP sans conflit est un réseau dans lequel chaque place a au plus une transition de sortie (Figure. 2.8.a.).

Un RdP avec conflit est un réseau qui possède donc une place avec au moins deux transitions de sorties (Figure. 2.8.b.). Un conflit est noté: $[P_i, \{T_1, T_2, \dots, T_n\}]$; avec T_1, T_2, \dots, T_n étant les transitions de sorties de la place P_i .

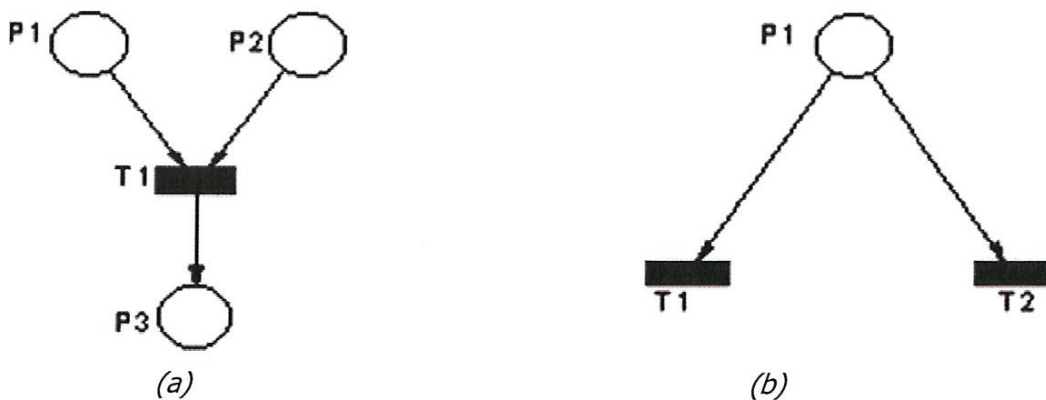


Figure. 2.8. RdP. (a) Sans conflit, (b) Avec conflit.

2.3.2.4. RdP généralisés

Un RdP généralisé est un RdP dans lequel des poids (nombres entiers strictement positifs) sont associés aux arcs (*Figure. 2.9.*).

- ❖ Si un arc (P_i, T_j) a un poids k : la transition T_j n'est franchie que si la place P_i possède au moins k jetons. Le franchissement consiste à retirer k jetons de la place P_i .
- ❖ Si un arc (T_j, P_i) a un poids k : le franchissement de la transition rajoute k jetons à la place P_i .
- ❖ Lorsque le poids n'est pas signalé, il est égal à un par défaut.

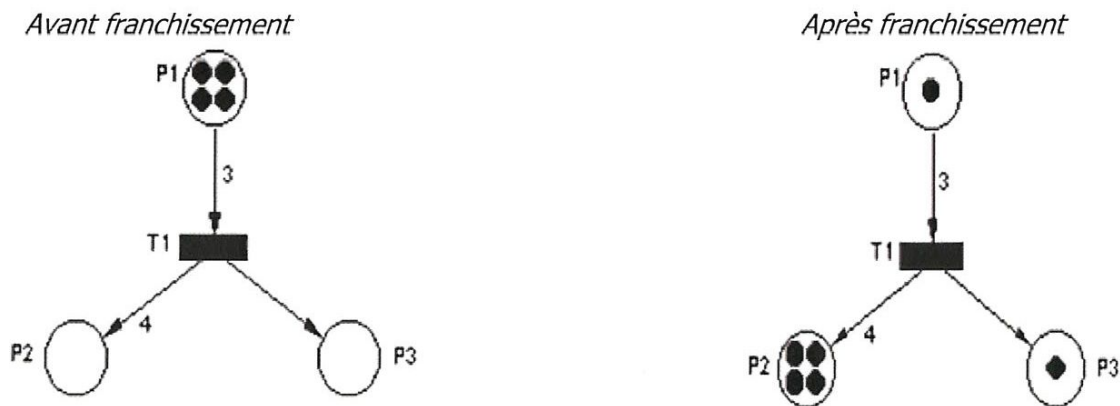


Figure. 2.9. RdP généralisé.

2.3.2.5. RdP à capacités

Un RdP à capacités est un RdP dans lequel des capacités (nombres entiers strictement positifs) sont associées aux places. Le franchissement d'une transition d'entrée d'une place P_i dont la capacité est $cap(P_i)$ n'est possible que si le franchissement ne conduit pas à un nombre de jetons dans P_i qui est plus grand que $Cap(P_i)$ (*Figure. 2.10.*).

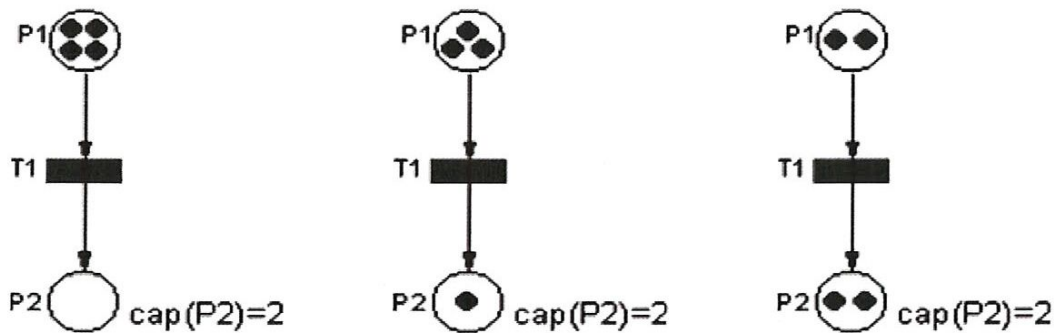


Figure. 2.10. RdP à capacités.

Le franchissement de T_1 conduit à 3 jetons dans P_2 d'où T_1 ne peut plus être franchie.

2.3.2.6. RdP à priorités

Dans un tel réseau si on atteint un marquage tel que plusieurs transitions sont franchissables, on doit franchir la transition qui a la plus grande priorité (Figure. 2.11.).

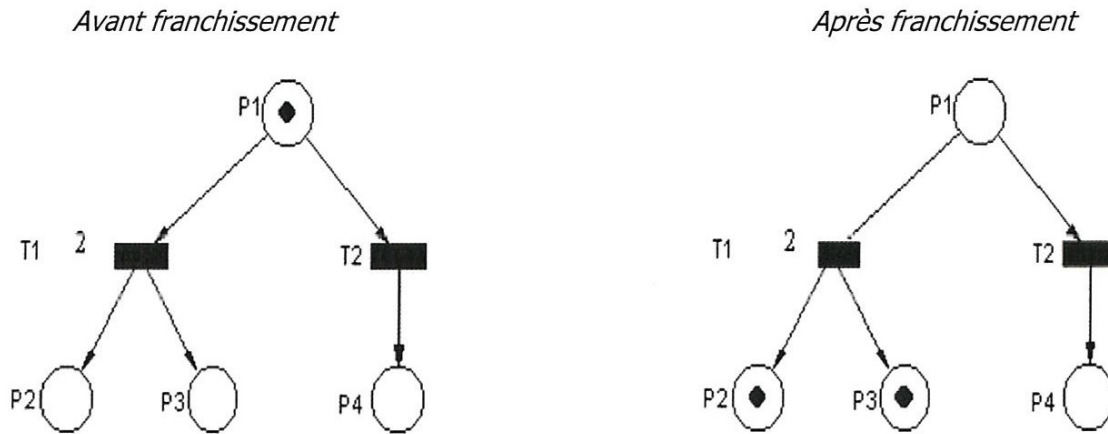


Figure. 2.11. RdP à priorité.

2.3.3. Algèbre Linéaire

2.3.3.1. Notations et définitions

" $\text{pré}(P_i, T_j)$ " est le poids "k" de l'arc reliant une place à une transition.

$$\text{pré}(P_i, T_j) = \begin{cases} k & \text{si l'arc } (P_i, T_j) \text{ existe} \\ 0 & \text{sinon} \end{cases}$$

" $\text{post}(P_i, T_j)$ " est le poids "k" de l'arc reliant une transition à une place .

$$\text{post}(P_i, T_j) = \begin{cases} k & \text{si l'arc } (T_j, P_i) \text{ existe} \\ 0 & \text{sinon} \end{cases}$$

on appelle "matrice d'incidence avant" :

$$W^- = [\text{pré}(P_i, T_j)]$$

on appelle "matrice d'incidence arrière" :

$$W^+ = [\text{post}(P_i, T_j)]$$

on appelle "matrice d'incidence " :

$$W = W^+ - W^-$$

Dans ces matrices les transitions représentent les colonnes et les places représentent les lignes.

2.3.3.2. Equation fondamentale ou equation d'état

- ❖ Soit S une séquence de franchissement réalisable à partir d'un marquage $M_i : M_i [S > M_k$.
- ❖ Soit \underline{S} le vecteur caractéristique de la séquence S : c'est un vecteur de dimension m égale au nombre de transitions dans le réseau. Sa composante numéro j correspond au nombre de fois où la transition T_j est franchie dans la séquence S.

Exemple 2.1. : si $S=T_2T_4T_1T_4T_2T_4$ alors $S=[1, 2, 0, 3]^T$

- ❖ Si la séquence de franchissement S est tel que $M_i [S > M_k$ alors l'équation fondamentale correspondante s'écrit.

$$M_k = M_i + W * \underline{S} \tag{2.1}$$

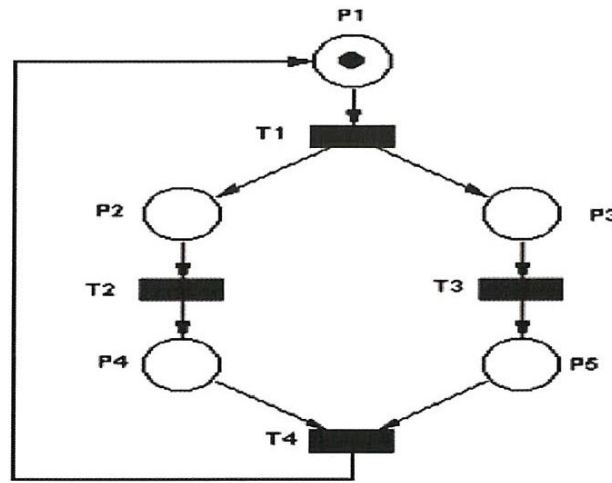


Figure. 2.12. RdP (équation fondamentale).

Soit la séquence $S= T_2$ donc $S = [0, 1, 0, 0]^T$

$$W^- = \begin{matrix} & T_1 & T_2 & T_3 & T_4 \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & P_1 \\ & P_2 \\ & P_3 \\ & P_4 \\ & P_5 \end{matrix}$$

- La matrice d'incidence avant -

$$W^+ = \begin{matrix} & T_1 & T_2 & T_3 & T_4 \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & P_1 \\ & P_2 \\ & P_3 \\ & P_4 \\ & P_5 \end{matrix}$$

- La matrice d'incidence arrière -

$$W = W^+ - W^- = \begin{bmatrix} T_1 & T_2 & T_3 & T_4 \\ -1 & 0 & 0 & +1 \\ +1 & -1 & 0 & 0 \\ +1 & 0 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{bmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{matrix}$$

- La matrice d'incidence -

L'équation fondamentale correspondante à cette séquence est :

$$M_k = M_i + W * \underline{S}$$

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

M_i W \underline{S} M_k

2.3.4. RdP temporisés

Les RdP temporisés permettent de décrire un système dont le fonctionnement dépend du temps. Par exemple, il peut s'écouler une certaine durée entre le début d'une opération et la fin de cette opération. Si une marque dans une certaine place indique que cette opération est en cours, un RdP temporisé permettra de rendre compte de cette durée. Les RdP temporisés sont utiles pour l'évaluation de performances des systèmes. Il y a principalement deux façons de modéliser la temporisation : Soit la temporisation est associée à la transition (on dira que l'on a un RdP T-temporisé), soit la temporisation est associée à la place (on dira que l'on a un RdP P-temporisé). Cependant, les deux modèles sont équivalents et on peut passer de l'un à l'autre.

2.3.4.1. Réseaux de Petri P-temporisés

On associe une temporisation (valeur rationnelle positive) à chaque place. On notera d_i la temporisation de la place P_i , lorsqu'une marque arrive dans une place temporisée, on dit qu'elle est indisponible pendant un temps d_i . Quand le temps est écoulé, la marque devient disponible. On parle de fonctionnement à vitesse maximale (Figure. 2.13.a.).

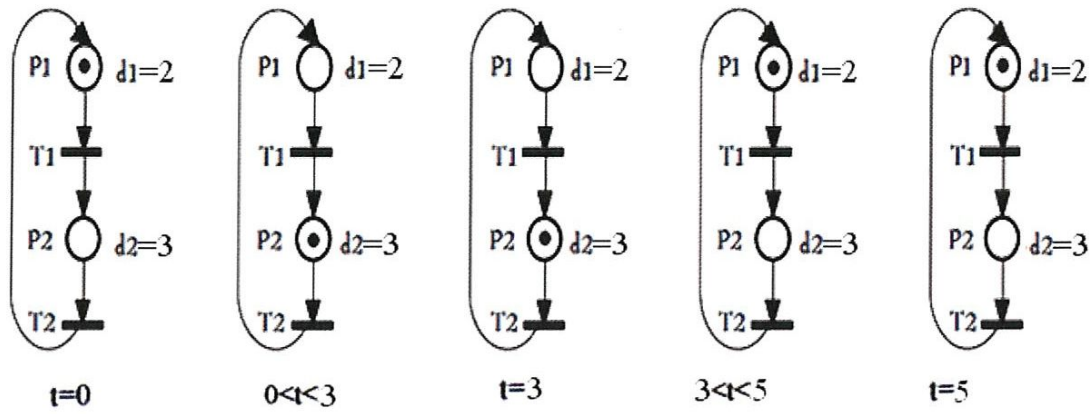


Figure. 2.13.a. Le fonctionnement à vitesse maximale du RdP P-temporisé.

2.3.4.2. Réseaux de Petri T-temporisés

On associe cette fois la temporisation aux transitions (Figure. 2.13.b.). Une marque peut avoir deux états :

- disponible (ou non réservée).
- ou bien réservée pour le franchissement d'une transition.

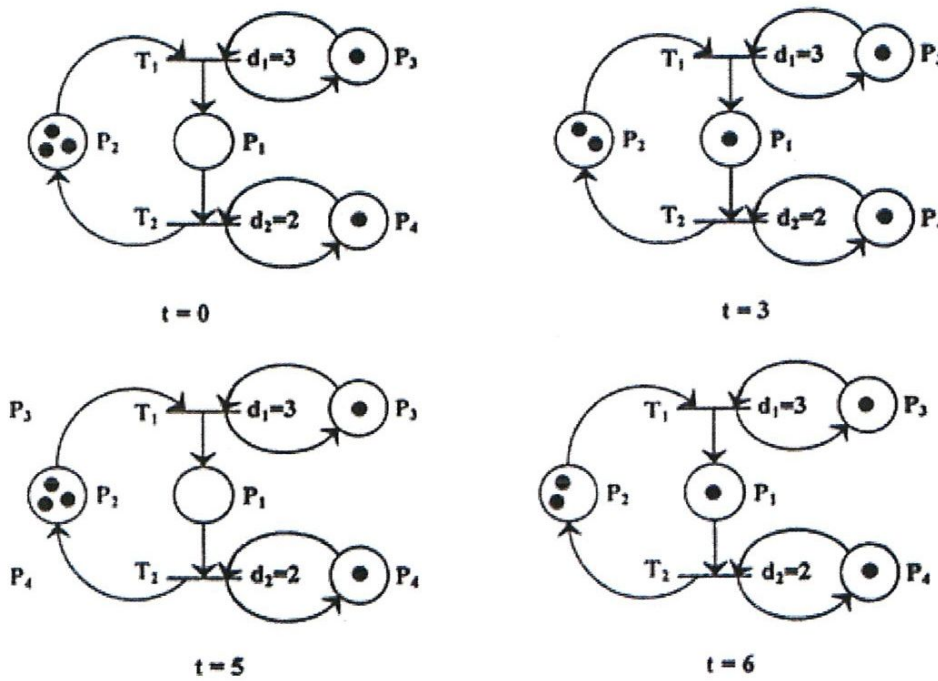


Figure. 2.13.b. Le fonctionnement à vitesse maximale du RdP T-temporisé.

Dans la suite, nous utilisons le mot RdP temporisé pour désigner un RdP T-temporisé.

2.3.4.3. RdP T-temporels

Les RdP T-temporels (Time Petri nets) ont été introduits en même temps que les RdP temporisés par *Merlin*. L'idée fondatrice des RdP T-temporels est d'associer un intervalle de temps $[\alpha_j, \beta_j]$ à chaque transition T_j . Si cette dernière est validée de façon continue pendant au moins α_j unités de temps, elle peut être franchie. De plus si elle est validée pendant β_j unités de temps de manière continue, elle doit être franchie. On trouve ici les notions de délais minimum et maximum dans un état au lieu de durée d'un état dans les modèles temporisés.

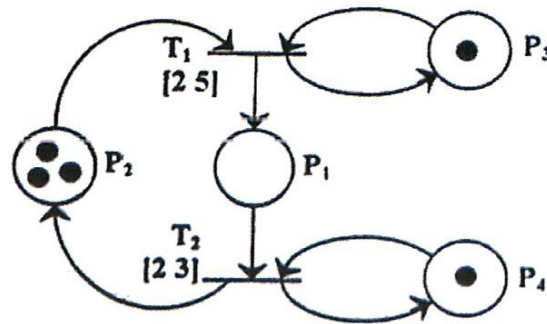


Figure. 2.14. Le modèle du RdP T-temporel.

2.4. Modélisation par RdP

Les RdP permettent de représenter graphiquement certaines relations, de visualiser certaines notions...etc. En voici quelques exemples :

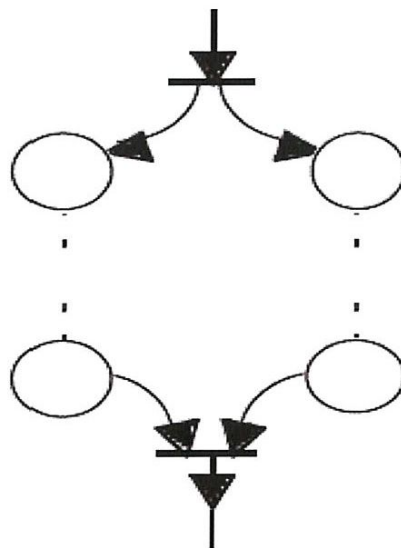


Figure. 2.15. Parallélisme.

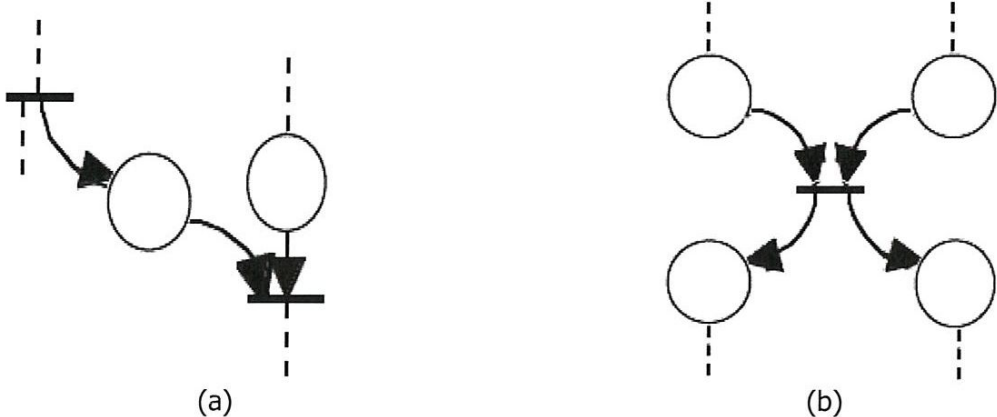


Figure. 2.16. (a) Synchronisation sémaphore, (b) Synchronisation rendez-vous

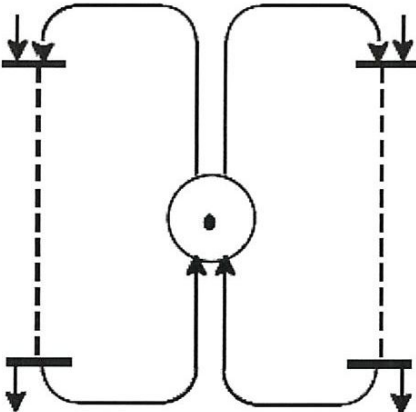


Figure. 2.17. Partage de ressource.

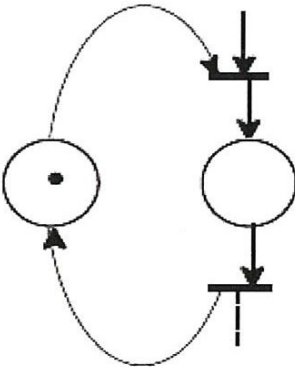


Figure. 2.18. Capacité limitée.

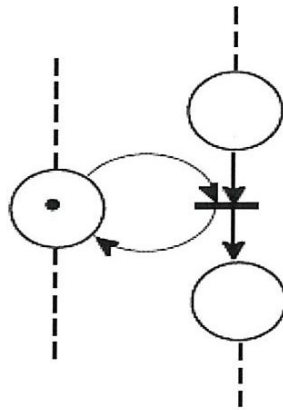


Figure. 2.19. Lecture.

2.5. Les RdP du discret aux Continu

Dans un RdP, le marquage d'une place comparable à une variable booléenne peut correspondre à l'état d'un appareil, par exemple une machine est disponible ou n'est pas disponible. Un marquage peut également être associé à un nombre entier, par exemple le nombre de pièces dans le stock d'entrée d'une machine (Figure. 2.20.).

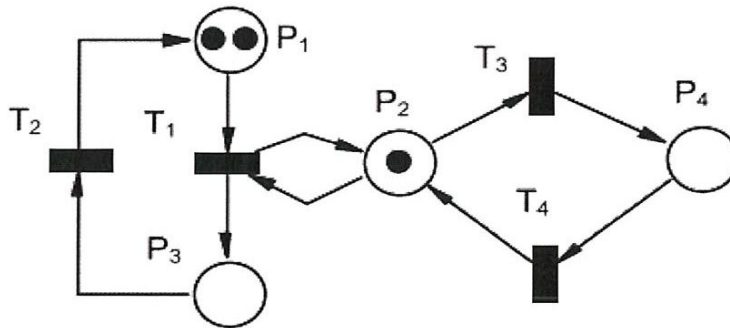


Figure. 2.20. Exemple d'un Réseau de Petri discret.

De manière plus formelle, un Réseau de Petri est un $PN = (P, T, A, W, M_0)$, où $P = \{P_1, P_2, \dots, P_n\}$ est un ensemble fini de places, $T = \{T_1, T_2, \dots, T_q\}$ un ensemble fini de transitions, $A(P, T)$ ou (T, P) est l'ensemble fini des arcs, $W: A \rightarrow \{1, 2, \dots\}$ la fonction poids des arcs et $M_0: P \rightarrow \{0, 1, 2, \dots\}$ est le marquage initial du réseau.

Une méthode générale d'analyse pour ces types de réseaux (places, transitions) consiste à déterminer tous les états atteignables et à en déduire diverses propriétés du système.

De plus, les Réseaux de Petri temporisés sont bien adaptés pour évaluer quantitativement un système. Cependant un modèle peut contenir un grand nombre de marques ou bien on peut y trouver des durées n'ayant pas le même ordre de grandeur. Par exemple, le temps de production d'une pièce est en général beaucoup plus court que le temps moyen entre deux pannes d'une machine. Dans ce cas le nombre d'états accessibles explose. Ce phénomène constitue une limite pratique à l'utilisation des réseaux discrets. Cela a conduit à introduire la notion de réseaux de Petri continus (RdPC) où le marquage devient un nombre réel positif. Les réseaux de Petri continus étendent le marquage dans l'espace d'état des réels en fonction du taux de franchissement des transitions.

2.6. Réseaux de Petri Continues

2.6.1. Introduction

Le Réseau de Petri continu défini comme un cas limite de réseau discret est un modèle dans lequel les nombres de marques dans les places sont des nombres réels. Ce modèle qui s'apparente aux Réseaux de Petri temporisés sur les transitions a été introduit en considérant que le franchissement d'une transition se fait de façon continue avec une certaine vitesse $v(t)$, (Figure 2.21.). Cela signifie qu'entre l'instant t et l'instant $t + dt$, une quantité de marque vdt est retirée aux places en amont de la transition et une quantité de marques vdt est ajoutée aux places en aval de cette transition (cas des réseaux ordinaires).

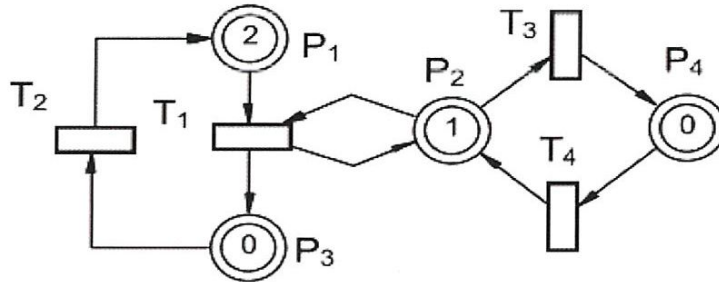


Figure. 2.21. Exemple d'un Réseau de Petri continu.

Dans cet état les transitions T_1 et T_3 sont validées, c'est-à-dire franchissables, puisque les marquages de leurs places d'entrée ne sont pas nuls. Par exemple, la transition T_1 peut être franchie mais on définit maintenant une quantité de franchissement qui est un nombre réel compris dans l'intervalle continu $[0, 1]$; la valeur maximale 1 dans ce cas correspond à m_2 (qui est le minimum de m_1 et m_2). Pour une quantité de franchissement 0.2 on atteint le marquage $(1.8, 1, 0.2, 0)$.

Dans un Réseau de Petri continu, une séquence de franchissements S à partir d'un marquage M implique une trajectoire correspondant à une suite de marquages successifs. Le vecteur caractéristique s d'une trajectoire est un vecteur dans lequel chaque composante est un nombre réel correspondant à une quantité de franchissements de la transition correspondante. Donc, la relation fondamentale pour un réseau de Petri continu est identique à la relation fondamentale pour un réseau discret. Donc, toute propriété des réseaux discrets basée sur cette relation peut être transposée aux réseaux continus.

2.6.2. Les Réseaux de Petri Continus à Vitesses Constantes

2.6.2.1. Définition et notations

Le réseau de Petri continu à vitesses constantes (RdPCC) est le réseau de Petri continu de base. Il est le modèle le plus simple, il permet de modéliser des systèmes aussi bien continus que discrets.

Définition 2.1. un RdP continu à vitesses constantes (RdPCC) marqué est un sextuplé .

$R = \langle P, T, V, \text{Pré}, \text{Post}, \text{Mo} \rangle$ tel que :

- P : est ensemble fini et non vide de places.
- T : est un ensemble fini et non vide de transitions.

$P \cap T = \emptyset$, i.e les ensembles P et T sont disjoints.

- Pré : est l'application d'incidence avant.
- Post : est l'application d'incidence arrière.
- M_0 : est le marquage initial.
- $V : T \rightarrow \mathbb{R}^+ \cup \{\infty\}$ est l'application qui associe une vitesse maximale à chaque transition.

Le marquage initial M_0 est un vecteur de nombres réels positifs. Sachant que $M(t)$ représente le marquage du RdP à l'instant t , le marquage initial peut être noté $M(0) = M_0$. Quant à la vitesse $V(T_j) = V_j$ elle correspond à la vitesse maximale de franchissement de T_j . La matrice d'incidence W est égale à (Post-Pré). Cette évolution est donnée par la relation fondamentale suivante :

$$\frac{dM}{dt} = W.v(t) \tag{2.2}$$

Où $v(t)$ est le vecteur des vitesses instantanées de franchissement. Cette relation montre que la variation du marquage par unité de temps est égale au flux entrant moins le flux sortant.

2.6.2.2. Principe de fonctionnement

La figure 2.22.a, représente un réseau de Petri continu à vitesse constante. Les vitesses de franchissement maximales sont $V_1=1/d_1=0.5$ et $V_2=1/d_2=0.33$ sont associées respectivement aux transitions T_1 et T_2 . Notons que ce modèle utilise une notation implicite, les franchissements simultanés d'une même transition sont limités à 1. Tant que P_1 n'est pas vide, la transition T_1 est franchissable à la vitesse $V_1(t)=V_1$, c'est à dire que la quantité $V_1 dt = 0.5dt$ est retirée de P_1 et ajoutée à P_2 entre les instants t et $t+dt$. T_2 peut aussi être franchie à la vitesse $V_2(t)=V_2$ car $V_2 < V_1$. On peut en déduire les marquages de P_1 et P_2 par les relations suivantes :

$$m_1(t) = m_1(0) - V_1(t) + V_2(t) = 2 - t/6 \tag{2.3}$$

$$m_2(t) = m_2(0) + V_1(t) - V_2(t) = t/6$$

Ces relations restent vraies jusqu'à $t=12$. Pour $t > 12$, la vitesse de franchissement de T_1 est limitée par la vitesse d'alimentation de P_1 qui vaut $V_2(t)=V_2$ donc $V_1(t)=V_2$. Pour $t > 12$, $m_1(t)=0$ et $m_2(t)=2$. Il existe deux types de validation. La transition T_1 est faiblement validée : Validée car $v_1(t) > 0$, faiblement car $m_1(t) = 0$. La transition T_2 est fortement validée car $m_2(t) > 0$. L'évolution du marquage est illustrée sur La figure 2.22.b.

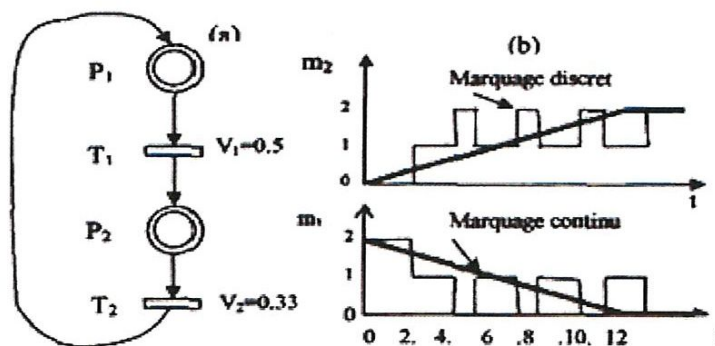


Figure. 2.22. (a). Réseau de Petri continu, (b) Evolution du marquage.

2.6.3. Les autres modèles continus

2.6.3.1. Le RdP Continu à Vitesses Variables

Dans ce modèle la vitesse maximale de franchissement d'une transition dépend du marquage des places en amont de la transition et d'une valeur constante associée à la transition. Les marquages et les vitesses sont donc des fonctions continues du temps, alors que dans le RdP continu à vitesse constante cette propriété est vraie pour les marquages mais pas pour les vitesses qui sont simplement constantes par morceaux. Ce modèle fournit une meilleure approximation du RdP discret, en particulier lorsque le nombre de marques est petit, mais les simulations sont longues. Ici, le comportement événementiel est perdu, il faut discrétiser le temps pour faire une simulation.

2.6.3.2. Le RdP Continu Asymptotique

Il est possible de distinguer plusieurs phases d'évolution du marquage. Chaque phase est caractérisée par une période durant laquelle le vecteur des vitesses de franchissement est constant. La vitesse de franchissement d'une transition est alors constante par morceaux, et l'évolution du marquage est donc composée de plusieurs phases de fonctionnement. Il s'agit de l'approximation du RdP Continu à Vitesses Variables, mais la simulation est facilitée.

2.7. Réseaux de Petri hybrides (RdPH)

2.7.1. Introduction

Les réseaux continus constituent une bonne approximation lorsqu'il s'agit de modéliser un flux de pièces par exemple. Cependant l'état opérationnel ou en panne d'une machine, ou en règle générale toute condition booléenne ne peut être modélisée par un nombre réel. Ceci a conduit naturellement à définir les réseaux hybrides, réunissant une partie discrète et une partie continue. Ainsi, dans un réseau hybride, on retrouve des places et des transitions discrètes (représentées en simple ligne) et des places et des transitions continues (représentées en double ligne), avec une influence mutuelle des parties discrètes et continues, (Figure 2.23.).

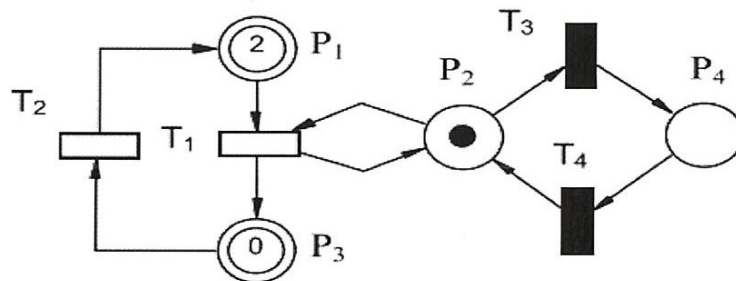


Fig. 2.23. Exemple d'un Réseau de Petri Hybride.

Les places continues sont P_1 et P_3 , les transitions continues sont T_1 et T_2 , les places discrètes P_2 et P_4 , et les transitions discrètes T_3 et T_4 . Les transitions T_1 et T_3 sont validées, donc franchissables. Alors, si la transition continue T_1 sera franchie avec une vitesse constante, pour une quantité de franchissement 0.1 on obtient le marquage $(1.9, 1, 0.1, 0)$. On a retiré donc une quantité de marquage 0.1 de P_1 et P_2 qui sont les places d'entrée, et l'on a ajouté la même quantité à P_3 et P_4 qui sont les places de sortie. On voit que le marquage de la place discrète P_2 est encore un entier, puisque l'on a retiré et ajouté la même quantité.

L'influence de la partie discrète du réseau (du système) sur la partie continue de celui-ci et vice versa décrit le comportement événementiel pour l'entière structure hybride du système. Par exemple, pour un système de deux réservoirs (Figure. 2.24.a) les places P_4 et P_5 qui se vident l'un dans l'autre à travers une vanne (transition T_3) qui peut être ouverte (place discrète P_1 avec un jeton) ou fermée (jeton dans la place P_2), l'état de la place discrète P_1 influence le fonctionnement de la partie continue, (Figure. 2.24.b). Le franchissement continu de la transition T_3 correspond à un écoulement à une vitesse V_3 quand la place P_1 n'est pas vide. Les durées d_1 et d_2 correspondent à des temporisations de la partie discrète. Quand la place P_1 n'est pas marquée, la transition T_3 n'est plus franchie (arrêt de l'écoulement). En cas de conflit entre une transition discrète et une transition continue, la transition discrète est toujours prioritaire. Alors, quand P_1 est marquée, le franchissement d'une quantité $V_3 dt$ de T_3 consiste à retirer $V_3 dt$ marques à P_1 et à ajouter la même quantité à P_1 , donc le marquage de P_1 reste constant à la valeur 1.

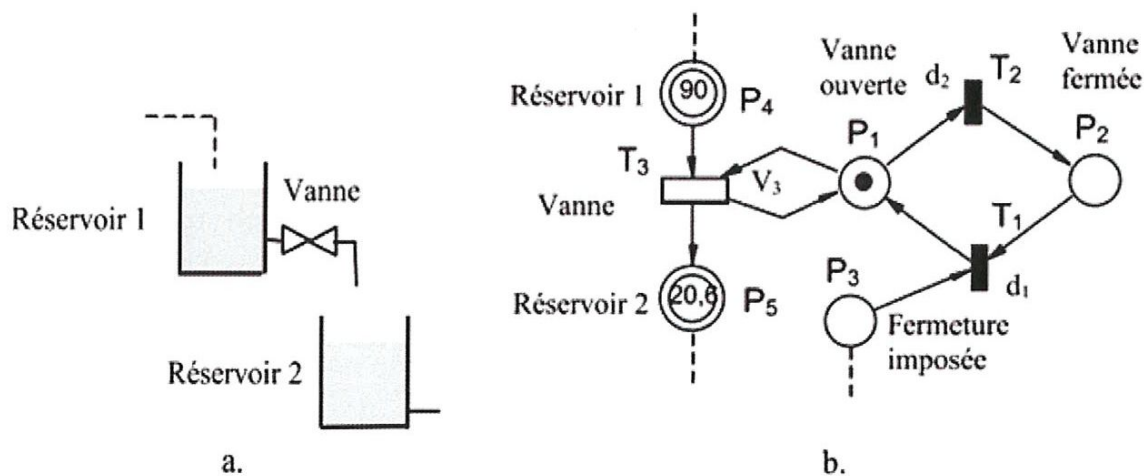


Fig. 2.24. Explicative à la modélisation du système, (a) Le schéma de principe, (b) Le Réseau de Petri hybride.

2.7.2. L'influence de la partie discrète sur la partie continue

La figure. 2.25. suivante modélise l'influence de la partie discrète sur la partie continue ou le franchissement de la transition continue T_3 est conditionné par le marquage de la place discrète P_1 .

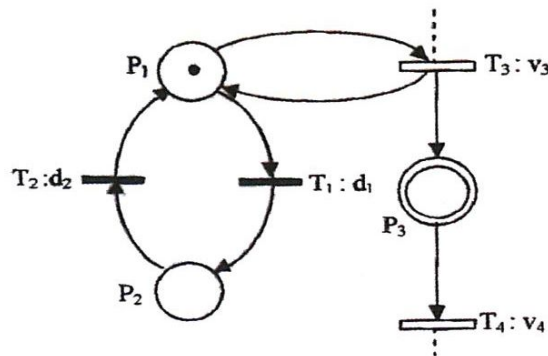


Figure. 2.25. Influence de la partie discrète sur la partie continue.

2.7.3. L'influence de la partie continue sur la partie discrète

L'influence de la partie continue sur la partie discrète (figure. 2.26.) est réalisée via la validation de la transition discrète T_1 par une place continue suivant que son marquage est supérieur ou non au seuil s .

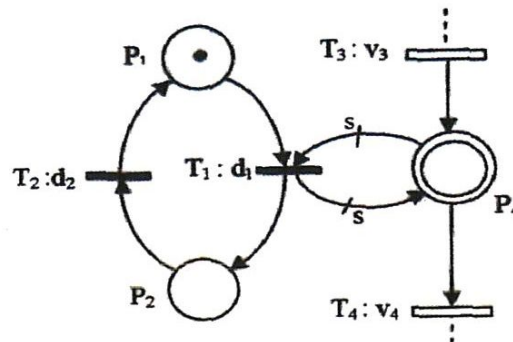


Figure. 2.26. Influence de la partie continue sur la partie discrète.

2.7.4. Partage de ressource

Lorsqu'on a un seul serveur pour deux lignes de production, la modélisation par RdPH est représentée sur La (figure. 2.27.) La place P_1 modélise le serveur partagé par les deux lignes de production.

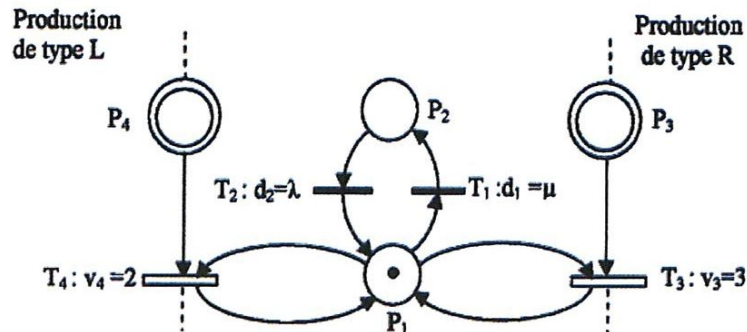


Figure. 2.27. Partage de ressource.

2.7.5. Transformation de marques

La figure. 2.28.a. montre la transformation de marques discrètes en marquages continus par l'intermédiaire de transition discrète. Nous pouvons par exemple, rencontrer cette configuration lots de la modélisation du transport d'un lot de p pièces (nécessitant d_1 unités de temps), ces pièces étant déversées dans le stock d'une machine modélisé par la place P_2 . De façon générale, on trouve cette structure chaque fois qu'un lot, après avoir été traité globalement, doit être transformé en pièces pour l'opération suivante qui réalise un traitement des pièces une à une. Pour ce réseau, dès qu'une marque arrive dans la place P_1 , c'est-à-dire qu'un lot est présent, la transition T_1 est validée. Après la durée d_1 , T_1 est franchie, ce qui consiste à retirer une marque de la place et ajouter p marques dans la places P_2 . Notons au passage que le nombre p est un nombre réel. La temporisation d_1 , qui peut avoir une valeur nulle, représente la durée de la transformation d'un lot en p pièces. La figure. 2.28.b. représente l'opération inverse à celle décrite précédemment. La place P_4 est une place continue, elle est reliée à la transition T_2 . Nous savons donc que le poids de l'arc

est interprété comme une valeur seuil que doit atteindre le marquage de P_4 pour valider la transition T_2 . Dès lors que cette condition est vérifiée et que d_2 unité de temps se sont écoulées, la transition T_2 est franchie, les p marques de P_4 sont retirées et une marque est ajoutée à P_3 . Cette structure sera rencontrée lorsque l'on souhaitera reconstituer un lot à partir de la quantité de pièces le constituant. La *figure. 2.28.c.* représente le franchissement d'une transition continue, la transition T_3 est validée puis franchie selon la vitesse de franchissement V_3 , pendant une unité de temps, trois marque sont retiré de la places P_6 et ajoutée la même quantité à la place P_7 .

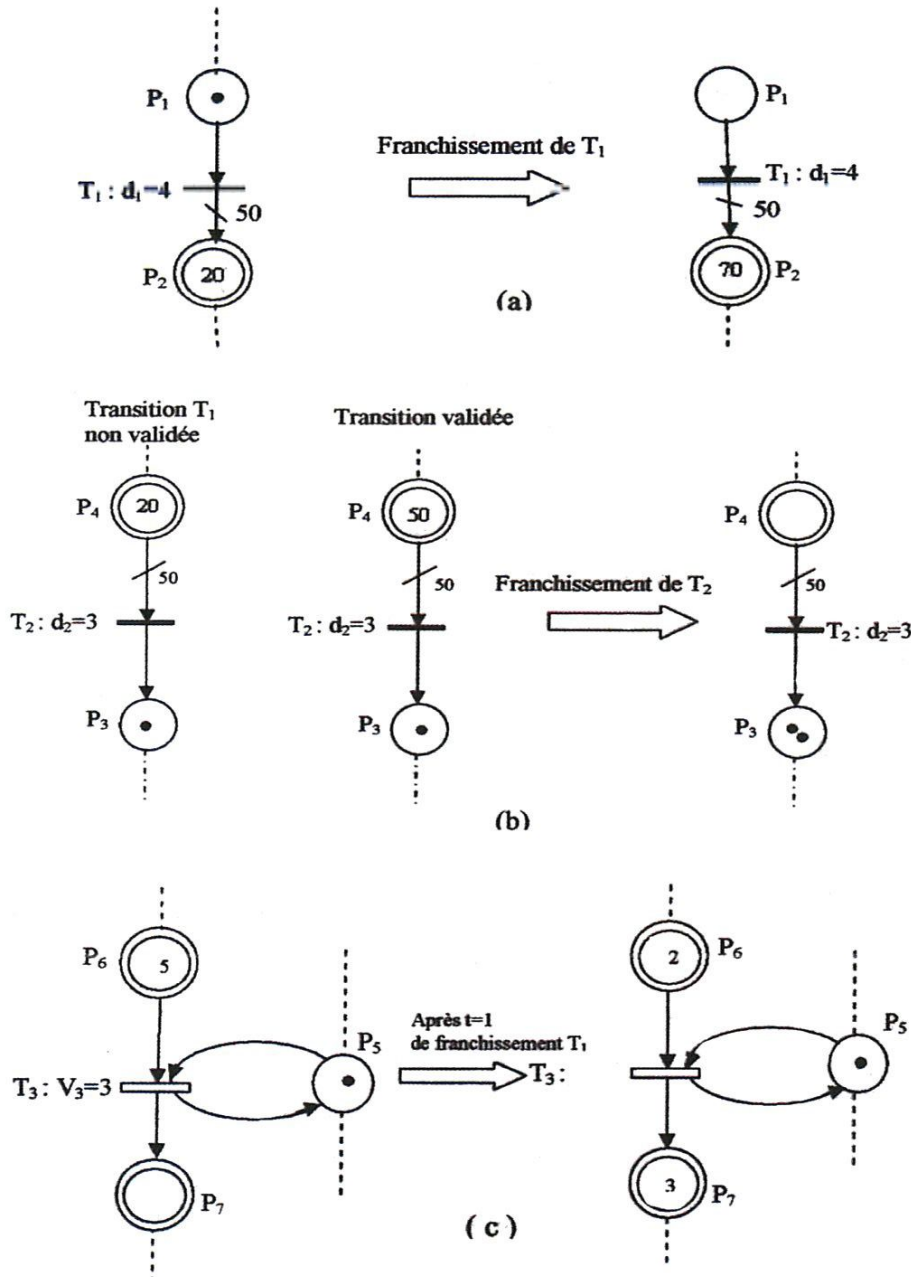


Figure. 2.28. (a) et (b) Transformation de la marque, (c) Franchissement de transition continue.

2.7.6. Test à zéro et un arc de poids 0^+

Le test à zéro d'une place discrète (D-place) dans un RdPH est similaire au test à zéro dans un RdP discret utilisant un arc inhibiteur *figure 2.29.a*. Un arc inhibiteur est un arc qui part d'une place P_i pour aboutir à une transition T_j . L'arc inhibiteur entre P_i et T_j signifie que la transition T_j n'est validée que si la place P_i ne contient aucune marque. Le franchissement consiste à retirer une marque dans chaque place d'entrée de T_j à l'exception de P_i , et ajouter une marque dans chaque place de sortie de T_j .

Le test à zéro d'une place continue (C-place) est différent. On associe à l'arc inhibiteur un poids 0^+ , avec 0^+ est nombre entier positif très petit proche de zéro tel que $0 < 0^+$. Le teste à zéro d'un place continue est illustré sur la *figure. 2.30.a*. La transition T_3 est validée car $m(P_9) = 0 < 0^+$. La transition T_4 dans la *figure. 2.30.b* est non validée car $m(P_{12}) = 0.1 \geq 0^+$. Dans la *figure. 2.26.c.*, la transition T_5 est franchissable seulement si la place P_5 est non vide ie si $m_5(P_5) \geq 0^+$.

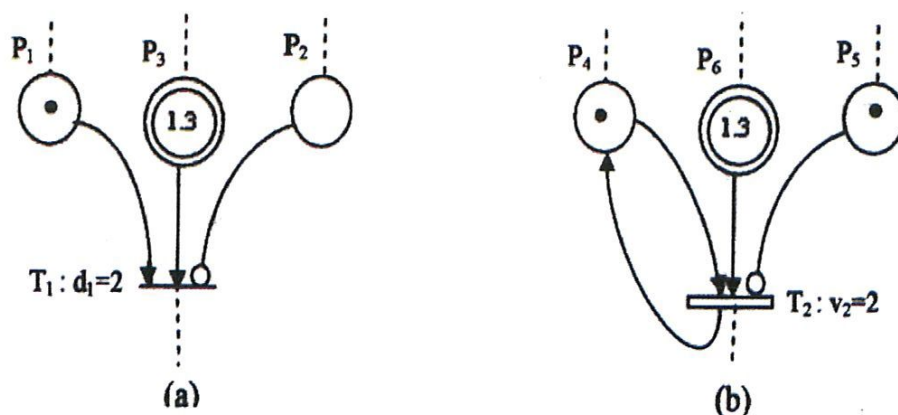


Figure 2.29. Test à zéro d'une place discrète, (a) T_1 est validée, (b) La transition n'est pas validée.

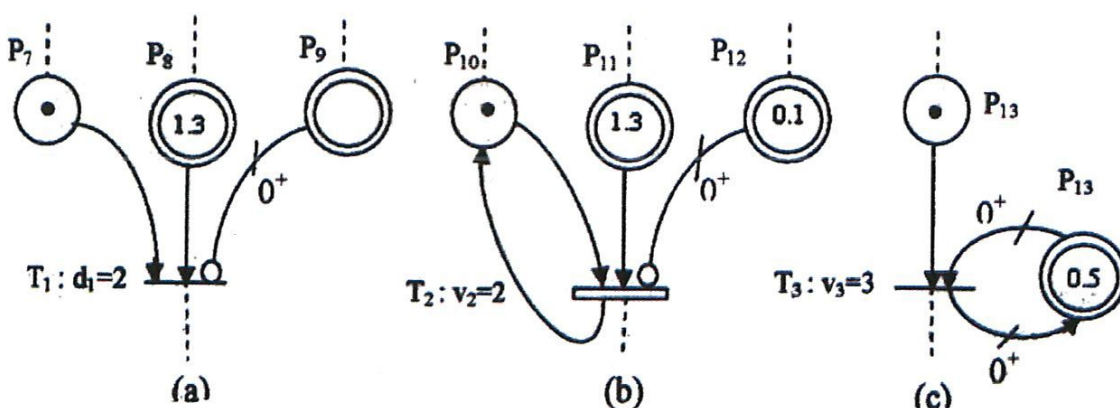


Figure 3.30. Test à zéro d'une place continue, (a) T_3 est validée, (b) La transition T_4 est non validée, (c) La transition est validée seulement si P_{13} est non vide.

2.8. Conclusion

A cours de ce chapitre nous avons abordé un certain nombre de notions concernant les réseaux de Petri, les réseaux de Petri temporisés, les réseaux de Petri continus, les réseaux de Petri hybrides. Ces notions, qui seront à nouveau mentionnées au cours de ce travail, ensuite on a discuté du passage de RdP du discret au continu, . Nous avons consacré la dernière section du chapitre, pour les RdP hybrides.

Le réseau de Petri est un outil graphique et mathématique pour modéliser et analyser les systèmes. Il nous aide à comprendre facilement le système modélisé, et en plus il nous permet de simuler les activités dynamiques et concurrentes. Avec le rôle d'outil mathématique, il nous permet d'analyser le système modélisé grâce aux modèles de graphes, aux équations algébriques...etc.

Dans le chapitre suivant nous allons utilisé l'application *SIRPHICO* pour simuler les systèmes dynamiques hybrides par les Réseaux de petri Hybrides.

Chapitre 3

Modélisation et simulation des SDH par RdPH

3.1. Introduction

Ce chapitre est consacré à la simulation du comportement dynamique des systèmes hybrides par le logiciel de simulation SIRPHYCO.

En effet la simulation est un moyen efficace et économique, couramment utilisée pour faire des études préliminaires et/ou comparatives. La simulation, objet de ce chapitre, est l'utilisation d'un outil, généralement informatique, qui s'applique à un système donné (hybride).

Actuellement, plusieurs outils de simulation, parmi lesquels MATLAB/SIMULINK, sont utilisés dans l'industrie et dans les milieux universitaires et le logiciel SIRPHYCO est destiné à la recherche, à la simulation des systèmes concrets modélisés par RdP continus et hybrides, éventuellement à l'enseignement.

3.2. Présentation de l'application SIRPHYCO

L'application *SIRPHYCO* (*Simulateur de RdP HYbrides et COntinus*) est un logiciel qui permet la simulation des RdP discrets, continus et hybrides. Cet outil permet aussi la réalisation de comparaisons entre les différents modèles de simulation, plus particulièrement, la comparaison des modèles continus par rapport aux modèles de RdP discrets. Cette application permet d'évaluer les performances d'un système modélisé par un RdP, en calculant des quantités moyennes relatives aux entités du RdP, tels que les marquages moyens des places et les fréquences moyennes de franchissement des transitions.

Au début, la première version de l'application *SIRPHYCO*, avait été développée et programmée en langage *FORTRAN* sur une station de travail *APPOLO*. Puis, sur la station de travail *Sun*. Pour cette version, l'interface homme-machine était presque entièrement graphique. Ensuite une autre version a été développée sur PC en utilisant la langage de programmation C++ Cela a permis la mise en valeur des avantages de la programmation orientée objets (*POO*), en enfin la dernière version du logiciel *SIRPHYCO* [2005] utilisant la programmation en MATLAB version7.

3.3. Architecture de l'application SIRPHYCO

Le logiciel est constitué de quatre parties distinctes par leur rôle mais néanmoins complémentaires.

- ❖ La première partie, concerne la gestion de l'interface utilisateur sous forme graphique.
- ❖ La seconde partie gestion des objets RdP constitue l'implémentation informatique des modèles RdP et leurs simulateurs associés.
- ❖ La troisième partie gère les deux types de documents produits par l'application SIRPHYCO. Le premier type de document contient les RdP, le deuxième type contient les résultats de simulation.
- ❖ La quatrième partie gère l'affichage des documents. Elle permet d'afficher les RdP sous forme graphique et les résultats des simulations sous forme de courbes ou sous forme numérique.

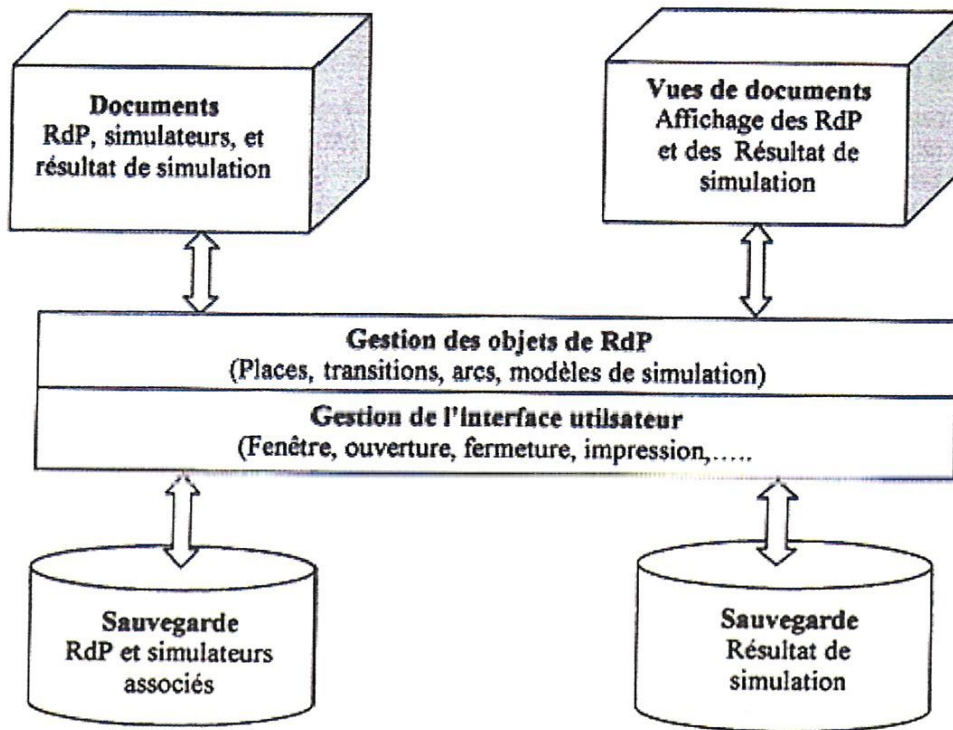


Figure. 3.1. Schéma de principe de la structure de SIRPHYCO

3.4. Fonctionnement de l'application SIRPHYCO.

L'application *SIRPHYCO* comporte un menu principal et des fenêtres, l'une sert à la saisie graphique des RdP et à la simulation, l'autre permet de représenter les résultats de simulation sous forme courbes. Ces fenêtres permettent de visualiser les documents gérés par l'application *SIRPHYCO*, et la sauvegarde sous forme de fichiers.

❖ Saisie des RdP :

Dans le menu principal on trouve des fenêtres pour la saisie des RdP, une fenêtre pour les places (continues ou discrètes) est une fenêtre pour les transitions (continues ou discrètes) et une fenêtre pour les arcs (ordinaire ou généralisé). Après la saisie d'un RdP, il est possible de vérifier sa validation. Dans tous les cas, le RdP est vérifié lors de l'affectation d'un mode de simulation.

❖ Simulation :

La simulation du RdP s'effectue selon le modèle de RdP introduit soit, pas à pas soit en seule itération. L'arrêt de la simulation est réalisé par une durée maximale de la simulation. Dès que le RdP contient des conflits structurels, il est nécessaire de définir une politique de résolution des conflits. Elle peut être définie soit à l'aide de priorités par défaut, soit par une résolution des conflits (résolution par partage).

3.5. Exemples de simulation

Nous représentons dans ce paragraphe la simulation des trois exemples suivants : un système de recyclage d'eau, une machine produisant des pièces par lots et un circuit électrique (Le hacheur abaisseur) et nous donnons en suites quelques résultats de simulation par le logiciel SIRPHYCO des RdPH.

3.5.1. Exemple 1 : Système de recyclage d'eau

Dans cet exemple représenté sur la *figure. 3.2.a.* un régulateur de volume d'eau entre deux réservoirs d'un système de recyclage d'eau. Initialement le réservoir A contient 180 m^3 d'eau froide, l'écoulement de l'eau du réservoir A vers le réservoir B s'effectue par la pompe A, qui fonctionne à une vitesse $V_1=3$ et qui est disponible durant 90 unités de temps. Alors que le transfert d'eau chaude du réservoir B vers le réservoir A s'effectue à travers la pompe B qui fonctionne à une vitesse $V_2=2$ et qui est disponible pendant 60 unités de temps. L'évolution du marquage des places $m(P_1)$, $m(P_3)$, $m(P_4)$ du RdPH du système pour un marquage initial

$$M_0 = [M_0^D, M_0^C] \text{ avec } M_0^D = [1, 0] \text{ et } M_0^C = [180, 0].$$

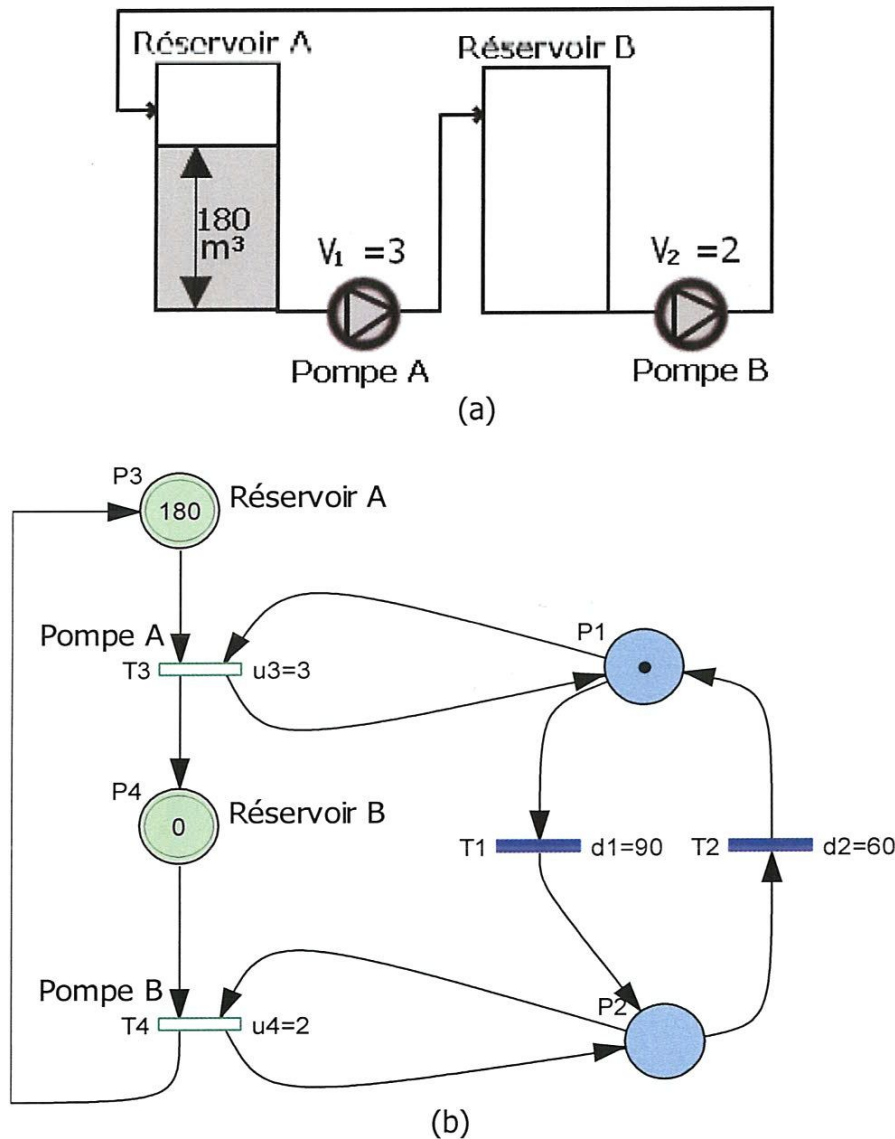
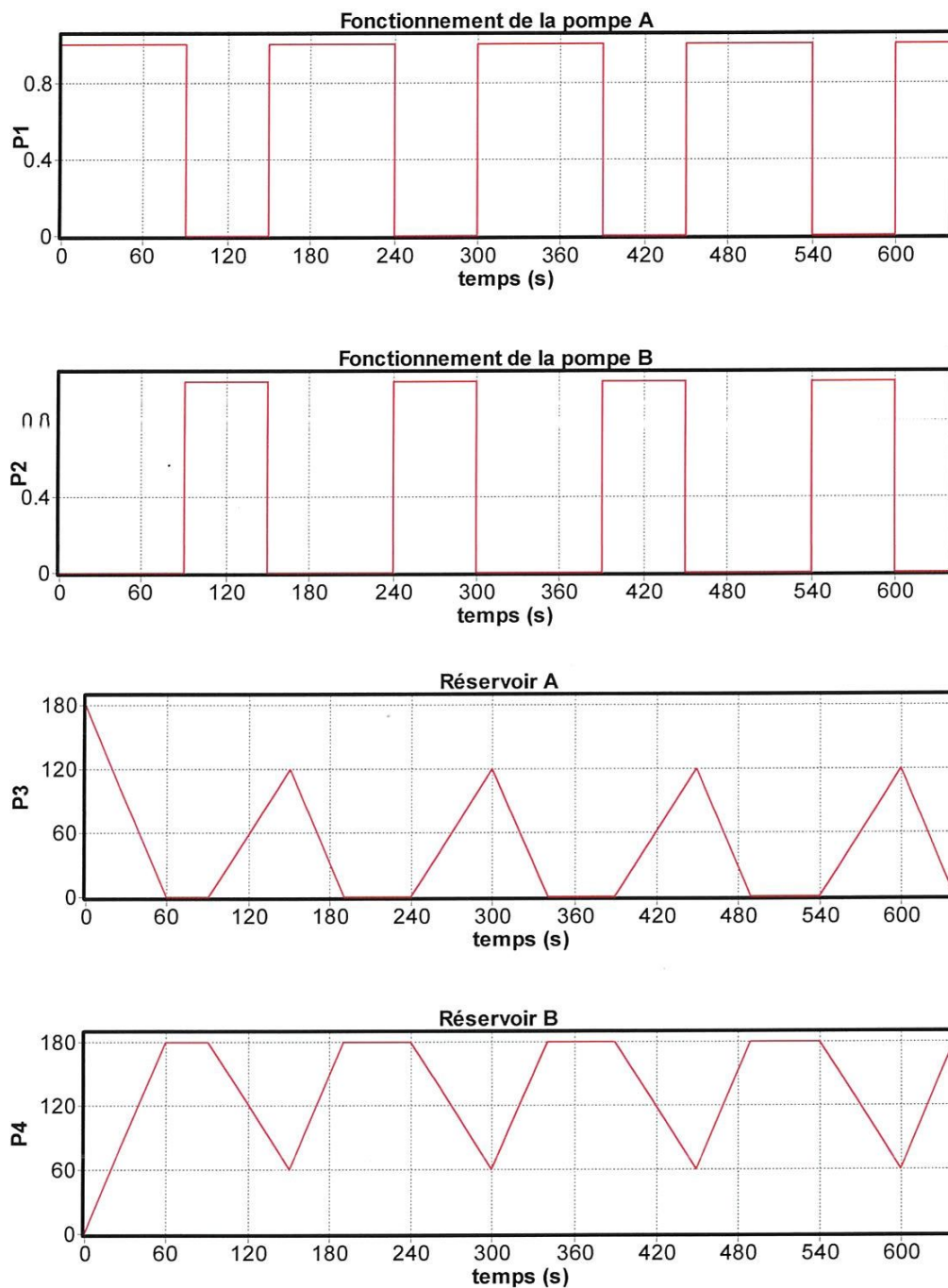


Figure. 3.2. (a) Système de recyclage d'eau, (b) Modèle du système.

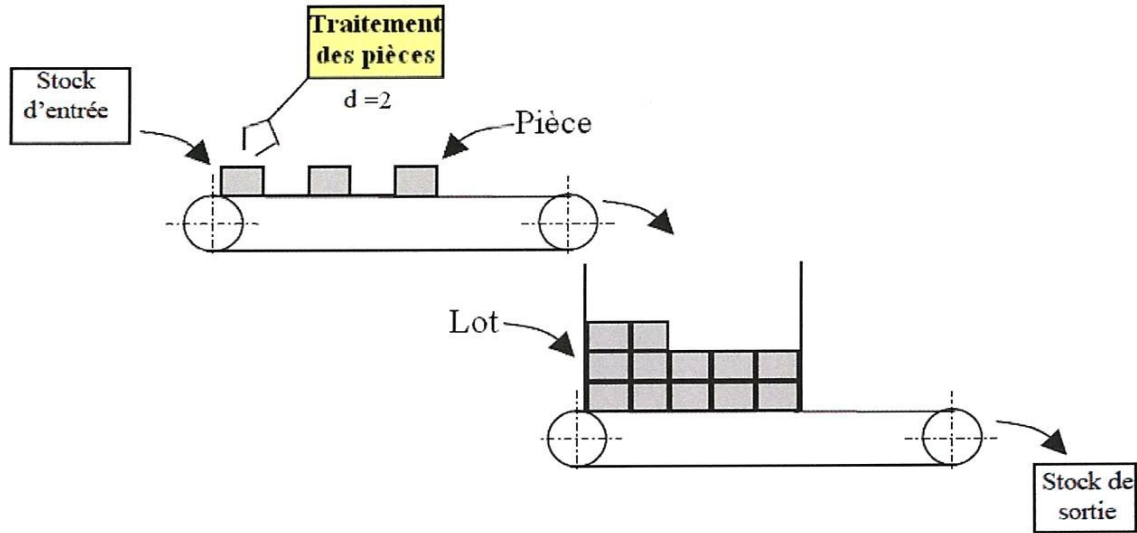


(c)

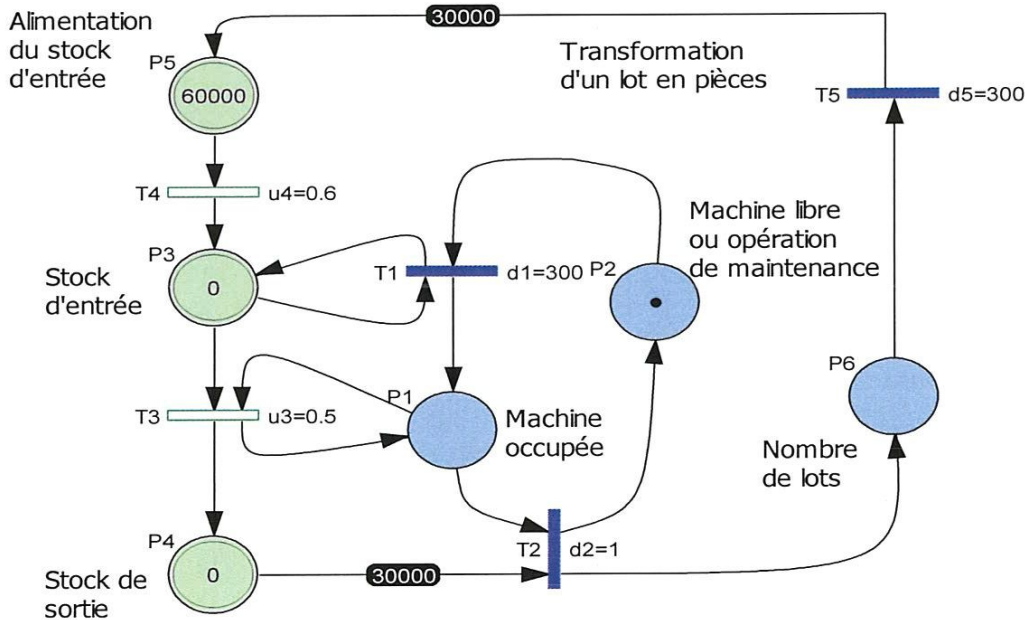
Figure. 3.2.c. Marquage des places P_1 , P_2 , P_3 et P_4 .

3.5.2. Exemple 2 : Machine produisant des pièces par lots.

La figure. 3.3.a. représente le travail d'une machine qui effectue une production par lots. Chaque lot est de 30000 pièces, et le temps de traitement de chaque pièce est 2 unités de temps. Cette machine a un stock d'entrée dont sont extraites les pièces à produire. La machine reste opérationnelle et affectée au lot jusqu'à ce que le lot soit terminé. On effectue alors une opération de maintenance qui dure 300 unités de temps, avant de pouvoir l'affecter à un autre lot. La modélisation par RdPH du système est représentée sur la figure. 3.3.b. et l'évolution du marquage dans les places P_1 , P_2 , P_3 et P_4 représentant sur la figure. 3.3.c.

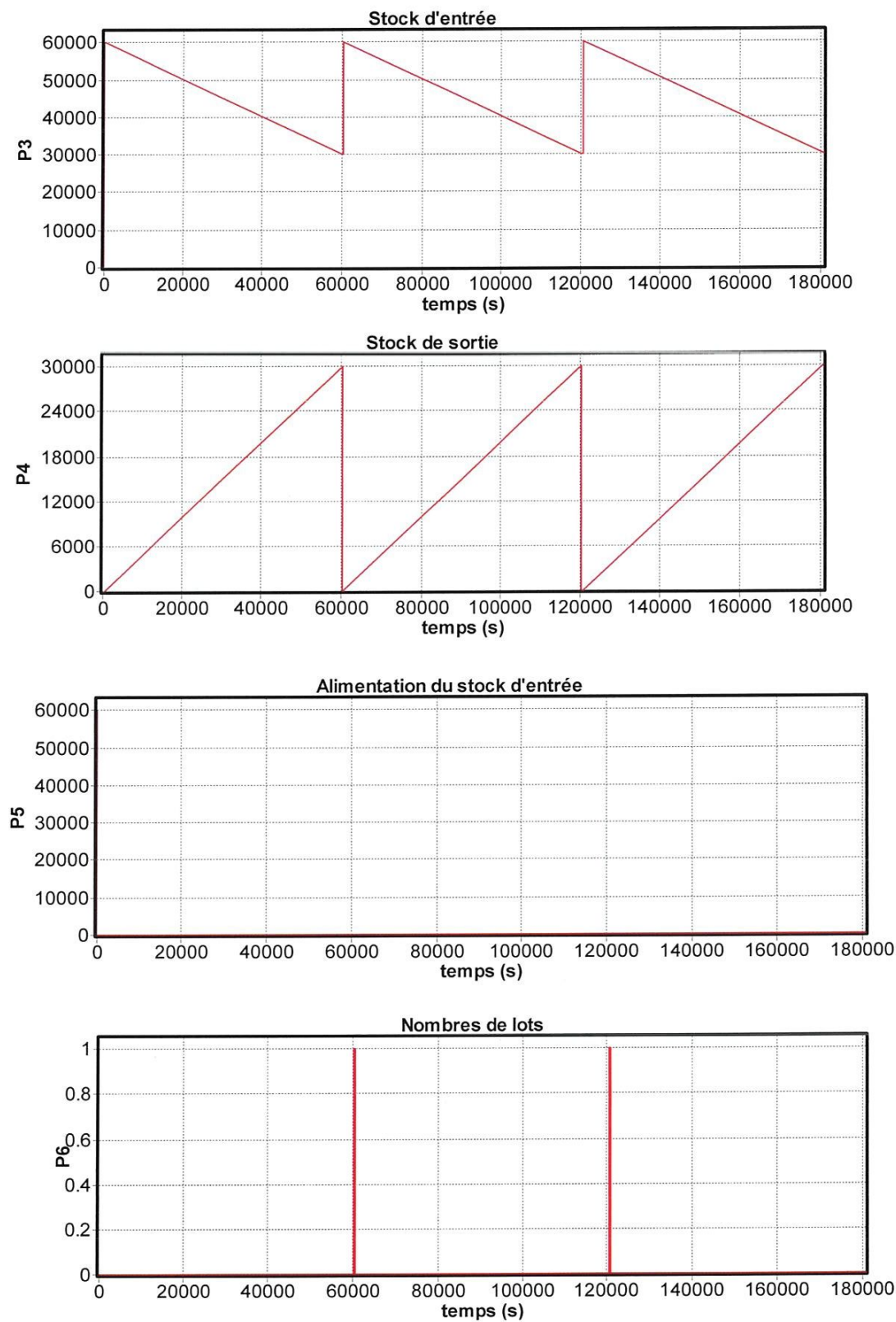


(a)



(b)

Figure. 3.3. (a) Système de production par lots, (b) Modèle du système.



(b)

Figure. 3.3.c. Marquage des places P_3 , P_4 , P_5 et P_6 .

3.5.3. Exemples 3 : Circuit électrique (Le hacheur abaisseur)

Le hacheur avec la charge de RL est montré sur le schéma de *figure. 3.4.a*. Le fonctionnement du hacheur peut être divisé en deux modes. Pendant le mode 1, le hacheur est fermé et le courant passe dans la charge. Pendant le mode 2, le hacheur est ouvert et le courant de charge continue à passer à travers la diode roue libre D.

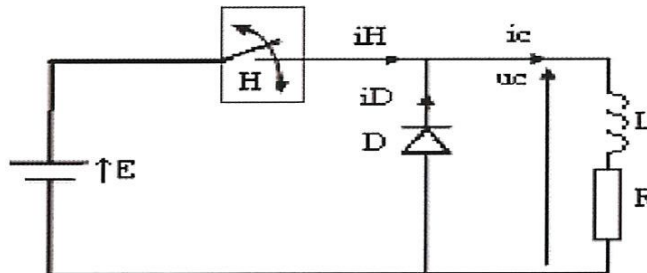


Figure. 3.4.a. Le hacheur abaisseur.

Le courant de charge pour le mode 1 peut être obtenu de :

$$E = R i_c + L \frac{di_c}{dt} \quad (3.1)$$

La solution de l'équation (3.1) avec courant initial $i_c(t=0) = i_n$ et l'approximation $e^{-x} = 1 - x$ Donne le courant de charge comme suit :

$$\begin{cases} i_c = i_H(t) = \frac{E}{R} + \left(i_n - \frac{E}{R}\right) \left(1 - \frac{t}{\tau}\right) \\ i_D = 0 \end{cases} \quad (3.2)$$

Ce mode est valable pour $0 < t < T$, et à l'extrémité de ce mode, le courant de charge devient :

$$i_c(t = \alpha T) = I_m$$

Le courant de charge pour le mode 2 peut être obtenu de l'équation suivante :

$$0 = R i_c + L \frac{di_c}{dt} \quad (3.3)$$

La solution de l'équation (3.3) avec $e^{-x} = 1 - x$ donne le courant de charge comme suit :

$$\begin{cases} i_c = i_D(t) = I_m \left(1 - \frac{(t - \alpha T)}{\tau}\right) \\ i_H = 0 \end{cases} \quad (3.4)$$

Ce mode est valable pour $T < t < T$. A la fin de ce mode, le courant de charge devient $i_c(t = T) = I_n$.

Le hacheur est mis en marche encore dans le prochain cycle après le temps $T = 1/f$. Avec, I_m, I_n et $\Delta I = I_m - I_n$ sont respectivement le courant, maximum, minimum et ondulation du courant, α et le rapport cyclique du hacheur, T est la période du hachage, f et la fréquence du hacheur et $\tau = L/R$ la constante du temps de la charge. Avec $E = 220V$, $R = 5 \Omega$, $L = 7.5mH$, $\alpha = 0.5$, nous obtenons $I_m = 25A$, $I_n = 18A$.

Le modèle hybride du hacheur est représenté sur la *figure. 3.4.b*. Les deux places P_1 et P_2 sont associées aux deux modes de l'opération du hacheur. Une marque dans P_2 correspond à la conduction de H et du courant qu'elle conduit est indiqué par marquage de la place P_3 . La forme d'onde du courant est représentée par marquage de P_3 montré sur la *figure. 3.4.c*.

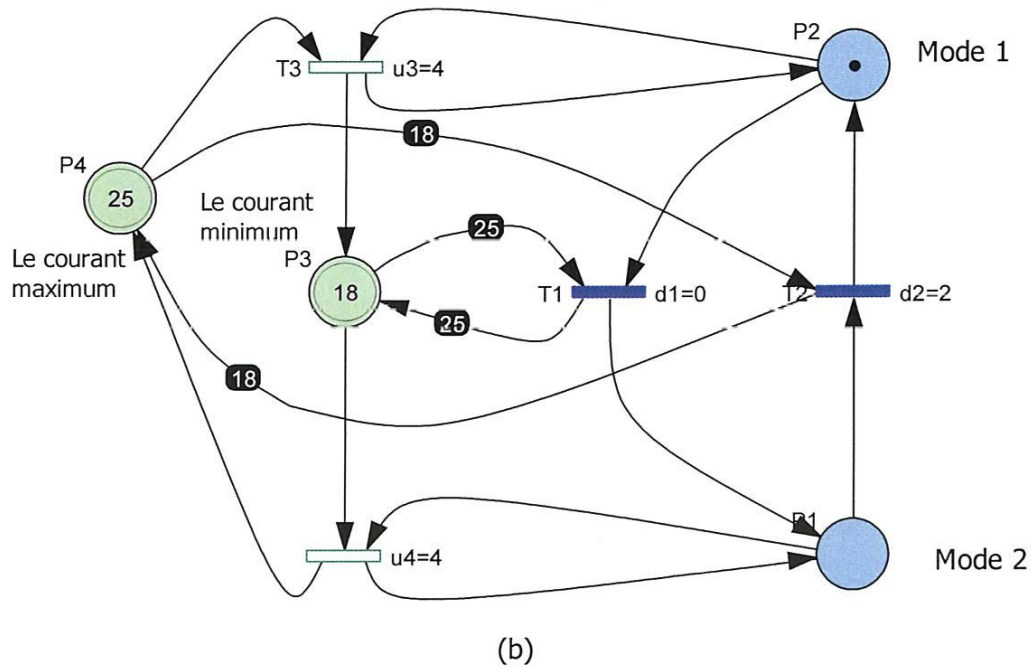
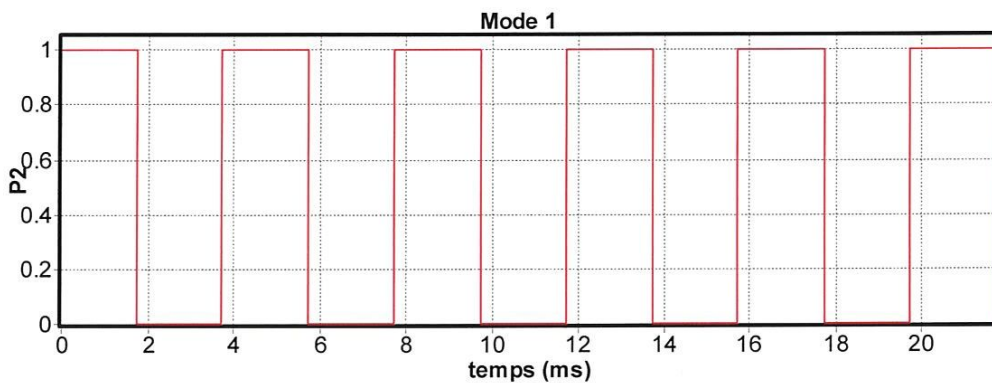
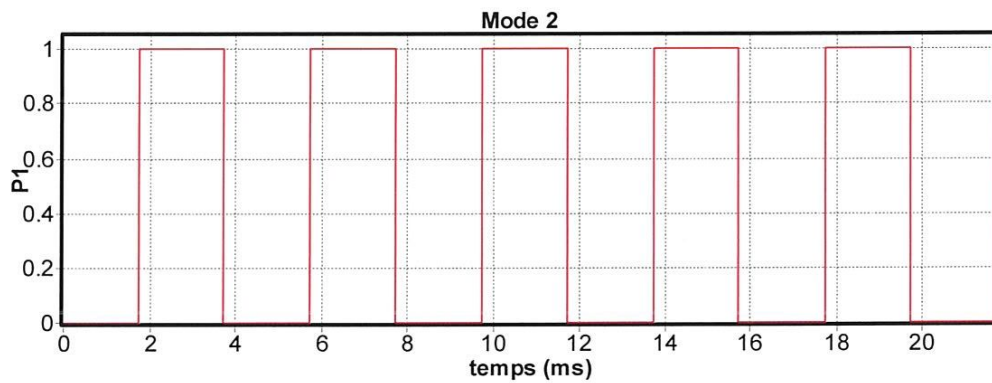


Figure. 3.3. (b) Modèle RdPH du comportement du hacheur.



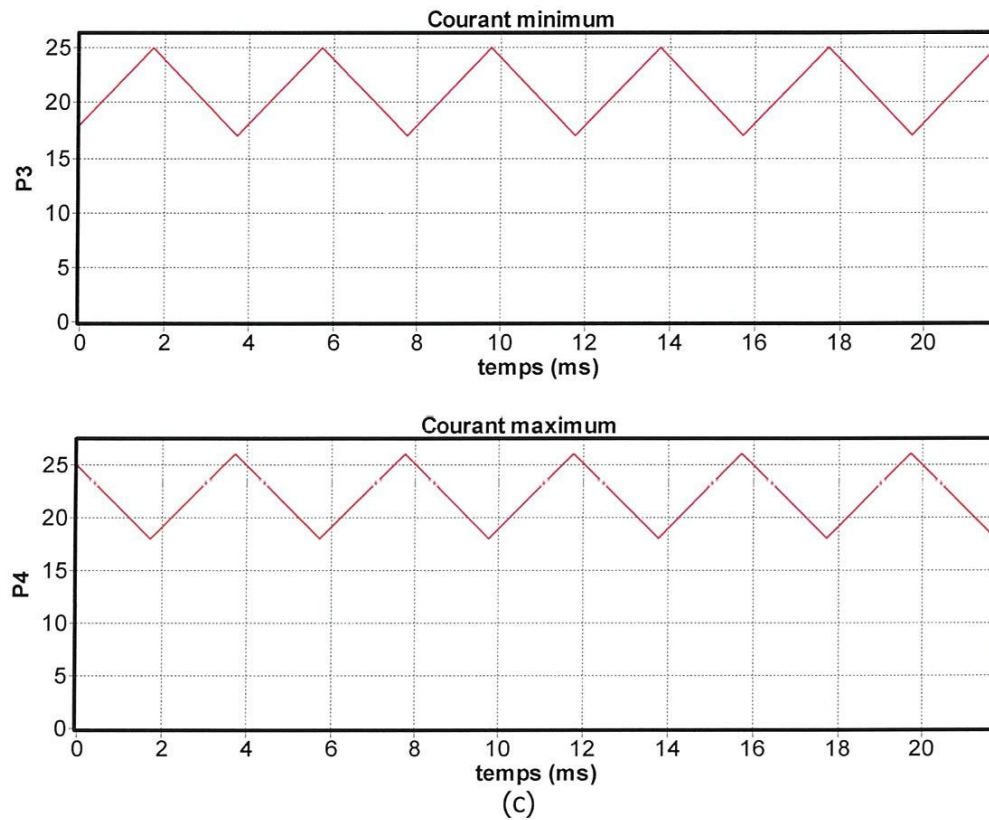


Figure. 3.3. (c) Marquage des places P_1 , P_2 , P_3 et P_4 .

3.6. Conclusion

L'intérêt essentiel des Réseaux de Petri hybrides réside dans leur capacité d'analyse du modèle graphique. Cette analyse, dite validation, permet d'extraire du modèle les propriétés que doit posséder le système physique représenté. On peut ainsi montrer que le modèle obtenu est borné, sans blocage ou réinitialisable.

Conclusion Générale

Les systèmes dynamiques hybrides sont des systèmes définis structurellement par la coopération de deux systèmes, l'un de type continu et le second est de type événementiel. L'objectif de notre travail est de modéliser et simuler les systèmes dynamiques hybrides et vérifier les modèles obtenus par la simulation. Dans ce contexte nous avons proposé un outil de modélisation puissant qui à une présentation graphique et formelle, il décrit les aspects continu et discret du système donc une représentation homogène du système. L'outil de modélisation est le Réseau de Petri Hybride, une extension des réseaux de Petri classiques, il intègre deux types de RdP : le Réseau de Petri continu modélisant la partie continue du SDH et le RdP Temporisé modélisant la partie discrète du SDH. L'analyse du modèle permet la détermination des états comportementaux du système.

Enfin, les modèles Réseaux de Petri Hybrides peuvent être utilisés en commande. C'est ainsi que l'on pourra associer aux jetons de chaque place, d'une part des actions de commande destination du procédé, d'autre part, des événements comptes rendus de ces actions. Il est ainsi possible de les utiliser comme aide à la décision dans l'exploitation en temps réel. Le formalisme de réseaux de Petri hybrides est très bien adapté pour l'analyse de systèmes d'une taille assez petite. Pour des systèmes complexes, la topologie du réseau, bien que sa dynamique, posent réels problèmes dues soit à la structure du système, soit à l'explosion d'états atteignables. Une approche qui soulève parfois cet inconvénient a comme sujet les Réseaux de Petri Objets, constitués par des réseaux primaires, avec une structure et une dynamique interne, qui permet leur liaison directe ou bien par des éléments auxiliaires, externes, qui assurent le bon fonctionnement du modèle.

Bibliographie

- [1] A. Chombart, "*Commande supervisée des systèmes hybrides*", Thèse de docteur de l'INPG, Grenoble, 1997.
- [2] C. Valentin Roubinet, "*Hybrid Systems modelling : Mixed Petri Nets*", 3rd IMACS/IEEE Conference CSCC, Athènes (Grèce), 4-8 Juillet, 1999.
- [3] J. Le Bail, "*Sur les Réseaux continus et hybrides*", Thèse de docteur de l'INPG, Grenoble, 1992.
- [4] J. ZAYTOON et coll, "*Systèmes dynamiques hybrides*", Hermes Sciences Publications, 2001.
- [5] M.A. Drighiciu, "*L'analyse de systèmes électromécaniques – une approche par les Réseaux de Petri*", Analele Université din Craiova, Seria Inginerie Electric, Anul 24, Nr.24, Editura Universitaria, 2000, pag.103 – 115.
- [6] M. Demongodin et F. Prunet, "*Extension of hybride Petri nets for accumulation systems*" IMACS, Decembre, 1992.
- [7] R. David et H. Alla, "*Du Grafcet aux Réseaux de Petri*", Edition Hermès, 1992, 2ième édition, Paris.
- [8] <http://www.daimi.au.dk/PetriNets/>
- [9] http://www.tn.refer.org/hebergement/cours/sys_disc/notions_de_base_RdP.html