



## AVERTISSEMENT

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

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

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

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

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

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

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

Université Henri Poincaré de NANCY I  
Ecole Doctorale I.A.E.+M.  
DFD Automatique et Production Automatisée  
UFR S.T.M.I.A.



CNRS URA 821



## THESE

Présentée pour l'obtention du

## DOCTORAT

DE L'UNIVERSITE HENRI POINCARÉ DE NANCY I

Spécialité Automatique

par

**Abdellatif BOURJIJ**

---

**CONTRIBUTION A LA SURETE DE FONCTIONNEMENT DES PROCESSUS  
INDUSTRIELS PAR LES RESEAUX DE PETRI**

---

Soutenue publiquement le 5 décembre 1994 devant la commission d'examen:

Président : G. KRZAKALA

Rapporteurs : G. VILLERMAIN-LECOLIER  
D. THOMASSET

Examineurs : M. DAROUACH  
J. BREMONT  
M. ZASADZINSKI

## AVANT-PROPOS

*Cette thèse a été préparée au Centre de Recherche en Automatique de Nancy, au sein de l'Equipe d'Automatique et de Recherche Appliquée de Longwy (C.R.A.N.-E.A.R.A.L. C.N.R.S. URA 821) dirigée par Messieurs les Professeurs Gérard KRZAKALA et Mohamed DAROUACH.*

*Je remercie les membres du Jury qui me font l'honneur de participer à l'examen de ce travail:*

*Monsieur le Professeur Gérard KRZAKALA, de l'Université de Nancy I, Directeur de l'Institut Universitaire de Technologie B de Nancy I, pour avoir accepté de m'accueillir au sein de son équipe, pour son soutien permanent et pour l'honneur qu'il me fait en présidant ce jury.*

*Monsieur le Professeur Mohammed DAROUACH, de l'Université de Nancy I, Directeur de l'Equipe d'Automatique et de Recherche Appliquée de Longwy, pour la qualité de son encadrement et de ses conseils.*

*Monsieur le Professeur Gérard VILLERMAIN-LECOLIER, de l'Université de Reims, Directeur du Laboratoire d'Applications de la Micro-électronique et Monsieur le Professeur Daniel THOMASSET de l'I.N.S.A. de Lyon, pour avoir accepté d'examiner ce travail et pour leurs précieuses remarques.*

*Monsieur le Professeur Jacques BREMONT, de l'Université de Nancy I, pour l'honneur qu'il me fait en participant à ce jury et pour l'intérêt qu'il porte à ce travail.*

*Monsieur Michel ZASADZINSKI, Chargé de Recherche au C.N.R.S., pour la qualité et la chaleur de ses conseils et pour la bienveillance qu'il me témoigne en participant à ce jury.*

*Je remercie également les personnes qui ont contribué au bon déroulement de ce travail:*

*Monsieur Michel MUSSET, paix à son âme, Maître de Conférences à l'I.U.T. de Longwy et initiateur de ce travail, pour sa sympathie, son soutien, sa disponibilité, et sa générosité.*

*Monsieur Roland MUSSET, Enseignant à l'I.U.T. de Longwy, Monsieur Mohamed BOUTAYEB et Monsieur Hugues RAFARALAHY, chercheurs à l'E.A.R.A.L., pour leur disponibilité et leurs conseils avisés.*

*Enfin, mes remerciements vont à tous mes collègues et au personnel de l'E.A.R.A.L., ainsi qu'à l'ensemble du personnel de l'I.U.T. de Longwy avec lesquels le travail a toujours été un plaisir.*

---

## "CONTRIBUTION À LA SURETE DE FONCTIONNEMENT DES PROCESSUS INDUSTRIELS PAR LES RESEAUX DE PETRI"

**Résumé:** Dans ce mémoire, nous nous sommes intéressés à la sûreté de fonctionnement des processus industriels. L'approche préconisée consiste à modéliser le système par les Réseaux de Petri. L'algèbre linéaire, notamment le calcul des invariants, est utilisée pour vérifier qu'un modèle a les propriétés requises (vivant, borné ..) et répond au cahier des charges. Aussi, nous proposons un algorithme efficient de calcul des invariants, particulièrement adapté aux systèmes de grande dimension. Puis deux méthodes de diagnostic basées, pour la première sur un modèle de référence (RdP), pour la seconde sur l'utilisation des invariants, sont présentées. Enfin, ces développements théoriques sont appliqués à deux processus industriels: un réseau de générateurs de vapeur simulé et un quantomètre à émission optique.

**Mots clefs:** Réseaux de Petri, diagnostic, P-semi-flots, T-semi-flots, sûreté de fonctionnement.

---

## "CONTRIBUTION TO THE INDUSTRIAL PROCESSES SAFETY WORKING BY THE USE OF PETRI NETS"

**Abstract:** The aim of this work is the safety working of industrial processes. The proposed approach consists in modelling the process through Petri Nets. The model validation, i. e. verifying if the model has the required specifications, is achieved by using linear algebra tools, especially by invariants computation. In fact, we suggest an efficient algorithm, which is particularly adapted to large scale systems, for the invariants computations. Then two diagnosis methods are presented: the first one is based on reference model and the second on the use of invariants. Finally, these theoretical results are applied to industrial processes: a steam boilers net simulator and a mass spectrometer.

**Keywords:** Petri Nets, diagnostic, S-invariants, T-invariants, safety working.

## **SOMMAIRE**

# SOMMAIRE

---

## INTRODUCTION GENERALE

CHAPITRE I	L'OUTIL RESEAUX DE PETRI
------------	--------------------------

	Pages N°
1. Intérêt	7
2. Notations et définitions de base	8
3. Représentation matricielle	11
4. Propriétés essentielles	14
5. Graphe des marquages	
5.1. Rappels sur la théorie des graphes	19
5.2. Application aux RdP	20
6. Extensions	
6.1. RdP Temporisés	
6.1.1. Présentation	22
6.1.2. Définitions	23
6.1.3. Fonctionnement en régime stationnaire	24
6.1.4. Fonctionnement en vitesse propre	26
6.2. RdP Continus	
6.2.1. Présentation	26
6.2.2. Définitions	28
6.3. RdP Hybrides	
6.3.1. RdPH autonomes	30
6.3.2. RdPH temporisés	32
7. Modélisation	33
8. Conclusion	37
9. Bibliographie	38

	Pages N°
1. Introduction	43
2. Position du problème	
2.1. P-semi-flots	44
2.2. T-semi-flots	45
2.3. Remarques	46
3. Etat de l'art	
3.1. Première approche	47
3.2. Deuxième approche	49
4. Méthode proposée	52
5. Discussion	60
6. Cas des RdP de grande dimension	62
6.1. Décomposition en blocs diagonaux	63
6.2. Résolution	68
6.3. Exemple numérique	69
6.4. Discussion	71
7. Vérification des propriétés des systèmes	73
8. Conclusion	76
9. Bibliographie	77

	Pages N°
1. Introduction	81
2. Les systèmes de surveillance	82
3. La détection	
3.1. Détection par modèle de référence	
3.1.1. Concept	85
3.1.2. Application	86
3.1.3. Localisation	88
3.1.4. Conclusion	92
3.2. Détection par les invariants	
3.2.1. Concept	92

3.2.2. Application	93
3.2.3. Conclusion	94
3.3. Remarques	95
4. Discussion	97
5. Conclusion	98
6. Bibliographie	98

CHAPITRE IV	APPLICATIONS INDUSTRIELLES
-------------	----------------------------

	Pages N°
Présentation	105
<b>Ière Partie Le simulateur</b>	
1. Introduction	106
2. Architecture du simulateur	
2.1. Modèle mathématique d'une bouilloire	106
2.2. Le réseau de générateurs de vapeur	110
2.3. Le système de commande	111
2.4. La supervision	112
3. Sûreté de fonctionnement	
3.1. Modélisation	113
3.2. Surveillance	117
<b>IIème Partie Le quantomètre</b>	
1. Introduction	121
2. Principe de fonctionnement	121
3. Structure de l'instrument	
3.1. La source et le statif	122
3.2. Le spectromètre	123
3.3. L'unité de commande	124
4. La commande et le calcul des concentrations	124
5. Sûreté de fonctionnement	
5.1. Modélisation	129
5.2. Surveillance	131



Conclusion	135
Bibliographie	136

## **CONCLUSION GENERALE**

## **LISTE DES FIGURES**

## **INTRODUCTION GENERALE**

Dans sa recherche effrénée de confort et de sécurité, l'homme a apporté des réponses technologiques toujours plus raffinées et complexes. A des contraintes de délai, de cadence et de qualité, les réponses ont été des machines de production complexes dont les temps de réponse sont souvent de l'ordre de la milliseconde, véhiculant d'énormes flots de données. Ces procédés de fabrication étant devenus impossibles à gérer par l'homme seul, il a fallu automatiser et informatiser. Aussi, l'informatique, née sans doute avec le boulier, a atteint un degré de sophistication tel que certaines unités de fabrication ne requièrent plus que la vigilance de quelques opérateurs. De plus, l'informatisation et l'automatisation ne sont plus réservés aux domaines sensibles comme l'aérospatiale, le nucléaire ou les télécommunications, mais touchent tous les secteurs de l'industrie ou de l'économie.

Cependant, plus un processus est complexe plus le risque d'apparition d'erreurs ou de défaillances est grand et plus sa maintenance est ardue et onéreuse. L'attention se porte donc de plus en plus vers la conception de systèmes de commande, sûrs, c'est-à-dire des systèmes capables de maintenir un rythme de production stable. Aussi, doit-on être en mesure de détecter les défauts, de les localiser et de trouver les solutions adéquates. Les systèmes de surveillance aident à cette tâche en espionnant l'ensemble processus et système de commande. Ils disposent d'informations en temps réel sur l'état du système et sont capables d'agir pour maîtriser les anomalies (fautes, erreurs, défaillances).

Le travail proposé dans ce mémoire vise à concourir dans ce sens et à apporter quelques éléments de contribution à la sûreté de fonctionnement des processus industriels. Que le processus, outil de production, soit une centrale nucléaire ou un simple appareil d'analyses, l'objectif d'une automatisation réussie reste toujours le même, c'est-à-dire garantir un fonctionnement le plus sûr possible, les investissements étant bien évidemment différents. Les Réseaux de Pétri (RdP) ont été choisis comme outil de modélisation pour la clarté de leur représentation graphique et la puissance des outils mathématiques associés.

Afin d'éviter de nombreux rappels sur les Réseaux de Petri, tout au long de ce mémoire, nous avons choisi de les regrouper dans un premier chapitre introductif, que le lecteur avisé pourra escamoter.

Le deuxième chapitre est consacré à une propriété saillante des RdP, à savoir les "invariants". Les méthodes de calcul des invariants sont introduites d'abord par un

algorithme classique puis par un second basé sur le calcul du noyau de la matrice d'incidence par échelonnement de cette dernière. Un algorithme efficient de recherche des "semi-flots" par réduction du système initial est proposé. L'efficacité de cet algorithme est montrée grâce à une étude comparative des trois méthodes. Le calcul des invariants est étendu au cas particulier des RdP de grande dimension. Une méthode de décomposition des systèmes de grande dimension en sous-systèmes plus simples à résoudre est présentée (par utilisation de la "Technologie de Groupes"). Enfin, l'utilité et l'intérêt que peut présenter la recherche de telles propriétés sont montrés sur un exemple.

Le troisième chapitre aborde un sujet d'actualité: les outils de surveillance des systèmes de production automatisée. Suite à une présentation générale de ces systèmes, une approche de la surveillance est exposée et deux algorithmes de détection de défauts sont proposés. Le premier par l'utilisation d'un modèle de référence, le second, par l'exploitation des techniques précédentes de calcul des invariants. Les résultats des applications exposées lors du chapitre suivant montrent la portée de ces algorithmes.

Epilogue de ce travail, le chapitre IV présente deux applications développées au sein de l'équipe de recherche de Longwy du Centre de Recherche en Automatique de Nancy (C.R.A.N.) et faisant l'objet de contrats avec des entreprises ou des laboratoires étrangers. L'une des applications est un simulateur composé d'un réseau de générateurs de vapeur, d'un système de commande et d'un logiciel de supervision. Cet outil implanté sur un réseau local hétérogène (une station SUN, un DPX2000 et des P.C.), a été développé à des fins de conception, de test et de mise au point de programmes de commande. L'autre application est un quantomètre, c'est-à-dire un appareil d'analyse de la composition atomique de matériaux. Le principe du quantomètre est d'analyser les spectres d'atomes émis par l'échantillon pour déterminer sa concentration en atomes particuliers. Pour chacune des deux applications modélisées par les RdP, les calculs des invariants par les méthodes préconisées sont donnés et discutés. Nous terminons, pour chaque cas, par une discussion sur la sûreté de fonctionnement et sur les procédures choisies de détection des défauts.

## CHAPITRE I

### L'OUTIL RESEAUX de PETRI

## 1. Intérêt

Les réseaux de Petri (abrégé dans la suite du texte en "RdP") sont nés en Europe en 1962 des travaux de l'Allemand Carl Adam PETRI [PET62]. Restés en désuétude pendant une décennie, ils nous sont revenus des Etats Unis, notamment du M.I.T., où des chercheurs s'intéressent à la modélisation et l'étude des systèmes à événements discrets [MUR77] [SIF77]. Ils prennent aujourd'hui un essor important en Europe, particulièrement dans les laboratoires de recherche français qui organisèrent le premier congrès européen sur les RdP, en 1980 à Strasbourg. Cet outil de modélisation a le double avantage d'être graphique et mathématique, atouts qui enthousiasment aussi bien les théoriciens que les praticiens:

- l'aspect graphique clarifie et synthétise la représentation du système modélisé (visualisation), le déplacement de marques dans les places simulant la dynamique de celui-ci,
- l'aspect mathématique permet de poser les équations d'état du système et d'analyser ce dernier par l'algèbre linéaire ou la théorie des graphes.

Les RdP peuvent donc représenter aussi bien des processus informatiques (tâches) que des processus industriels (chaîne de commande de machines) mais aussi tous systèmes caractérisés par leurs aspects parallèles, asynchrones, distribués, concurrents, non déterministes et/ou stochastiques [BRA83] [DAV89] [MUR89] [BES88] [CHO87] [BEN92] [REI82] [RIC83] [THE78].

Dans de nombreuses phases d'une automatisation on peut utiliser les RdP. En effet l'automaticien est souvent confronté à ces deux types de situations:

- concevoir un système (ensemble de procédures, processus ou éléments physiques constituant une installation industrielle) [NOY90] [VAL80],
- étudier un système existant (en vue de l'améliorer, le dépanner etc.) [LAM85] [LEC87] [MUR89].

Dans le premier cas, les RdP facilitent la conception de l'architecture du système, en fournissant un outil de représentation graphique. Ils permettent de poser les problèmes susceptibles d'intervenir:

- blocages,
- incohérences des données,
- famines,

contraintes de temps,  
conflits d'accès aux ressources,  
etc.

et de leur apporter des solutions optimales [BEN92] [BRA93] [CHR84] [DAV94] [DIF93].

Dans le deuxième cas de figure, ils permettent de réaliser un modèle (plus ou moins fin) qui servira à l'analyse du système, la recherche de ses propriétés et l'évaluation de ses performances. Les RdP peuvent aussi générer des modèles de référence pour le diagnostic (voir chapitre III) [HOL87] [EZP91].

## 2. Notations et définitions de base

Un RdP est un graphe orienté composé de deux types d'objets appelés places et transitions:

les places, notées  $P_i$ , représentées par des cercles, symbolisent les états d'un système,

les transitions, notées  $T_i$ , représentées par des barres, symbolisent les événements dont l'occurrence modifie l'état du système.

Ces deux objets sont reliés par des arcs orientés décrivant les liens entre les états du système. Les liens sont des applications d'incidence:

avant notée "pré" pour les arcs reliant une place à une transition,

arrière notée "post" pour les arcs reliant une transition à une place.

### Définition 2.1:

Un **RdP** est un quadruplet  $R = \langle P, T, \text{pré}, \text{post} \rangle$   
avec:

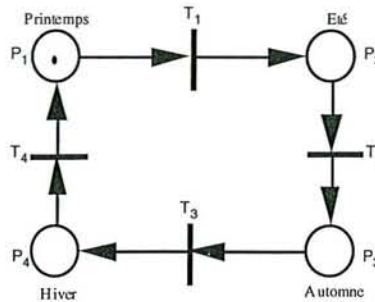
$P$  : un ensemble de places où  $|P|$  est le nombre fini de places du réseau,

$T$  : un ensemble de transition où  $|T|$  est le nombre fini de transitions du réseau

et  $P \cap T = \emptyset$

$\text{pré} : P \times T \rightarrow \{0, 1\}$ ,

$\text{post} : T \times P \rightarrow \{0, 1\}$ .

**Exemple 2.1:****Figure 2.1** *Quatre saisons*

L'ensemble des places est  $P = \{ P_1, P_2, P_3, P_4 \}$

et l'ensemble des transitions est  $T = \{ T_1, T_2, T_3, T_4 \}$ .

Le RdP de la figure 2.1 décrit le séquençement des saisons. La présence d'une marque dans une place indique la saison actuelle. Nous pouvons remarquer que ce réseau ne peut contenir qu'un jeton et un seul puisqu'une seule saison à la fois a lieu. Le franchissement (ou tir) d'une transition, indique le passage d'une saison à l'autre. Le tir de  $T_1$  indique la fin du printemps et le début de l'été.

**Définition 2.2:**

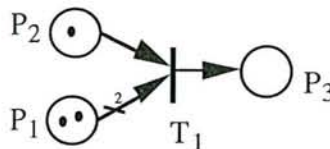
Un **RdP généralisé** est un quadruplet  $R = \langle P, T, \text{pré}, \text{post} \rangle$

où:

$\text{pré} : P \times T \rightarrow \mathbb{N}$  ( $\mathbb{N}$  est l'ensemble des entiers naturels)

$\text{post} : P \times T \rightarrow \mathbb{N}$

les applications pré (respectivement post), font correspondre à tout couple  $(P_i, T_j) \in P \times T$  (respectivement  $(T_j, P_i) \in T \times P$ ) un entier qui est le poids de l'arc. Lors de la représentation graphique, en règle générale, si un arc n'a pas de poids associé (appelé aussi valuation), c'est que ce poids vaut 1 (définition 2.1) sinon l'arc est biffé et une valuation lui est attribuée.

**Exemple 2.2:****Figure 2.2** *Assemblage*



Le RdP généralisé de la figure 2.2 représente une fonction dans une chaîne d'assemblage.  $P_1$  et  $P_2$  sont des stocks d'entrée (roues et cadres de bicyclettes par exemple),  $P_3$  est un stock de sortie (bicyclettes), la transition  $T_1$  représente la chaîne d'assemblage. Pour fabriquer une bicyclette, il faut 2 roues et un cadre, aussi  $\text{pré}(P_1, T_1)=2$ ,  $\text{pré}(P_2, T_1)=1$  et  $\text{post}(T_1, P_3)=1$ .

**Définition 2.2.1:**

On note  $\mathbf{T}_i$ , l'ensemble des places reliées à la transition  $T_i$  par un arc pré (appelées aussi places en entrée de  $T_i$ ):

$$\mathbf{T}_i = \{P_j \in P / \text{pré}(P_j, T_i) > 0\}.$$

On note  $\mathbf{T}_i^*$ , l'ensemble des places reliées à la transition  $T_i$  par un arc post (appelées aussi places en sortie de  $T_i$ ):

$$\mathbf{T}_i^* = \{P_j \in P / \text{post}(T_i, P_j) > 0\}.$$

On note  $\mathbf{P}_i$ , l'ensemble des transitions reliées à la place  $P_i$  par un arc post (appelées aussi transitions en entrée de  $P_i$ ):

$$\mathbf{P}_i = \{T_j \in T / \text{post}(P_i, T_j) > 0\}.$$

On note  $\mathbf{P}_i^*$ , l'ensemble des transitions reliées à la place  $P_i$  par un arc pré (appelées aussi transitions en sortie de  $P_i$ ):

$$\mathbf{P}_i^* = \{T_j \in T / \text{pré}(T_j, P_i) > 0\}.$$

Le marquage est une application de l'ensemble des places vers  $\mathbb{N}$ . Chaque place pourra contenir à un instant donné une, plusieurs ou pas de marques. La présence d'une marque, appelée aussi jeton, dans une place peut symboliser qu'un état est actif ou qu'une ressource est disponible, etc....

**Définition 2.3:**

Un **RdP marqué** est défini par  $\langle R, M \rangle$ ,

où

$R$  est un RdP,

$M$  une application de  $P$  dans  $\mathbb{N}$  où pour toute place  $P_i \in P$ ,  $M(P_i)$  est le nombre de marques contenues par la place  $P_i$ .

**Définition 2.3.1:**

On définit un **RdP marqué**, par le couple  $\langle R, M_0 \rangle$ ,

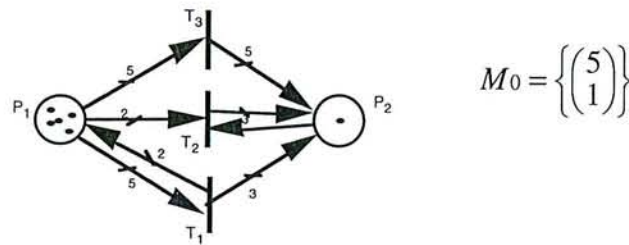
où

$R$  est un RdP

$M_0$  un ensemble fini non vide de marquages initiaux.

Un marquage est donc un vecteur de dimension  $|P|$  dont chaque élément d'indice  $i$  est le nombre de marques contenues dans la place  $P_i$ .

**Exemple 2.3:**



$$M_0 = \left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix} \right\}$$

Figure 2.3 RdP généralisé

L'évolution du marquage simule la dynamique du système. Ce changement d'état se fait par franchissement de transitions. Pour être franchie, une transition doit être validée. Le franchissement d'une transition  $T_j$  est instantané (de durée posée nulle), il consiste à retirer  $\text{pré}(P_i, T_j)$  marque(s) des places  $P_i$  ( $P_i \in {}^*T_j$ ) en entrée de  $T_j$  puis de rajouter  $\text{post}(T_j, P_k)$  marque(s) aux places  $P_k$  ( $P_k \in T_j^*$ ) en sortie de  $T_j$ .

**Définition 2.4:**

Une transition est **validée** si et seulement si la pré-condition suivante est satisfaite:

$$\forall P_i \in {}^*T_j \quad M(P_i) \geq \text{pré}(P_i, T_j)$$

on note  $M(T_j >)$  cette relation de  $\mathbb{N}^n \times T$  dans  $\mathbb{N}^n$  où  $n = |P|$ .

Le nouveau marquage  $M'$  obtenu par franchissement de  $T_j$  est noté:

$$M(T_j > M')$$

$$\text{avec } M'(P_i) = M(P_i) - \text{pré}(P_i, T_j) + \text{post}(T_j, P_i) \quad [I.2.1]$$

### 3. Représentation matricielle

La représentation graphique ayant ses faiblesses et ses limites, les RdP disposent d'une description matricielle du graphe orienté.

L'application d'incidence avant peut être représentée par une matrice  $n \times m$  ( $n=|P|$  et  $m=|T|$ ) notée  $C^-$  ( $C^- \in \mathbb{N}^{n \times m}$ ) comme suit:

pré	$T_1$	...	$T_m$
$P_1$	$\text{pré}(P_1, T_1)$	...	$\text{pré}(P_1, T_m)$
...	...	...	...
$P_n$	$\text{pré}(P_n, T_1)$	...	$\text{pré}(P_n, T_m)$

L'application d'incidence arrière peut être représentée par une matrice notée  $C^+$  ( $C^+ \in \mathbb{N}^{n \times m}$ ) comme suit:

post	$T_1$	...	$T_m$
$P_1$	$\text{post}(T_1, P_1)$	...	$\text{post}(T_m, P_1)$
...	...	...	...
$P_n$	$\text{post}(T_1, P_n)$	...	$\text{post}(T_m, P_n)$

L'évolution des marquages (définition 2.4) se fait toujours en rajoutant au marquage précédent la quantité  $[\text{post}(T_j, P_i) - \text{pré}(P_i, T_j)]$ . On peut alors introduire la matrice d'incidence  $C$  ( $C \in \mathbb{Z}^{n \times m}$ ).

**Définition 3.1:**

Un réseau  $R$  est décrit par sa **matrice d'incidence**  $C$ , application de  $P \times T$  dans  $\mathbb{Z}$  (ensemble des entiers relatifs) telle que:

$$C(P_i, T_j) = \text{post}(T_j, P_i) - \text{pré}(P_i, T_j) \quad \forall P_i \in P \text{ et } \forall T_j \in T \quad [\text{I.2.2}]$$

$$C = C^+ - C^-$$

$C$  est une matrice à  $|P|$  lignes et  $|T|$  colonnes,  $C \in \mathbb{Z}^{n \times m}$ .

	$T_1$	...	$T_m$
$P_1$	$\text{post}(T_1, P_1) - \text{pré}(P_1, T_1)$	...	$\text{post}(T_m, P_1) - \text{pré}(P_1, T_m)$
...	...	...	...
$P_n$	$\text{post}(T_1, P_n) - \text{pré}(P_n, T_1)$	...	$\text{post}(T_m, P_n) - \text{pré}(P_n, T_m)$

**Exemple 3.1:**

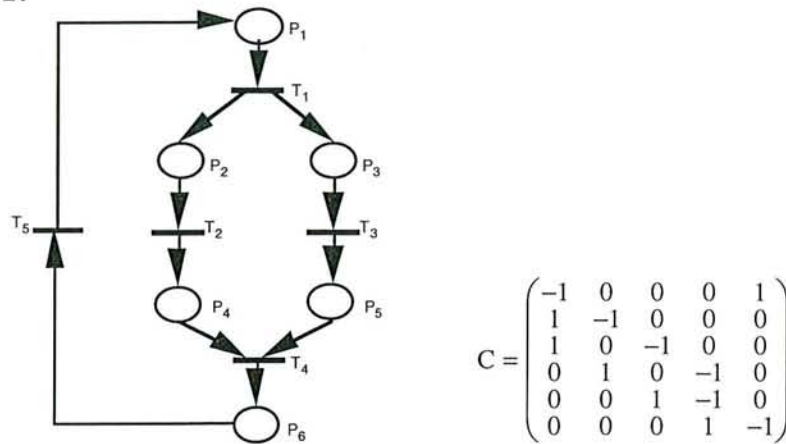


Figure 3.1 Matrice d'incidence

**Définition 3.2:**

Une **séquence de franchissement**, notée  $\sigma$ , est une suite de transitions  $T_i$  validées et franchies séquentiellement.

Soit  $\bar{\sigma}$  un vecteur de  $\mathbb{N}^m$  ( $m = |T|$ ) où la  $j^{\text{ème}}$  composante de  $\bar{\sigma}$  indique le nombre d'occurrence de la transition  $T_j$  dans la séquence  $\sigma$ .  $\bar{\sigma}$  est appelé *vecteur caractéristique*.

Pour l'exemple 3.1, la séquence  $\sigma = "T_1 T_2"$  alors  $\bar{\sigma} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$  représente le franchissement des

transitions  $T_1$  et  $T_2$ . Il y a perte d'information (ordre des transitions franchies) dans cette

représentation vectorielle du franchissement. Partant du marquage initial  $M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$  on atteint

le marquage  $M = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$  par franchissement de la séquence  $\sigma$ . Ce qui se note:  $M_0(\sigma) > M$ .

Pour simplifier la notation, le vecteur caractéristique est noté  $\sigma$ .

**Remarque 3.1:**

Nous pouvons observer que, pour calculer le marquage atteint après franchissement de la transition  $T_j$ , il suffit d'ajouter au marquage initial  $M_0$  la  $j$ ème colonne de la matrice d'incidence. Pour extraire la  $j$ ème colonne de  $C$ , il suffit de la multiplier à droite par un vecteur de  $\mathbb{N}^m$  tel que tous ses éléments soient nuls sauf la  $j$ ème composante qui vaut 1.

Pour l'exemple 3.1 (transition  $T_1$ )

$$\begin{pmatrix} -1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

On obtient alors l'**équation fondamentale** qui permet de calculer le nouveau marquage ( $M$ ) atteint depuis un marquage initial ( $M_0$ ), par franchissement d'une séquence ( $\sigma$ ) :

$$M = M_0 + C.\sigma \tag{I.3.1}$$

L'évolution d'un réseau peut être analysée par cette équation, aussi, remarquons la similitude de cette dernière avec l'équation d'état des systèmes discrets:

$$x_{k+1} = A.x_k + B.u_k \tag{I.3.2}$$

L'état d'un système à l'instant  $k+1$  décrit dans l'équation [I.3.1] par  $M$  est fonction de l'état à l'instant  $k$  ( $M_0$ ) et de la commande  $\sigma$  qu'on lui applique.

#### 4. Propriétés essentielles

La notion de **vivacité** est fondamentale dans la dynamique d'un RdP car elle exprime la possibilité pour toute transition d'être franchie à un instant de la vie du réseau [BRA83] [LEC84] [LEC86]. Dans le cas contraire, on dit qu'il y a un **blocage**, en d'autres termes, une partie du réseau est (ou bien est devenue) inactive [PAS92]. En effet, si une transition est infranchissable, certains états du système ne seront jamais atteints, il y a donc peut être erreur de conception car le système devient incapable de réaliser certaines fonctionnalités

pour lesquelles il a été conçu. Aussi nous nous attacherons à affiner les moyens de détecter ce genre de défauts.

Nous nous intéresserons aussi à la quantité de marques que peut accumuler une place. Nous chercherons donc à trouver les bornes supérieures des marquages accessibles par un système. Cette information est importante car elle peut signifier, par exemple, le nombre maximum de pièces que peut traiter une machine dans une chaîne de production.

**Définition 4.1:**

Une place  $P_i$  est dite **bornée** ou **k-bornée** pour un marquage initial  $M_0$  si:

$$\exists k < \infty, k \in \mathbb{N} \quad \text{tel que} \quad M(P_i) \leq k, \text{ pour tout marquage}$$

accessible à partir de  $M_0$ .

Si cette condition est vérifiée pour toutes les places du réseau, alors il est **borné**.

**Définition 4.2:**

Soit  $\langle R, M_0 \rangle$  un RdP marqué, si pour tout marquage accessible à partir de  $M_0$ ,  $\langle R, M_0 \rangle$  est 1-borné alors il est **sauf**.

**Définition 4.3:**

Une transition  $T_i$  est **1 fois vivante** si et seulement si elle peut être tirée une fois à partir de  $M_0$ .

Une transition  $T_i$  est **k fois vivante** si et seulement si elle peut être tirée k fois à partir de  $M_0$ .

Une transition  $T_i$  est **infiniment vivante** si et seulement si elle peut être tirée une infinité de fois à partir de  $M_0$ .

Une transition  $T_i$  est **fortement vivante** pour un marquage initial  $M_0$  si pour tout marquage  $M$  accessible à partir de  $M_0$ , il existe une séquence de franchissement contenant  $T_i : M \langle T_i \rangle$ .

**Définition 4.4:**

Un réseau  $\langle R, M_0 \rangle$  est **vivant** pour un marquage initial  $M_0$  si et seulement si toutes ses transitions sont une fois vivantes pour  $M_0$ .

Un réseau  $\langle R, M_0 \rangle$  est **fortement vivant** pour un marquage initial  $M_0$  si et seulement si, pour toute transitions  $T_i$  et tout marquage  $M_i$  accessible à partir de  $M_0$ , il existe une séquence  $\sigma$  comprenant  $T_i$  et franchissable à partir de  $M_i$ .

**Définition 4.5:**

Un RdP contient un **conflit structurel** lorsqu'une même place  $P_i$  sert à alimenter plusieurs transitions:

$$\exists P_i \in P \quad \text{tel que} \quad |P_i^\bullet| > 1$$

Le conflit est **effectif** lorsque le marquage de  $P_i$  est insuffisant pour permettre le franchissement simultané des transitions  $T_j$  en sortie de  $P_i$ :

$$M(P_i) < \sum_{j=1}^s \text{pré}(P_i, T_j) \quad \text{où } T_j \in P_i^\bullet \text{ et } s = |P_i^\bullet|$$

Une propriété très riche en informations pour les RdP est l'**invariant**. Il en existe deux types:

les invariants de marquages ou P-semi-flots (S-invariants),

les invariants de franchissements ou T-semi-flots (T-invariants).

Dans les deux cas, on cherche à caractériser les états conservatifs du réseau, c'est-à-dire des bilans de marquages constants ou un état toujours accessible.

**P-semi-flot:** on cherche des vecteurs  $F \in \mathbb{N}^n$ :

$$F = (f_1, f_2, \dots, f_n) \quad \text{tel que} \quad \sum_{i=1}^n f_i M(P_i) = K \quad K \text{ étant une constante.} \quad [I.4.1]$$

Si de tels vecteurs existent alors la relation fondamentale [I.3.1]

$$\text{devient} \quad F^T M_0 = F^T M + F^T C \sigma,$$

$$\text{d'après [I.4.1]} \quad F^T M_0 = F^T M,$$

$$\text{donc} \quad F^T C \sigma = 0 \quad \forall \sigma \in \mathbb{N}^m \quad \text{d'où} \quad F^T C = 0.$$

Un **invariant** (ou support)  $I$  est un sous-ensemble de  $P$  tel qu'il existe un P-semi-flot  $F$  avec  $f_i \neq 0$  si et seulement si  $P_i \in I$ .

Un **invariant** est **minimal** si et seulement si il ne contient pas d'autre invariant que l'ensemble vide et lui-même.

**Théorème 4.1:**

Un RdP  $\langle R, M_0 \rangle$  est **vivant** si tout P-semi-flot minimal F de support minimal vérifie:

$$F^T M_0 \geq \max_{T_i \in T} \sum_{P_j \in {}^*T_i} F(P_j) \text{Pré}(P_j, T_i)$$

■

**T-semi-flot:** on cherche des vecteurs  $\sigma \in \mathbb{N}^m$ :

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_m) \quad \begin{array}{l} \text{tels que} \quad M = M_0, \\ \text{donc} \quad C\sigma = 0, \end{array}$$

$\sigma$  étant une séquence de franchissement, elle représente la possibilité pour un RdP de se réinitialiser et donc de pouvoir réexécuter la séquence  $\sigma$ .

Ces propriétés étant importantes, nous leur consacrerons le chapitre II.

**Propriétés structurelles:**

Les RdP sont classifiés en plusieurs catégories et ceci grâce à leurs propriétés, notamment structurelles. Le but de ce travail n'étant pas de développer les RdP mais de les utiliser, les classes importantes sont présentées dans le tableau suivant [DAV89]:




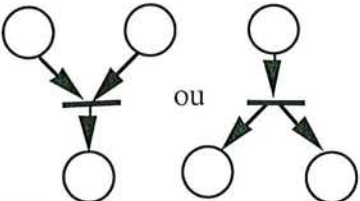

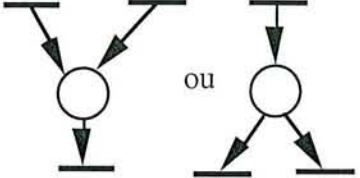
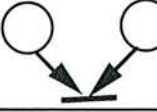
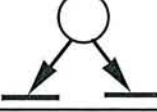

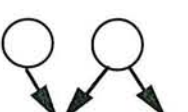
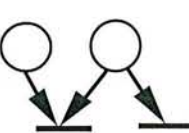
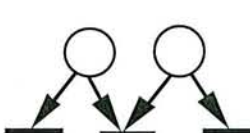
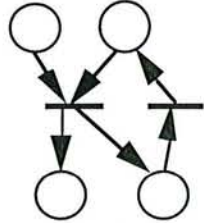
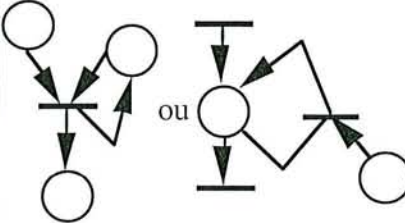
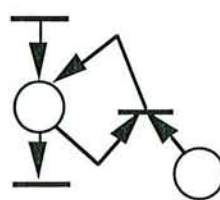
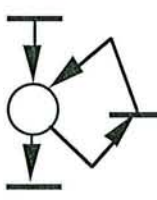
RdP	On peut trouver	On ne peut pas trouver
Graphe d'état		
Graphe d'évènements		
Sans conflit		
à choix libre		
Simple		
pur		
Sans boucle		

Figure 4.1 Quelques classes de RdP

## 5. Graphe des marquages

### 5.1. Rappels sur la théorie des graphes

Un graphe est un couple  $G(M, U)$  d'ensembles où  $M$  est un ensemble d'éléments  $M_i$  appelés sommets du graphe ( $|M| = n$ ) et  $U$  est un ensemble de lignes reliant seulement deux sommets distincts ou non. Chaque ligne est notée  $(M_i, M_j)$  et porte le nom d'arc si elle est orientée et le nom d'arête si elle ne l'est pas [TIT79] [DAR86].

#### Définition 5.1.1:

Soit une suite de sommets (non nécessairement distincts):

$$M_{i1}, \dots, M_{ip}, M_{ip+1}, \dots, M_{iq}$$

cette suite est une **chaîne** si:

$$(M_{ip}, M_{ip+1}) \in U \text{ ou } (M_{ip+1}, M_{ip}) \in U \text{ pour } p=1, q-1$$

si de plus  $(M_{iq}, M_{i1}) \in U$  ou  $(M_{i1}, M_{iq}) \in U$ , cette chaîne est un **cycle**,

cette suite est une **chemin** si:

$$(M_{ip}, M_{ip+1}) \in U \text{ pour } p=1, q-1$$

si de plus  $(M_{iq}, M_{i1}) \in U$ , ce chemin est un **circuit**.

#### Définition 5.1.2:

Soit un graphe  $G(M, U)$ ; on définit deux relations binaires de connexité.

Relation binaire de **connexité simple**  $R_s$ :

$M_i R_s M_j$  si et seulement si il existe une chaîne entre  $M_i$  et  $M_j$  ou si et seulement si  $M_i = M_j$ .

Relation binaire de **connexité forte**  $R_f$ :

$M_i R_f M_j$  si et seulement si il existe un circuit passant par  $M_i$  et  $M_j$  ou si et seulement si  $M_i = M_j$ .

On montre que ces relations binaires sont des relations d'équivalences (réflexives, symétriques et transitives). Chacune d'elles induit une partition de  $M$  en classes d'équivalence qui portent les noms respectivement de composantes simplement connexes (s-connexes) et de composantes fortement connexes (f-connexes) du graphe  $G$ .

## 5.2. Application aux RdP

Le moyen le plus direct pour analyser un RdP est de chercher à recenser l'ensemble des états qu'il peut atteindre. Pour cela, nous pouvons tracer un graphe dont les sommets sont les marquages accessibles et les arcs les transitions dont le franchissement permet d'atteindre un nouveau marquage [BRA83] [KUM84] [COM88].

### Définition 5.2.1:

L'ensemble des marquages accessibles à partir d'un marquage M est:

$$A(R, M) = \{ \bar{M} \in \mathbb{N}^n / \exists \sigma \ M(\sigma > \bar{M}) \}$$

Exemple:

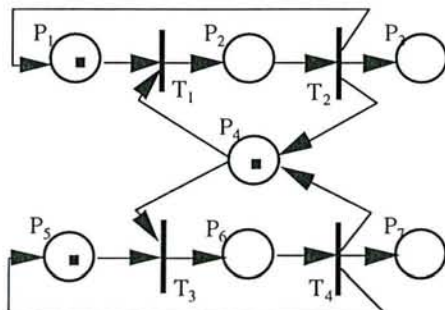


Figure 5.1 RdP d'une double file FIFO

On peut alors construire un graphe dont les sommets sont les vecteurs des marquages atteints et les arcs les transitions dont le tir permet le passage à un autre marquage (sommets). Le graphe des marquages du réseau de la figure 5.1, obtenu après franchissement des différentes transitions validées, est le suivant:

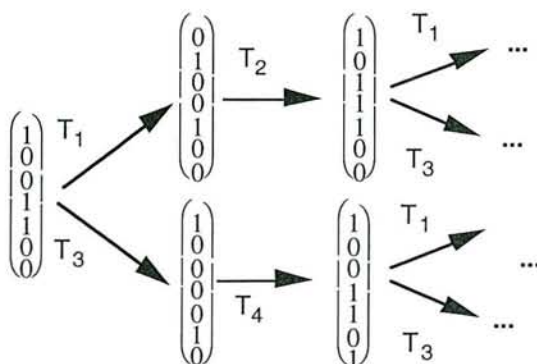


Figure 5.2 Graphe des marquages

Malgré la taille réduite du réseau de la figure 1 (7 places et 4 transitions seulement), le nombre des marquages possibles est infini. Aussi, pour donner une représentation du graphe, nous utilisons le symbole  $\omega$  comme marquage des places  $P_3$  et  $P_7$  ( $M(P_3)$  et  $M(P_7)$ ).

Le symbole  $\omega$  satisfait les relations suivantes:

$$\text{soit } \mathbb{N}_\omega = \{\omega\} \cup \mathbb{N}$$

$$\forall n \in \mathbb{N} : n < \omega, \forall n \in \mathbb{N}_\omega : n + \omega = \omega - n = \omega,$$

$$\min(n, \omega) = n, \max(n, \omega) = \omega, (n + 1) \cdot \omega = \omega, 0 \cdot \omega = \omega \cdot 0 = 0.$$

Le graphe des marquages (fig. 5.2) est ainsi transformé en un graphe de couverture:

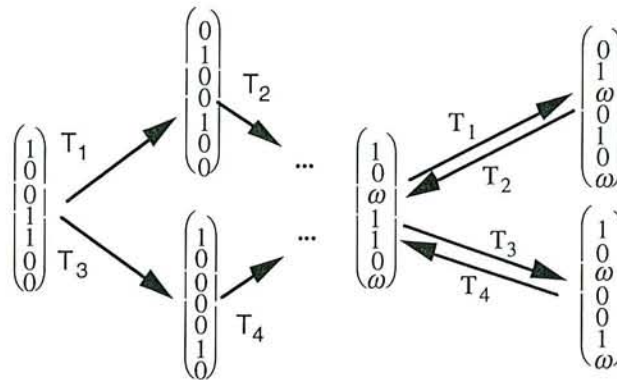


Figure 5.3 Graphe de couverture

Ce graphe présente l'ensemble des séquences de franchissement possible à partir d'un marquage initial et les marquages atteints. On peut donc conclure que les places  $P_1$ ,  $P_2$ ,  $P_4$ ,  $P_5$  et  $P_6$  sont 1-bornées. Par contre la présence du marquage  $\omega$  dans les places  $P_3$  et  $P_7$  indique qu'elles ne sont pas bornées.

Il existe une représentation mathématique plus concise de ceci grâce à une expression algébrique appelée le **langage** du RdP [RAM87] [RAM89].

Le langage du RdP de la figure 5.1 est:

$$L = (T_1 T_2)^* (T_3 T_4)^* + (T_3 T_4)^* (T_1 T_2)^*$$

où  $T_1 T_2$  est une séquence de franchissement. On doit franchir  $T_1$  suivi de  $T_2$ ,

$(T_1 T_2)^*$  : la séquence  $T_1 T_2$  peut être répétée indéfiniment,

le signe "+" est équivalent à un "ou", il exprime un choix entre deux séquences de franchissement validées.

## 6. Extensions

Les RdP étant relativement récents et disposant d'une représentation graphique attrayante, ils connaissent de nombreux développements. En effet, vu la complexité croissante des systèmes industriels et pour répondre aux besoins de modélisation toujours plus sévères, les chercheurs ont inventé de nouvelles formes de RdP plus performants et plus spécifiques [CAR90]. Cependant cette évolution est trop dense pour être présentée ici. Nous ne présenterons donc que les extensions qui nous intéressent par la qualité de leurs réponses aux besoins des applications industrielles auxquelles nous nous sommes intéressés et leur adéquation à l'analyse des systèmes étudiés. Nous utiliserons une nouvelle extension des RdP, les RdP continus (RdPC) introduits par DAVID et ALLA [DAV89] et plus particulièrement les RdP Hybrides (RdPH) [LEB92]. En effet les outils de production industriels ont souvent un comportement continu dans le temps, parallèlement l'informatique, indispensable pour le contrôle de ceux-ci, est par essence discrète. La modélisation de systèmes intégrant le procédé commandé et les procédures de contrôle se trouve facilitée par les RdPH grâce à leur nature hybride (continue et discrète). Les RdPC s'apparentent aux RdP temporisés aussi nous présenterons d'abord ces derniers pour faciliter la transition, avant de déduire les RdPH.

### 6.1. RdP Temporisés (RdPT)

#### 6.1.1. Présentation

A l'origine les RdP ne prennent pas en compte la notion de temps, les systèmes sont décrits avec leur dynamique et l'enchaînement des états accessibles, mais le temps d'exécution d'une tâche et donc de franchissement d'une transition n'apparaît pas. Cependant, il est difficile de concevoir un outil de production indépendant des contraintes de temps. Toute installation industrielle est soumise à une cadence, ce qui nécessite des calculs d'optimisation du temps et d'ordonnancement du passage des pièces fabriquées sur les machines.

Aussi pour remédier à cette incapacité à répondre à un impératif de l'industrie, les RdP Temporisés (RdPT) ont été introduits. Les RdPT pallient à cette lacune, par deux approches équivalentes:

- associer des temporisations aux transitions, comme le fait C. RAMCHANDANI [RAM73], le RdP est dit T-temporisé,

- associer des temporisations aux places, comme le fait J. SIFAKIS [SIF77], le RdP est dit P-temporisé.

Le passage d'une représentation à l'autre est aisé [CHR84]. Ces deux types de RdPT sont qualifiés de non autonomes par opposition aux réseaux classiques dits autonomes puisque leur évolution est dépendante du monde extérieur. Les RdPT sont efficaces dans la modélisation des problèmes d'ordonnancement, souvent posés par les ateliers flexibles, mais aussi dans l'évaluation des performances des systèmes notamment par la recherche de séquences permettant un fonctionnement à vitesse maximale.

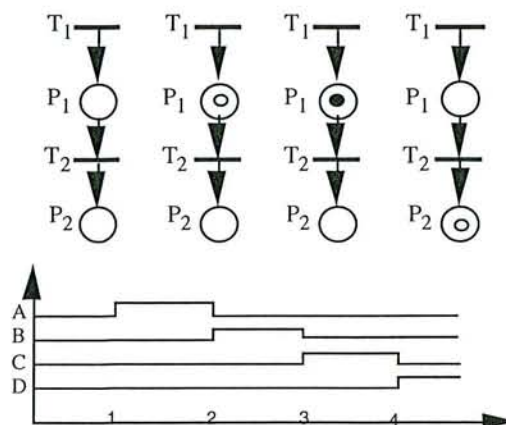
### 6.1.2. Définitions

Un **RdPT** est un couple  $\langle R, \tau \rangle$  où:

$R = \langle P, T, \text{pré}, \text{post} \rangle$  est un RdP,

$\tau$  est une application de  $P$  dans  $\mathbb{Q}^+ \cup \{0\}$  telle que  $\forall P_i \in P, \tau(P_i)$  est le retard associé à la place  $P_i$ ,  $\mathbb{Q}$  est l'ensemble des nombres rationnels.

Lorsqu'une marque arrive dans une place  $P_i$ , elle est indisponible pendant un temps  $\tau(P_i)$  (donc, ne peut servir à valider les transitions  $T_j$  vérifiant  $\text{pré}(P_i, T_j) > 0$ ). Au bout du laps de temps  $\tau(P_i)$  cette marque redevient disponible et peut valider les transitions en aval. Le mécanisme de tir est alors identique à celui des RdP classiques.



**Figure 6.1** Fonctionnement d'un RdPT

Les états en ordonnées (A, B, C, D) indiquent la présence de marques disponibles ou indisponibles dans les places  $P_1$  et  $P_2$ :

A indique la présence de marques indisponibles dans  $P_1$ ,

B indique la présence de marques disponibles dans  $P_1$ ,

C indique la présence de marques indisponibles dans  $P_2$ ,

D indique la présence de marques disponibles dans  $P_2$ .

A l'instant 1 la transition  $T_1$  est franchie, ce qui provoque l'arrivée d'une marque dans  $P_1$ , elle sera indisponible pendant  $\tau(P_1)$  c'est-à-dire jusqu'à l'instant 2. Entre les instants 2 et 3 la marque dans  $P_1$  est disponible. A l'instant 3 la transition  $T_2$  étant validée depuis l'instant 2, elle est tirée et reste indisponible jusqu'à l'instant 4.

Le marquage d'un RdP temporisé se décompose en  $M = M^r + M^n$ ,  $M^r$  et  $M^n$  sont respectivement, le marquage réservé ou non disponible et le marquage non réservé ou disponible. Seul le marquage  $M^n$  peut servir à valider des transitions.

### 6.1.3. Fonctionnement en régime stationnaire

Pour permettre l'étude du régime stationnaire, nous allons rappeler les relations importantes liant le marquage initial  $M_0$ , les retards associés aux places et les fréquences de franchissements des transitions.

Soient:

$(Q(t))^T = [q_1(t), q_2(t), \dots, q_n(t)]$  vecteur marquage du réseau à l'instant  $t$ ,

$(X(t))^T = [x_1(t), x_2(t), \dots, x_n(t)]$  vecteur de franchissement à l'instant  $t$ ,

la relation [I.3.1] devient:

$$Q(t) = Q(t_0) + C.X(t) \quad [I.6.1]$$

si  $\Delta t = t - t_0$

$$\text{alors } \frac{\Delta Q(t)}{\Delta t} = \frac{Q(t) - Q(t_0)}{\Delta t} = C \frac{X(t)}{\Delta t}$$

$\frac{\Delta Q(t)}{\Delta t}$  est le vecteur représentant la variation du marquage dans les places, pendant

l'intervalle de temps  $\Delta t$ . On appelle  $F = \frac{X(t)}{\Delta t}$  le vecteur des fréquences de franchissement

des transitions pendant l'intervalle de temps  $\Delta t$ .

**Propriété 6.1.3.1:**

Soit  $R_T$  un RdP P-temporisé. Le fonctionnement à vitesse maximum conduit à un **fonctionnement périodique** au bout d'un temps fini, pour tout marquage initial tel que  $R_T$  soit borné.

Si le fonctionnement est périodique, alors  $F$  est un T-semi-flot tel que :

$$C.F = 0 \quad [I.6.2]$$

soient  $Q(t_1), Q(t_2), \dots, Q(t_s)$  les marquages successifs atteints par un RdPT au cours d'un fonctionnement périodique et  $d_1, d_2, \dots, d_s$  leurs durées respectives,

alors la valeur moyenne  $\bar{Q}$  de  $Q(t)$  est: 
$$\bar{Q} = \frac{\sum_{i=1}^s d_i Q(t_i)}{\sum_{i=1}^s d_i}$$

Si  $J$  est P-semi-flot du RdPT alors:

$$J^T \bar{Q} = J^T M_0 = J^T Q(t_0).$$

Posons  $Z = \begin{pmatrix} z_1 & 0 & \dots & 0 \\ 0 & z_2 & 0 & \dots \\ \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & z_n \end{pmatrix}$  la matrice des retards associés aux places, alors:

$$\bar{Q} \geq ZC^+F$$

Cette relation exprime le fait que le marquage moyen d'une place est supérieur ou égal à la fréquence moyenne d'arrivée des marques à cette même place pendant le retard associé. Avec les deux relations ci-dessus, on obtient une inéquation liant le marquage initial, les retards et les fréquences de franchissements:

$$J^T \bar{Q} = J^T M_0 \geq J^T ZC^+F$$

Les fréquences de franchissements en régime stationnaire vérifient donc les relations:



$$\begin{cases} C.F = 0 & \text{avec} & F > 0 \\ J^T M_0 \geq J^T ZC^+ F \end{cases} \quad [I.6.3]$$

#### 6.1.4. Fonctionnement en vitesse propre

Cette notion a été introduite par J. SIFAKIS [SIF77]. Un RdPT fonctionne en vitesse propre si toute marque est retirée dès qu'elle devient disponible. C'est donc un fonctionnement à vitesse maximale où aucune marque ne reste disponible. Pour cela il doit satisfaire les équations suivantes:

$$\begin{cases} C.F = 0 & \text{avec} & F > 0 \\ J^T M_0 = J^T ZC^+ F \end{cases} \quad [I.6.4]$$

les retards  $z_i$  sont alors exactement égaux aux  $\tau(p_i)$ , temps d'indisponibilité des marques.

Pour l'exemple de la figure 6.1, la transition  $T_2$  serait alors tirée à l'instant 2 et non à l'instant 3.

## 6.2. RdP Continus (RdPC)

### 6.2.1. Présentation

Introduits par R. DAVID et H. ALLA [DAV87] [DAV89] [LEB93], ils sont issus des RdP T-temporisés dont ils constituent une bonne approximation. Les places des RdPC peuvent contenir un marquage réel, plus proche de la réalité physique pour certains cas. En effet il n'est pas très précis d'utiliser une représentation entière pour le contenu d'une cuve. Le franchissement des transitions se fait de façon continu et à une vitesse qui peut être constante ou variable.

Dans un RdP si on partitionne chaque marque en deux, l'arbre des marquages contiendra de nouveaux sommets correspondants à des états intermédiaires du système modélisé. Si on divise chaque marque par un nombre  $k$  très grand, le nombre de sommets atteints est infini

et on va tendre vers un fonctionnement continu. Ceci est facilement vérifiable sur le RdP suivant [DAV90]:

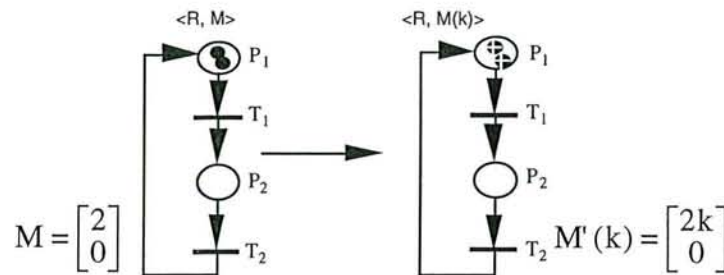


Figure 6.2 Introduction aux RdPC

Marquages accessibles pour  $\langle R, M \rangle$

$$\begin{matrix} [2] & \xrightarrow{T_1} & [1] & \xrightarrow{T_1} & [0] \\ [0] & \xleftarrow{T_2} & [1] & \xleftarrow{T_2} & [2] \end{matrix}$$

Marquages accessibles pour  $\langle R, M'(k) \rangle$

$$\begin{matrix} [2k] & \xrightarrow{T_1} & [2k-1] & \dots & [k] & \dots & [1] & \xrightarrow{T_1} & [0] \\ [0] & \xleftarrow{T_2} & [1] & \dots & [k] & \dots & [2k-1] & \xleftarrow{T_2} & [2k] \end{matrix}$$

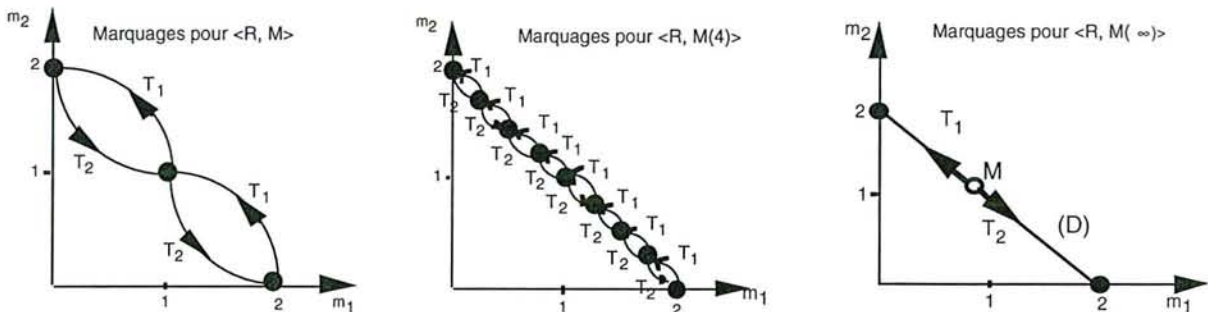


Figure 6.3 Trajectoire des marquages

Dans le RdP discret de la figure 6.3, le passage du marquage  $[2 \ 0]$  au marquage  $[1 \ 1]$  se fait par franchissement de la transition  $T_1$  puis le retour au marquage  $[2 \ 0]$  par franchissement de  $T_2$ . Par contre, dans un RdPC, les transitions  $T_1$  et  $T_2$  sont franchies continûment. Il y a donc une infinité de marquages accessibles, qui sont les points de la droite (D). On peut alors déduire la définition suivante:

Une transition  $T_j$  est **validée** si toutes les places  $P_i$  en entrée de  $T_j$  ( $P_i \in \bullet T_j$ ) ont un marquage non nul:

$$\forall P_i \in \bullet T_j \quad M(P_i) > 0.$$

Le franchissement d'une transition  $T_j$  consiste à retirer une quantité  $q_j \in \mathbb{R}^+$  (ensemble des réels positifs) aux places en entrée de  $T_j$  et à la rajouter aux places en sortie de  $T_j$ .  $q_j$  appelée **quantité de franchissement**, est telle que:

$$0 \leq q_j \leq \min[M(P_i)] \quad P_i \in \text{``}T_j\text{'}$$

On ne peut plus parler de séquence de franchissement mais de trajectoire de franchissement. Le vecteur caractéristique est formé des quantités de franchissement  $q_j$  ( $1 \leq j \leq |T|$ ).

### 6.2.2. Définitions

#### Définition 6.2.1:

Un **RdP Continu à vitesses constantes** (RdPCC) est un sextuplet:

$$\langle P, T, \text{pré}, \text{post}, V, M_0 \rangle$$

$V$  : est une application de  $T$  dans  $\mathbb{R}^+ \cup \{\infty\}$ ,  $V(T_j)$  correspond à la vitesse maximale de franchissement de la transition  $T_j$

$M_0$ , le marquage initial, est un vecteur de dimension  $|P|$  composé de réels positifs ou nuls.

#### Définition 6.2.2:

Une place  $P_i$  est **alimentée** à un instant  $t$  si et seulement s'il y a au moins une transition  $T_j \in P_i\text{'}$  telle que  $v_j(t) > 0$ .

Pour être franchissable, une transition doit être validée. Cependant la pré-condition est différente du cas des RdP discrets. En effet, une place  $P_i$ , même vide, peut valider une transition  $T_j \in P_i\text{'}$  si elle est elle même alimentée par une transition  $T_k$  pour laquelle  $v(T_k) > 0$  ( $T_k \in \text{``}P_i\text{'}$ )

#### Définition 6.2.3:

Une transition  $T_j$  est **validée** à un instant  $t$ , si  $\forall P_i \in \text{``}T_j\text{'}$  alors l'une des deux conditions suivantes est vérifiée:

- 1)  $M(P_i) > 0$
- 2)  $P_i$  est alimentée.

Si la première condition est satisfaite pour toutes les places  $P_i \in \text{``}T_j\text{'}$  alors la transition  $T_j$  est **fortement validée** sinon elle est **faiblement validée**.

**Propriété 6.2.1:**

La vitesse de franchissement d'une transition  $T_j$  fortement validée à un instant  $t$  est égale à sa vitesse maximale soit  $v_j(t) = V_j$ .

Posons:

$$\text{pré}(P_i) = \sum_{k=1}^m \text{post}(P_i, T_k) \cdot v_k \quad \text{et} \quad \text{post}(P_i) = \sum_{k=1}^m \text{pré}(P_i, T_k) \cdot v_k \quad \text{les}$$

quantités de marques qui entrent dans  $P_i$  (resp. sortent de  $P_i$ ) par unité de temps.

$B_i = \text{pré}(P_i) - \text{post}(P_i)$  est donc un bilan dynamique ou l'accroissement du nombre de marques par unité de temps.

**Relation fondamentale:**

Le calcul de l'évolution du marquage est donné par:

$$m_i(t+dt) = m_i(t) + \sum_{k=1}^m [\text{post}(P_i, T_k) - \text{pré}(P_i, T_k)] \cdot v_k(t) \cdot dt \quad 1 \leq i \leq n$$

$$M(t+dt) = M(t) + C \cdot v(t) \cdot dt$$

$$\text{d'où} \quad \frac{dM}{dt} = C \cdot v(t)$$

avec:

$m_i(t)$  est le marquage de la place  $P_i$  à l'instant  $t$ ,

$m_i(t+dt)$  est le marquage de la place  $P_i$  à l'instant  $t+dt$ ,

$v_k(t)$  est la vitesse de franchissement de la transition  $T_k$ ,

$\frac{dM}{dt}$  est la variation du marquage,

$v(t)$  est le vecteur vitesse de franchissement à l'instant  $t$ ,

$C$  est la matrice d'incidence.

**Propriété 6.2.4:**

La vitesse de franchissement d'une transition  $T_j$  faiblement validée à un instant  $t$  est donnée par la relation:

$$v_j(t) = \min[V_j, \min_i(B_i + v_j(t))]$$

avec  $i$  tel que  $P_i \in {}^*T_j$  et  $m_i(t) = 0$ .

Les notions essentielles des RdPC se trouvent ainsi définies. La propriété de vivacité est identique au cas des RdP classiques, quant au calcul des invariants, il ne pose pas de problème particulier puisqu'il se fait par résolution d'un système d'équations homogènes dans  $\mathbb{R}$ .

### 6.3. RdP Hybrides (RdPH)

L'idée génératrice est due au fait qu'un processus industriel, généralement complexe, n'est souvent ni complètement continu, ni complètement discret. Modéliser le contenu d'un réservoir par un nombre entier n'est pas toujours une bonne approximation surtout lorsqu'on doit en étudier l'écoulement et donc une variation continue, il a donc fallu introduire les RdPC. Cependant, modéliser les deux états marche/arrêt d'une machine, par un marquage réel et un franchissement de transitions continues, est quasi-impossible. C'est pourquoi les auteurs [LEB91] [LEB92] [DUB93] ont créé cette nouvelle extension des RdP, judicieuse et plus proche des réalités physiques modélisées: les **RdP Hybrides**.

La présence dans un RdPH de deux types de places (discrètes ou continues) ainsi que deux types de transitions a nécessité une représentation graphique appropriée, afin d'éviter toute confusion. Aussi, on représente un RdPH avec les icônes suivantes:



Figure 6.4 Nœuds d'un RdPH

#### 6.3.1. RdPH autonomes

##### Définition 6.3.1.1:

Un **RdPH non marqué**  $R^* = \langle R, h \rangle$

où:

$1/ R = \langle P, T, \text{pré}, \text{post} \rangle$  est un RdP non marqué

pré:  $P \times T \rightarrow \{0, 1\}$

post:  $P \times T \rightarrow \{0, 1\}$

2/  $h : P \cup T \rightarrow \{D, C\}$  est la fonction hybride. Elle indique pour chaque nœud s'il est Discret ou Continu.

3/ Les fonctions pré et post doivent vérifier:

si  $P_i$  et  $T_j$  sont telles que  $h(P_i) = D$  et  $h(T_j) = C$  alors  $\text{pré}(P_i, T_j) = \text{post}(P_i, T_j)$ .

Cette dernière condition permet de s'assurer qu'une place discrète liée par un arc à une transition continue gardera toujours un marquage entier.

Les définitions suivantes rejoignent celles établies pour les RdP discrets. Le comportement d'un RdP est toujours étudié pour un marquage initial donné. De plus il est important de connaître les conditions de validation des transitions.

#### Définition 6.3.1.2:

Un **RdPH marqué** est un couple  $H = \langle R^*, M_0 \rangle$  où  $R^*$  est un RdPH non marqué et  $M_0$  est un marquage initial. Le marquage initial d'une D-place est un entier naturel alors que celui d'une C-place est un nombre réel positif ou nul.

#### Définition 6.3.1.3:

Un **RdPH généralisé** est défini comme dans la définition 1 sauf pour la valuation des arcs:

si  $P_i$  est une D-place,  $\text{pré}(P_i, T_j)$  et  $\text{post}(P_i, T_j)$  sont des entiers naturels,

si  $P_i$  est une C-place,  $\text{pré}(P_i, T_j)$  et  $\text{post}(P_i, T_j)$  sont des réels positifs.

#### Définition 6.3.1.4:

Une D-Transition est **validée** si chacune des places  $P_i \in {}^*T_j$  vérifie:

$$M(P_i) \geq \text{pré}(P_i, T_j)$$

Une C-Transition est validée si chacune des places  $P_i \in {}^*T_j$  vérifie:

$$\text{si } P_i \text{ est une D-place, } M(P_i) \geq \text{pré}(P_i, T_j)$$

$$\text{si } P_i \text{ est une C-place, } M(P_i) \geq 0.$$

**Relation fondamentale:**  $M = M_0 + C\sigma$

La seule différence avec les relations fondamentales présentées précédemment est due à la particularité de la séquence de franchissement  $\sigma$ . La  $j$ ème composante de  $\sigma$  est entière si la transition correspondante est une D-transition et réelle si c'est une C-transition.

### 6.3.2. RdPH temporisés

#### Définition 6.3.2.1:

Un RdPH est un couple  $\langle R, \text{Tempo} \rangle$

où:

R est un RdPH (marqué et éventuellement généralisé)

Tempo est une application de  $T_j$  dans  $\mathbb{Q}^+$ .  $\text{Tempo}(T_j) = d_j =$  temporisation associée à la transition  $T_j$ . La temporisation associée à une D-transition  $T_j$ . La vitesse maximale de franchissement d'une C-transition  $T_j$  vaut  $V_j = 1/d_j$ .

#### Définition 6.3.2.2:

Une D-transition  $T_j$  est **validée** à un instant  $t$  si chacune des places  $P_i \in {}^*T_j$  vérifie:

$$M^n(P_i) \geq \text{pré}(P_i, T_j)$$

#### Définition 6.3.2.3:

Une C-place  $P_i$  est **pourvue** à un instant  $t$  si  $P_i$  possède au moins une C-transition d'entrée,  $T_j$ , qui est **validée**.

#### Définition 6.3.2.4:

Une C-transition  $T_j$  est **validée** à un instant  $t$

1- si chacune des D-places  $P_i \in {}^*T_j$  contient autant de marques non réservées que la quantité  $\text{pré}(P_i, T_j)$ .

2- chaque C-place de  ${}^*T_j$ , qui est vide ou qui ne contient que des marques réservées, est **pourvue**.

#### Relation fondamentale:

$$M(t) = M(0) + C \left( \sigma(t) + \int_{u=0}^t v(u) du \right) \quad [\text{I.6.5}]$$

## Exemple de RdPH

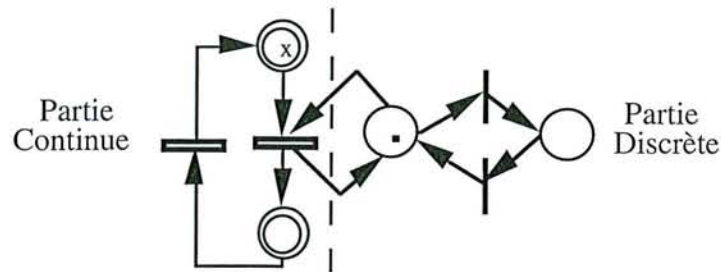


Figure 6.5 RdP Hybride

## 7. Modélisation

La présentation de quelques exemples de modèles RdP paraît être une bonne conclusion de l'efficacité d'un tel outil. Chaque système est d'abord introduit tel qu'il se présente dans la réalité avec ses objectifs et ses fonctionnalités, le modèle RdP est à son tour détaillé. Les chapitres suivants montrent l'intérêt de la modélisation par les RdP par une approche théorique puis par une application sur des processus industriels.

### Exemple 7.1: Un automatisme de commande d'une machine [VAL80]

Soit une machine de fabrication de paires de pots de yaourt. Le lait et le ferment sont respectivement préparés en parallèle dans des appareils  $A_1$  et  $A_2$  avant d'être mélangés dans un appareil  $A_3$ . Pendant ce temps, les pots sont préparés par une machine  $A_4$ . Ensuite, ils sont remplis en parallèle et mis en étuve. Le tableau ci-dessous résume les phases essentielles de ce cahier des charges. Le RdP détaillé décrivant un tel automatisme est donné par la figure 7.1.

Places	Signification	Transitions	Signification
$P_1$	Etat initial	$T_1$	Si mise en marche
$P_2, P_3, P_4$	Attente $A_i$ prête ( $i=2,3$ ou 4)	$T_2, T_3, T_4$	Si $A_i$ prête ( $i=2,3$ ou 4)
$P_5, P_6, P_7$	Préparation lait (resp. ferment, pots)	$T_5, T_6, T_7$	Si lait (resp. ferment, pots) prêt
$P_8, P_9$	Fin préparation lait (resp. ferment)	$T_8$	Si $A_4$ prête
$P_{10}$	Mélange lait-ferment	$T_9$	Si mélange prêt
$P_{15}, P_{16}$	Remplir pot 1 (resp. pot 2)	$T_{12}, T_{13}$	Si pot $i$ ( $i=1,2$ ) plein
$P_{19}$	Mettre à l'étuve	$T_{14}, T_{15}$	Début (resp. fin) temporisation



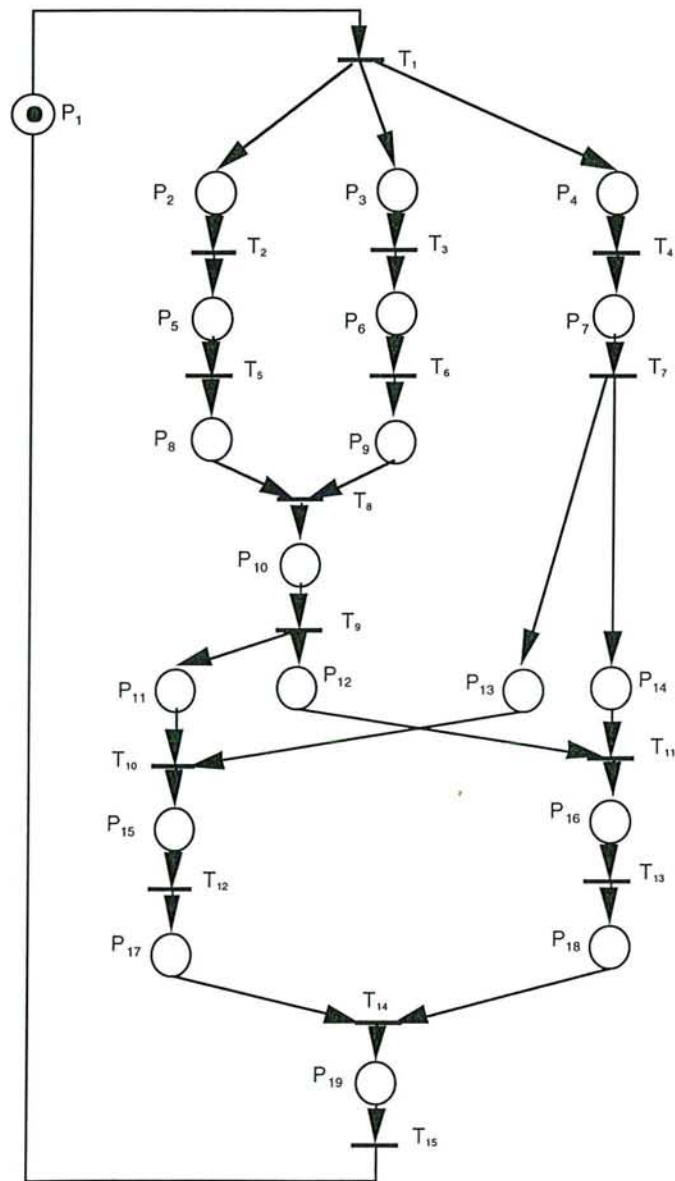


Figure 7.1 RdP d'un automatisme de fabrication.

Les places symbolisent les différents états pris par le système et les transitions les événements dont l'occurrence permet le changement d'état. La matrice d'incidence du RdP de ce système est la suivante:

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

**Exemple 7.2:** *Un protocole de communication* [GAR85]

Pour illustrer la modélisation par les RdP, nous avons choisi comme deuxième exemple, un de leurs domaines d'application les plus fréquents: les protocoles de communication complexes et plus précisément, le protocole de transport E.C.M.A. (European Communication Manufacturer's Association). C'est un protocole point à point de la couche basse (transport) du modèle de référence I.S.O.. Les étapes d'une phase de connexion-déconnexion sont les suivantes:

Les entités de transport sont initialement déconnectées. A l'état initial ( $P_1$ ) l'entité initiatrice du dialogue envoie ( $T_1$ ) une requête de connexion ( $P_1'$ ). L'entité réceptrice ( $P_6$ ) accepte la requête ( $T_1$ ) en envoyant une confirmation de connexion ( $P_2'$ ) et la connexion est établie ( $T_1''$ ). Les deux entités peuvent passer à la phase de transfert des données ( $P_2$  et  $P_7$ ). A partir de cet instant, chaque entité peut rompre la connexion en envoyant ( $T_2$  et  $T_6$ ) une requête de déconnexion ( $P_4$  et  $P_9$ ) et en passant dans un état d'attente de confirmation de déconnexion ( $P_3$  et  $P_8$ ).

A la réception d'une requête de déconnexion, deux cas peuvent se produire:

si elle est en état de transfert, l'entité connectée retourne à l'état initial en envoyant ( $T_3$  et  $T_7$ ) une confirmation de déconnexion ( $P_5$  et  $P_{10}$ )

si une collision a eu lieu, elle se met en attente d'une confirmation de déconnexion ( $T_5$  et  $T_9$ ).

A tout moment, un signal d'erreur provoque l'interruption des deux entités et leur retour à l'état initial. Une temporisation est associée à chaque place  $P_i$  (sauf les places  $P_3$  et  $P_8$ ). Le tableau suivant donne un résumé des états et événements que connaît ce système et le RdP correspondant est donné ci-dessous:

Places	Signification	Transitions	Signification
$P_1$ et $P_6$	Prêt	$T'_1$	Requête de connexion
$P'_1$	Attente confirmation connexion	$T''_1$	Connexion établie
$P'_2$	Confirmation connexion	$T_1$	Réception connexion
$P_2$ et $P_7$	Transfert de données	$T_2$ et $T_6$	Requête déconnexion
$P_3$ et $P_8$	Attente confirmation déconnexion	$T_3$ et $T_7$	Acceptation déconnexion
$P_4$ et $P_9$	Requête déconnexion	$T_4$ et $T_8$	Fin déconnexion
$P_5$ et $P_{10}$	Confirmation déconnexion	$T_5$ et $T_9$	Acceptation déconnexion

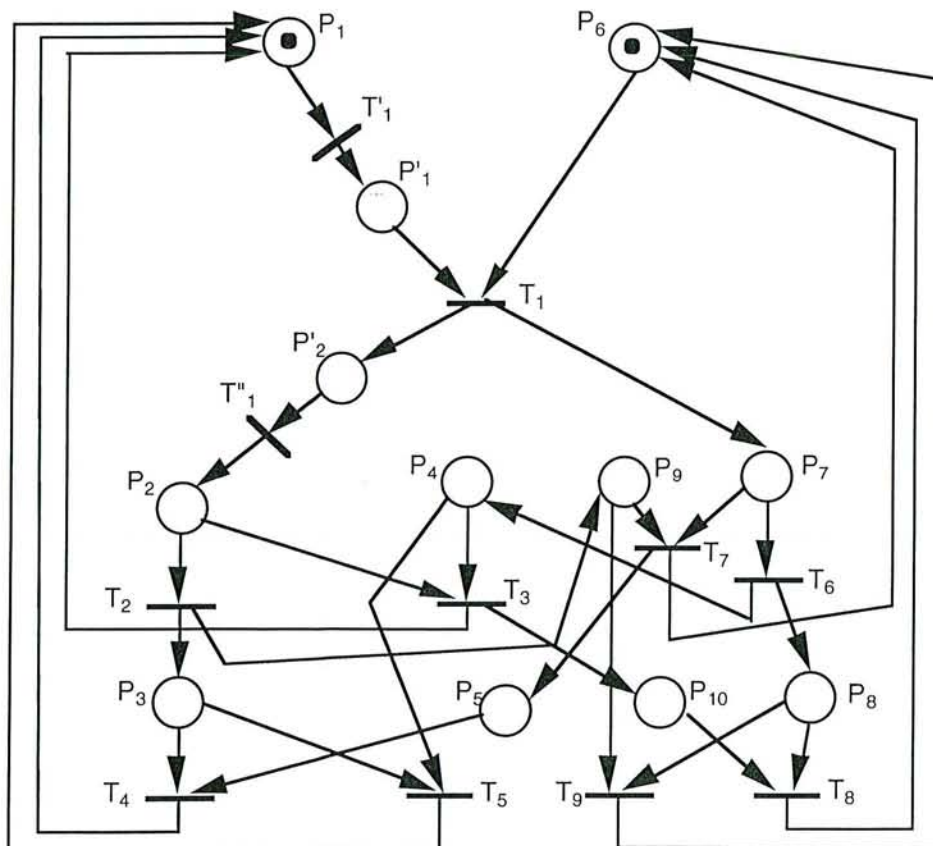


Figure 7.2 RdPT d'une phase de connexion-déconnexion (ECMA).

En application des règles de réduction des RdP [BER83], les états  $P_1$  et  $P'_1$  peuvent être confondus et la transition  $T'_1$  supprimée, de même pour les états  $P_2$ ,  $P'_2$  et la transition  $T''_1$ .

La matrice d'incidence du réseau réduit est alors la suivante:

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

## 8. Conclusion

Comme nous nous sommes efforcés de le montrer lors de ce tour d'horizon, les RdP peuvent être très utiles en automatique, notamment par la simplicité de leur représentation des systèmes et la diversification des outils mathématiques associés. Le champ d'utilisation des RdP couvre un large éventail de disciplines, on peut citer:

- l'ingénierie logicielle [LAM85],
- la spécification et la vérification des protocoles [BER82] [DUF84] [LAG93],
- la production automatisée [ROU92],
- l'architecture des ordinateurs [MOR92],
- l'étude des systèmes de commande [BRE87].

L'apport des RdP est très important dans la conception, l'analyse et le diagnostic des systèmes. Cependant, la représentation graphique n'est valable que pour des systèmes de petite dimension sinon, la lisibilité du graphe est compromise. Pour pallier à ce problème les chercheurs ont développé plusieurs méthodes telles que:

- la réduction [BER83],
- le pliage avec les RdP colorés [JEN81],
- la décentralisation [SHI90].

Nous essayerons tout au long de ce document, de reprendre certaines de ces techniques, de les présenter, de les affiner et d'en montrer l'adéquation aux problèmes industriels.

## 9. Bibliographie

- [**BEN92**] L.P.M. BENDERS, M.P.J. STEVENS "Petri Net modelling in embedded system design" CompEuro 92, Computer systems and software engineering, pp. 612-17.
- [**BER82**] G. BERTHELOT "Petri nets theory for correctness of protocols" IEEE Trans. on Communications, Vol. com-30, N° 12, dec. 1982.
- [**BER83**] G. BERTHELOT "Vérification des Réseaux de Petri" Thèse de doctorat de troisième cycle, université P. et M. Curie (PARIS IV), 12 janv. 1978.
- [**BES88**] E. BEST, C. FERNANDEZ "Nonsequential processes" EATCS Monographs on Theoretical Computer Science, W. BRAUER, G. ROZENBERG, A. SALOMAA, Springer-Verlag, Vol. 13, 1988.
- [**BRA83**] G. W. BRAMS "Réseaux de Petri: théorie et pratique" T1 et T2 Masson 1983.
- [**BRA93**] J. G. BRAKER "An extended algorithm for performance evaluation of timed event graph" ECC-93 Groningen.
- [**BRE87**] J. BREMONT, A. GOLIOT, M. LAMOTTE, J. RAGOT "Identification d'un système de commande séquentiel par les réseaux de Petri" R.A.I.R.O. APII, Vol. 21, N° 3, pp. 273-91, 1987.
- [**CAR90**] J. CARDOSO "Sur les réseaux de Petri avec marquages flous" Thèse de Doctorat de l'université Paul SABATIER, 23 oct. 1990.
- [**CHO87**] C. CHOU "Performance evaluation of concurrent programs modeled by timed PQ-net" Proceedings of COMPSAC 87 (Computer software and applications conference) Tokyo pp. 465-73.
- [**CHR84**] P. CHRETIENNE "Controlled execution of timed Petri nets" Technology and Science of Informatics Vol. 3, N° 1, pp. 19-26, 1984.

- [COM88] M. A. COMEAU, K. THULASIRAMAN "Structure of submarking-reachability problem and network programming" IEEE Transactions on circuits and systems, Vol. 35, N° 1, pp. 89-100, Janv. 1988.
- [DAR86] M. DAROUACH, "Observabilité et validation de données de systèmes de grande dimension. Application à l'équilibrage des bilans de mesures", Thèse d'état, Université de Nancy I, Nancy, juin 1986.
- [DAV87] R. DAVID, H. ALLA "Continuous PNs" 8<sup>th</sup> European Workshop on Application and Theory on PNs zaragoza (spain) june 1987 pp 275-294.
- [DAV89] R. DAVID, H. ALLA "Du Grafset aux Réseaux de Petri" HERMES 1989.
- [DAV90] R. DAVID, H. ALLA "Properties of continuous models of transfer lines with unreliable machines and finites buffers" IMA Journal of Mathematics Applied Business & Industry, N° 6, pp. 281-308, 1990.
- [DAV94] R. DAVID, H. ALLA "Petri Nets modeling of dynamic systems, a survey" Automatica, Vol. 30, N° 2, pp. 175-202, 1994.
- [DIF93] A. DI FEBBRARO, R. MINCIARDI "Definition and performance evaluation using timed event graph" ECC-93 Groningen.
- [DUB93] E. DUBOIS, H. ALLA "hybrid Petri Net with stochastic discrete part" ECC-93 Groningen.
- [DUF84] J. DUFAU "Un outil pour la vérification des protocoles décrits par Réseaux de Petri" Thèse de docteur-ingénieur, université P. Sabatier (Toulouse), 27 janv. 1984.
- [EZP91] J. EZPELETA., J. MARTINEZ "Petri nets as a specification language for manufacturing systems" IMACS 13th World Congress and IMACS conference on Modelling and Control of Technological Systems juillet 1991-mai 1991, Dublin-Lille.

- [GAR85] K. GARG "An approach to performance specification of communication protocols using timed Petri Nets" IEEE Trans. on Software Engineering Vol. SE11, N° 10, pp. 1216-1224, oct 1985.
- [JEN81] K. JENSEN "Coloured Petri nets and the invariant-method" Theoretical Computer Science, Vol. 14, N° 3, pp. 317-336, jun.1981.
- [HOL87] M. A. HOLLIDAY, M. K. VERNON "A generalized timed net model for performance analysis" IEEE Transactions on software engineering, Vol. SE-13, N° 12, pp. 1297-1310, Dec. 1987.
- [KUM84] S. KUMAGAI, S. KODAMA, M. KITAGAWA "Structure of submarking-reachability problem and network programming" IEEE Transactions on circuits and systems, Vol. CAS-31, N° 2, pp. 159-64, Fev. 1984.
- [LAG93] Y. LAGOUE, F. DESBRANDES "Modélisation et simulation de réseaux de télécommunications" Revue Technique Thomson CSF, Vol. 25, N° 2, juin 1993.
- [LAM85] G. LAMARCHE, P. TAILLIBERT "Utilisation des RdP pour le test des programmes temps réel" Technique et Science Informatiques, Vol. 4, N° 1, pp. 83-87, 1985.
- [LEB91] J. LE BAIL, R. DAVID, H. ALLA "Hybrid Petri Nets" European Conference, Grenoble , july 1991, pp. 1472-77.
- [LEB92] J. LE BAIL, R. DAVID, H. ALLA "RdP Hybrides" TSI vol. 11, N° 5 1992, pp 95-120.
- [LEB93] J. LE BAIL, H. ALLA, R. DAVID "Asymptotic continuous Petri Nets" Discrete Event Dynamic Systems: Theory and Applications 2, pp.35-63, 1993.
- [LEC84] "Petri nets: central properties" Lecture Notes in Computer Science, G. GOOS, J. HARTMANIS, Springer-Verlag, Proceedings of an advanced course Bad Honnef, dec. 1984, 188.

- [**LEC86**] "Advances in Petri nets" Lecture Notes in Computer Science, G. GOOS, J. HARTMANIS, Springer-Verlag, Proceedings of an advanced course Bad Honnef, sept 1986, part. I, 254.
- [**LEC87**] "Petri nets: Applications and relationships to other models of concurrency" Lecture Notes in Computer Science, G. GOOS, J. HARTMANIS, Springer-Verlag, Proceedings of an advanced course Bad Honnef, sept. 1986, part. II, 255.
- [**MOR92**] S. MORIGUCHI "Diagnostic of computer systems by stochastic Petri Nets" IEEE International Symposium on Circuits and Systems, pp. 407-10, 10-13 mai 1992, San Diego.
- [**MUR77**] T. MURATA "State equation, controllability and maximal matchings of Petri nets" IEEE Transactions on Automatic Control, pp. 412-16, jun. 1977.
- [**MUR89**] T. MURATA "Petri Nets: properties, analysis and applications" Proceedings of IEEE Vol. 77, N° 4, pp. 571-580, april 1989.
- [**NOY90**] D. NOYES "Les réseaux de Petri et la modélisation des systèmes de production" R.A.I.R.O. APII, N° 24, pp. 511-28, 1990.
- [**PAS92**] K. PASSINO, A. N. MICHEL "Stability and boundedness analysis of DES" ACC-92/FM7 Chicago.
- [**PET62**] C.A. PETRI, "Kommunikation mit Automaten" Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 3, 1962. Traduction anglaise, "Communication with Automata" New York:Griffiss Air Force Base. Tech. Rep. RADCTR-65-377, Vol. 1, Suppl. 1, 1966.
- [**RAM73**] C. RAMCHANDANI "Analysis of asynchronous concurrent systems by timed Petri nets" PhD Thesis MIT, sept. 1973.
- [**RAM87**] P. J. RAMADGE, W. M. WONAHAM "Supervisor control of a class of discrete-event processes" SIAM J. Contr. Optimiz., Vol. 25, pp. 206-30, janv. 1987.



- [RAM89] P. J. RAMADGE "Some tractable Supervisory control problems for discrete-event systems modeled by buchi automata" IEEE Trans. on Automatic Control, Vol. 34, N° 1, Jan. 1989.
- [REI82] W. REISIG "Petri nets" EATCS Monographs on Theoretical Computer Science, W. BRAUER, G. ROZENBERG, A. SALOMAA, Springer-Verlag, Vol. 4, 1982.
- [RIC83] M. RICHARD "Réseaux de Petri grafcet et automates" cours INSA de Lyon, juin 1983.
- [ROU92] G. ROUX, B. DESCOTES-GENON, P. LADET "Operations cheking in flexible manufacturing systems" Robotics and Flexible Manufacturing Systems, IMACS 13th World congress, pp. 305-14, july 1991.
- [SHI90] Y. SHIEH, D. GHOSAL, P. R. CHINTAMANENI, S. K. TRIPATHI "Modeling of hierarchical distributed systems with fault tolerance" IEEE Trans. on Software Engineering, Vol. 10, N° 4, april 1990.
- [SIF77] J. SIFAKIS "Use of Petri nets for performance evaluation" Measuring, Modelling and Evaluation Computer Systems, H. Beilner and E. Gelenbe, (eds), pp.75-93, 1977.
- [THE78] S. THELLIEZ "Pratique séquentielle et réseaux de Petri" Editions Eyrolles Paris 1978.
- [TIT79] A. TITLI et al. "Analyse et commande des systèmes complexes" CEPADUES Editions 1979.
- [VAL80] R. VALETTE, M. COURVOISER "Systèmes de commande en temps réel" Editions SMC, 1980.

## CHAPITRE II

### DU CALCUL DES INVARIANTS

## 1. Introduction

Au cours du chapitre précédent, nous avons introduit les propriétés d'invariants de marquage et de franchissement sans toutefois aborder les moyens de résoudre ce type de problème. Aussi ce chapitre est-il entièrement dédié à cette notion que nous jugeons particulièrement riche. L'invariant de marquage permet de retrouver des ensembles de places dont le nombre global de marques n'est pas modifié par le tir de transitions. Cette propriété est donc étroitement liée à la structure du graphe (graphe marqué, graphe d'événements,...) puisque cette notion reste vraie quel que soit le marquage et quel que soit la séquence de franchissement [MUR89] [DAV89] [BRA83]. Elle est donc uniquement due aux arcs de connexion du réseau et à leur valuation. La connaissance de tels ensembles apporte une aide à l'analyse de la vivacité du RdP, ainsi qu'à la recherche d'autres propriétés telles que l'appartenance à un cycle, l'existence de marquages reproductibles ou de places bornées, etc....

La propriété d'invariant étant liée à la structure du RdP, elle permet de vérifier la validité du modèle, de s'assurer qu'il répond bien au cahier des charges et qu'il effectue toutes les fonctions pour lesquelles il a été conçu [BER78] [BER79]. Un ensemble de places liées par une relation linéaire constitue souvent un bloc fonctionnel dont un exemple simple est constitué par l'état d'un interrupteur (marche/arrêt). Le bloc modélisant les états d'un interrupteur est constitué de deux places symbolisant les états et deux transitions permettant le passage d'un état (marche/arrêt) à l'autre (arrêt/marche). La présence d'un jeton dans l'une des deux places indique l'état actuel de l'interrupteur. La somme des jetons des places marche et arrêt vaut toujours 1 puisqu'un seul de ces états est actif à un instant donné, ceci constitue un invariant quel que soit l'évolution du système.

Les invariants de franchissement sont des séquences de tir permettant au RdP de revenir à l'état initial. Aussi l'existence de telles séquences constitue la preuve pour un système itératif qu'il est capable de se réinitialiser. Les processus industriels ont souvent un comportement répétitif (chaînes d'usinage, d'assemblage ...), cette propriété est donc essentielle à vérifier.

Au cours du paragraphe suivant, nous montrons que nous pouvons réduire la recherche des invariants, c'est-à-dire T-semi-flots et P-semi-flots, à la résolution, dans  $\mathbb{N}$ , d'un système d'équations linéaires homogènes ( $Ax = 0$ ). Dans un premier temps, nous présentons deux méthodes algébriques, une méthode couramment utilisée dans la littérature pour laquelle nous proposons des améliorations puis une méthode plus rigoureuse d'où découle la première. Dans un deuxième temps, nous développons une nouvelle méthode originale et rapide, particulièrement adaptée aux grands réseaux, résultat que nous appliquons à des systèmes particuliers (RdP de grande dimension). Enfin, nous concluons par une présentation des propriétés déduites à l'aide des invariants, sur un exemple du chapitre précédent.

## 2. Position du problème

### 2.1. P-semi-flots

L'invariant de marquage consiste à trouver des relations linéaires entre les marquages des différentes places d'un RdP, ce qui revient à calculer des poids  $f_1, f_2, \dots, f_n$  tels que:

$$f_1 M(P_1) + f_2 M(P_2) + \dots + f_n M(P_n) = K$$

où  $F = [f_1, f_2, \dots, f_n]$  et  $K$  une constante quelconque.

on a donc,  $\forall M_i \in A(R, M_0) \quad F^T M_i = K,$

et, en particulier pour  $M_0,$

$$F^T M = F^T M_0 \tag{II.2.1}$$

Or la relation fondamentale:

$$M = M_0 + C\sigma$$

multipliée à gauche par  $F^T$ , devient:

$$F^T M = F^T M_0 + F^T C \sigma$$

la séquence de franchissement  $\sigma$  étant quelconque, de la relation [II.2.1] on peut déduire:

$$F^T C = 0$$

si et seulement si:

$$C^T F = 0$$

$C$  est une matrice d'incidence,  $n$  lignes et  $m$  colonnes à éléments dans  $\mathbb{Z}$  et  $F$  un vecteur à  $n$  éléments dans  $\mathbb{N}$  (voire  $\mathbb{Z}$  ou  $\mathbb{Q}$ ). Il faut résoudre dans  $\mathbb{N}$  un système de  $m$  équations à  $n$  inconnues.

## 2.2. T-semi-flots

Le calcul de T-semi-flots revient à chercher des vecteurs  $\sigma$  tels que le RdP puisse se réinitialiser. Ceci est équivalent à trouver des séquences de franchissement qui partant d'un marquage initial permettent d'y revenir (soit  $M_0(\sigma) > M_0$ ). Cette propriété est importante pour les systèmes réalisant un traitement itératif (bouclage).

L'évolution du marquage d'un RdP est décrite par l'équation:

$$M_0(\sigma) > M \quad \Leftrightarrow \quad M = M_0 + C \sigma \quad \text{[II.2.2]}$$

Le franchissement de la séquence  $\sigma$  permet le retour à l'état initial si:

$$M = M_0$$

d'après [II.2.2], on doit avoir:

$$C \sigma = 0$$

où  $C$  est une matrice d'incidence,  $n$  lignes et  $m$  colonnes à éléments dans  $\mathbb{Z}$  et  $\sigma$  un vecteur à  $m$  éléments dans  $\mathbb{N}$ . Il faut résoudre dans  $\mathbb{N}$  un système de  $n$  équations à  $m$  inconnues.

### 2.3. Remarques

1/ Le calcul des invariants, qu'ils soient de type T-semi-flots ou P-semi-flots, peut être ramené à un *seul et même problème*: la recherche de solutions dans  $\mathbb{N}$  à un système d'équations linéaires homogènes:

-  $C$  (matrice d'incidence du RdP) pour les T-semi-flots:

$$Cs = 0, \quad \text{avec } C \in \mathbb{Z}^{n \times m}, s \in \mathbb{N}^m, n = |P| \text{ et } m = |T| \quad [\text{II.2.3}]$$

-  $C^T$  (transposée de la matrice d'incidence du RdP) pour les P-semi-flots.

$$C^T y = 0, \quad \text{avec } C^T \in \mathbb{Z}^{m \times n}, y \in \mathbb{N}^n, n = |P| \text{ et } m = |T|$$

2/ Certains auteurs font la recherche des P-semi-flots dans  $\mathbb{Z}$  voire dans  $\mathbb{Q}$  ce qui est logique puisque les solutions ne sont que des poids affectés aux places du RdP, il peut être avantageux d'étendre l'espace d'investigation des solutions notamment pour disposer de plus de relations pour le diagnostic. Par contre pour le calcul des T-semi-flots, les solutions du système [II.2.2] étant des séquences de franchissement, elles ne peuvent être qu'entières.

3/ Dans ce chapitre, les RdP sont souvent représentés par leur matrice d'incidence.

4/ Nous entamerons cette analyse par le rappel de théorèmes de base pour la résolution des systèmes homogènes, à savoir des systèmes du type:

$$C.x = 0 \quad \text{avec } C \in \mathbb{Z}^{n \times m} \quad [\text{II.2.4}]$$

Les théorèmes suivants décrivent les conditions d'existence d'une solution  $x$  dans  $\mathbb{Z}^m$  [LAN76] [MIN83] [BAR84] [LAN85] [DUF89] [HOR90]:

**Théorème 2.3.1:**

Une condition nécessaire et suffisante pour que [II.2.4] possède une solution autre que la solution triviale est que le rang  $r$  de  $C$  soit strictement inférieur à  $m$ . •

**Théorème 2.3.2:**

Une condition nécessaire et suffisante pour qu'un système à  $n$  équations homogènes et  $n$  inconnues ait une solution autre que la solution triviale est  $\text{Det}(C) = 0$ . •

**Théorème 2.3.3:**

Si le rang  $r$  de  $C$  est inférieur à  $m$ , le système a exactement  $m-r$  solutions linéairement indépendantes et toute solution est une combinaison linéaire de ces  $m-r$  solutions. De plus toute combinaison linéaire de solutions est encore solution. •

Cependant, dans la section 4 de ce chapitre, vu la nécessité de calculer les solutions dans  $\mathbb{N}$ , nous montrerons les restrictions engendrées sur l'espace des solutions.

### 3. Etat de l'art

#### 3.1. Première approche

L'algorithme suivant, initié par [MAR81] puis repris et amélioré par de nombreux auteurs, notamment [ALA81], [ALA85], [SIL88], [TAN91], [TOU82] ou [MAR91], consiste à effectuer des transformations régulières sur la matrice d'incidence  $C$ , afin d'annuler successivement ses colonnes. Ces transformations appliquées parallèlement à une matrice identité fournissent une base génératrice de l'espace des solutions du système [II.2.3]. Soit une matrice d'incidence  $C \in \mathbb{Z}^{n \times m}$  où  $C(i,j)$  est l'élément de la  $i^{\text{ème}}$  ligne et de la  $j^{\text{ème}}$  colonne:

**Algorithme 1:**

$B \leftarrow I(m \times m)$ ;  $I(m \times m)$  est la matrice identité à  $m$  lignes et  $m$  colonnes

**Tant que** ( $C$  a au moins une colonne) **Faire**

**M<sub>1</sub>**- Déterminer les ensembles  $I$  et  $J$  tels que  $I = \{ i / C(i,1) > 0 \}$   
 et  $J = \{ j / C(j,1) < 0 \}$

**M<sub>2</sub>**- **Pour** (tout couple  $(i,j) \in I \times J$ ) **Faire**

**M<sub>21</sub>** Ajouter après la dernière ligne de  $C$  le vecteur ligne  
 $C(j,1).(i^{\text{ième}} \text{ ligne de } C) - C(i,1).(j^{\text{ième}} \text{ ligne de } C)$

**M<sub>22</sub>** Ajouter après la dernière ligne de  $B$  le vecteur ligne  
 $C(j,1).(i^{\text{ième}} \text{ ligne de } B) - C(i,1).(j^{\text{ième}} \text{ ligne de } B)$

**FinPour**

**M<sub>3</sub>**- Supprimer de  $C$  et  $B$  les lignes d'indice  $i \in I \cup J$

**M<sub>4</sub>**- Supprimer de  $C$  la première colonne

**M<sub>5</sub>**- Supprimer de  $B$  toutes les lignes à support non minimal (c.à.d.  
 combinaison linéaire de lignes de  $B$ )

**FinTq**

**M<sub>6</sub>**- Alors les lignes de  $B$  forment une famille génératrice minimale des semi-flots  
 de  $C$

Cet algorithme donne la solution dans  $\mathbb{N}^m$ , mais présente deux inconvénients majeurs:

- un risque d'évolution exponentielle en temps et en espace du fait de l'ajout de lignes (voir module **M<sub>2</sub>** de l'algorithme 1). Ceci est d'autant plus vrai que le système est de grande dimension.

- il ne peut être appliqué à des matrices quelconques. En effet, le calcul des solutions du système homogène  $C.x=0$ , est basé sur l'élimination des éléments de la première colonne de la matrice  $C$ . Pour cela, les colonnes de  $C$  doivent contenir des éléments positifs et des éléments négatifs. C'est le cas des matrices d'incidence des RdP si toute transition a au moins une place en entrée et une place en sortie, néanmoins, les transformations successives peuvent mener à un blocage.



Cependant une heuristique permet d'améliorer cette méthode en choisissant comme colonne pour construire I et J (voir module  $\mathbf{M}_1$  de l'algorithme 1), celle qui minimise l'accroissement du nombre de lignes de C [ALA85], à savoir:

$$|I| \cdot |J| - |I| - |J|$$

Une seconde heuristique qui diminue sensiblement le temps de calcul, consiste à appliquer l'algorithme précédent non pas directement sur C, mais sur une matrice  $C_1$  obtenue en supprimant de C toutes les lignes qui sont combinaisons linéaires d'autres lignes de C. Si de telles lignes existent, elles n'apportent rien à la résolution du système. On a alors :

$$\begin{array}{l} \text{Si } \text{rang}(C) = n_1 \quad \text{alors } \text{rang}(C_1) = n_1 \\ \text{avec } C(n \times m), C_1(n_1 \times m) \text{ et } n_1 \leq n. \end{array}$$

### 3.2. Deuxième approche

Soient :

C une matrice  $n \times m$  et b un vecteur à coefficients entiers (c.à.d. dans un anneau principal  $\mathbb{Z}^m$ )

$SL_m(\mathbb{Z}^m)$  le groupe des matrices  $m \times m$  à coefficients dans  $\mathbb{Z}^m$  de déterminant 1.

La résolution du système :

$$C \cdot x = b$$

se ramène au calcul de l'image et du noyau de la matrice C [NOD92]. Pour cela on construit une base de  $\text{Im}(C)$ , sous-module de  $\mathbb{Z}^n$  engendré par les m colonnes de C. L'idée est de remplacer la matrice  $C(n \times m)$  par une matrice C' équivalente à droite, c'est-à-dire pour laquelle :

$$\exists R \in SL_m(\mathbb{Z}^m) \quad \text{tel que} \quad C' = C.R$$

les matrices  $C$  et  $C'$  ont alors même image. Pour calculer l'image de  $C$ , on utilise la classe des matrices échelonnées en colonnes.

**Définition 3.2.1:**

Une matrice échelonnée,  $C$ , est une matrice telle que si  $C_h$  est le vecteur de la colonne  $h$ , alors  $h(C_1) > h(C_2) > h(C_3) \dots > h(C_k) > 0$  et  $C_{k+1} = C_{k+2} = C_{k+3} = \dots = C_m = 0$ .

La fonction  $h(C_h)$ , hauteur du vecteur  $C_h$  de  $\mathbb{Z}^n$ , est l'entier  $n-i+1$  où  $i$  est le plus grand indice tel que  $c_j = 0$  pour  $1 < j < i$ , ce qui peut être schématisé par:

$$C_h^T = (0, 0, \dots, 0, c_{i+1}, c_{i+2}, \dots, c_n) \quad \text{avec} \quad c_{i+1} \neq 0 \text{ et } h(C_h) = n-i+1.$$

**Exemple 3.2.1:**

Une matrice  $C$  échelonnée: 
$$C = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 & 0 & 0 \\ 3 & 1 & 2 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 3 & 0 & 0 \end{pmatrix}$$

avec  $h(C_1)=5, h(C_2)=4, h(C_3)=3, h(C_4)=1, h(C_5)=h(C_6)=0$ .

**Proposition 3.2.1:**

Soit  $C'$  une matrice échelonnée sur un anneau intègre ( $\mathbb{Z}$ ).

1/ Les colonnes non nulles de la matrice échelonnée  $C'$  forment une base de l'espace engendré par les colonnes.

2/ Les vecteurs  $(e_i)_{i>k}$  correspondant aux colonnes nulles  $(C'_i)_{i>k}$  de  $C'$  forment une base du noyau de la matrice  $C'$ .

**Théorème 3.2.1:**

1/ Pour toute matrice  $C(n \times m)$  à coefficients dans  $\mathbb{Z}$ , il existe une matrice  $R \in SL_m(\mathbb{Z}^m)$  permettant la transformation de  $C$ , en une matrice échelonnée  $C'$ , telle que :

$$C' = C.R \quad \text{[II.3.1]}$$

Si la matrice de passage à droite  $R$  appartient à  $SL_m(\mathbb{Z}^m)$  et vérifie  $\text{Det}(R) = 1$ , alors la matrice  $C'$  est dite sl-équivalente à droite à  $C$ .

2/ Le résultat subsiste si l'anneau des entiers est remplacé par un anneau principal où l'on dispose d'un algorithme permettant le calcul des coefficients de Bezout. •

Explicitons un algorithme réalisant la transformation de  $C$  en une matrice échelonnée  $C'$ , ainsi que la création de la matrice de passage à droite  $R$ . D'après la proposition précédente, les colonnes de  $R$  correspondant aux colonnes nulles de  $C'$  forment une base du noyau de  $C$ . L'échelonnement consiste, à chaque étape, à annuler les coefficients  $k > j$  de la ligne  $i$ , comme ci-dessous:

$$\begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 \\ \cdot & a_{32} & 0 & 0 & 0 & 0 \\ \cdot & \cdot & * & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \leftarrow i$$

La matrice de passage  $R$  est obtenue en l'initialisant à la matrice identité, puis en lui appliquant les mêmes combinaisons linéaires élémentaires à droite, que celles effectuées sur  $C$  pour obtenir  $C'$ .

**Algorithme 2:**

```

R ← I(mxm);      I(mxm) est la matrice identité à m lignes et m colonnes
C' ← C;
j ← 1;
Pour (i variant de 1 à n) Faire
    Si (j == m) Alors
        quitter;
    Finsi
    Pour (k variant de j+1 à m) Faire

```

$$\text{Calculer } \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} \text{ vérifiant } \begin{cases} \gamma C'_{ij} + \delta C'_{ik} = 0 \\ \text{et} \\ \alpha\delta - \beta\gamma = 1 \end{cases}$$

$$(C'_j, C'_k) \leftarrow (\alpha C'_j + \beta C'_k, \gamma C'_j + \delta C'_k);$$

$$(R_j, R_k) \leftarrow (\alpha R_j + \beta R_k, \gamma R_j + \delta R_k);$$

**FinPour****Si**  $(C'(i,j) \neq 0)$  **Alors** $j \leftarrow j+1;$ **Finsi****FinPour**

Afficher(C',R);

$\alpha, \beta, \delta$  et  $\gamma$  sont les coefficients de Bezout [NOD92].

La relation [II.3.1] est vraie au début de l'algorithme et reste invariante à chaque étape (puisque l'on multiplie les deux termes de l'égalité par la même quantité). Elle est donc toujours vraie à la fin. Soit  $I$  l'ensemble des indices des colonnes nulles de  $C'$ , alors  $|I|$  donne la dimension de l'espace des solutions. De plus les vecteurs  $R_i$  de  $R$  ( $i \in I$ ) forment une famille génératrice minimale des solutions du système autonome [II.2.4].

#### 4. Méthode proposée [BOU93]

Dans les méthodes que nous avons exposées précédemment et qui finalement consistent à transformer la matrice d'incidence  $C$ , de manière à faire apparaître des colonnes nulles, le temps de calcul varie énormément d'un système à l'autre, selon que  $C$  est plus ou moins creuse. Aussi, nous présentons un algorithme plus rapide que les précédents mais aussi, facilement implémentable. Ce dernier permet de calculer une famille génératrice minimale des invariants du système. Cette approche consiste à transformer le système initial à  $n$  équations et  $m$  inconnues, en une forme équivalente réduite et donc plus simple à résoudre.

La matrice  $C$  étant quelconque, nous allons la partitionner en blocs munis de propriétés particulières. En effet, les lignes de  $C$  n'étant pas forcément indépendantes, certaines d'entre elles sont combinaisons linéaires d'autres lignes, elles sont donc porteuses d'information redondante, inutile à la résolution du système [II.2.4].

Soit :

$$C = \begin{pmatrix} C_1 & C_2 \\ C_3 & C_4 \end{pmatrix} \quad \text{où } C_1(m_1 \times m_1), C_2(m_1 \times m_2) \text{ et } m_1 + m_2 = m \quad [\text{II.4.1}]$$

avec  $\text{rang}(C) = \text{rang}(C_1)$

Pour obtenir ces blocs, il suffit de prendre une nouvelle matrice, de l'initialiser avec la première ligne de  $C$ , puis de rajouter les lignes de  $C$  qui augmentent le rang. En effectuant la même opération sur les colonnes de cette nouvelle matrice, on obtient la décomposition  $(C_1 \ C_2)$ . Le bloc  $(C_3 \ C_4)$  n'est pas pris en compte puisqu'inutile à la résolution du système:

$$\text{car } \forall i, (C_{3i} \ C_{4i}) = \sum_j b_{ij} (C_{1j} \ C_{2j})$$

On se ramène donc au système réduit:

$$(C_1 \ C_2) \cdot x = 0 \quad [\text{II.4.2}]$$

$x \in \mathbb{N}^m$  peut être partitionné selon  $(C_1 \ C_2)$  en  $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

avec,  $x_1 \in \mathbb{N}^{m_1}$  et  $x_2 \in \mathbb{N}^{m_2}$

$C_1$  étant une matrice carrée de rang plein, elle est donc inversible. De l'équation [II.4.2] on déduit :

$$x_1 = M.x_2 \quad \text{avec} \quad M = -(C_1)^{-1}.C_2 \quad \text{ou encore} \quad M = \frac{-1}{\text{Det}(C_1)}.(\text{com}(C_1))^T.C_2 \quad [\text{II.4.3}]$$

$\text{com}(C_1)$  est la matrice des cofacteurs de  $C_1 \in \mathbb{Z}^{m_1 \times m_2}$

Cette expression montre qu'il suffit de fixer  $x_2$  pour obtenir  $x_1$ . Toutefois,  $C_1$  étant une matrice à éléments dans  $\mathbb{Z}$ ,  $\text{com}(C_1)$  est elle aussi à éléments entiers. Donc, si  $|\text{Det}(C_1)| \neq 1$ , les éléments de  $M$  seront rationnels et les solutions risquent elles aussi d'être rationnelles. La génération des solutions dépendra donc de la matrice  $M$  et de l'espace dans lequel on calcule les solutions.

#### Solutions dans $\mathbb{Q}$ :

D'après l'expression [II.4.3], les solutions sont immédiates. Il suffit de prendre pour  $x_2$  les éléments de la base canonique de dimension  $m_2$  ou encore la matrice identité  $I$  alors  $x_1 = M.I$ . Une base génératrice des solutions est donnée par les vecteurs colonnes de la matrice  $S$  suivante:

$$S = \begin{bmatrix} M \\ I \end{bmatrix} \quad \text{où} \quad M = \frac{-1}{\text{Det}(C_1)}.(\text{com}(C_1))^T.C_2 \quad \text{et } I \text{ est la matrice identité.}$$

#### Solutions dans $\mathbb{Z}$ :

Comme  $M$  peut avoir des éléments rationnels si  $|\text{Det}(C_1)| \neq 1$ , on choisit  $x_2 = \text{Det}(C_1).I$  pour assurer que les éléments de  $M$ ,  $x_2$  et  $x_1$  seront entiers. Une base génératrice des solutions est donnée par les vecteurs colonnes de la matrice  $S$  suivante:

$$S = \begin{bmatrix} kM \\ kI \end{bmatrix} \quad \text{où} \quad k = \text{Det}(C_1), \quad M = \frac{-1}{\text{Det}(C_1)}.(\text{com}(C_1))^T.C_2 \quad \text{et } I \text{ est la matrice}$$

identité.

**Solutions dans  $\mathbb{N}$ :**

Les règles qui suivent permettent le calcul des solutions dans  $\mathbb{N}$ :

**Règle 1:**

1/ Si  $M$  a une ligne à coefficients tous strictement négatifs alors il n'existe pas d'autre solution que zéro. •

2/ Si  $M$  a une ligne à coefficients tous négatifs ou nuls, la dimension de l'espace des solutions est  $m_2 - s$  où  $s$  est le nombre d'éléments strictement négatifs de cette ligne.

3/ Enfin, si  $M$  a plusieurs lignes à coefficients tous négatifs ou nuls, alors créer un vecteur  $v$  contenant la somme de toutes ces lignes. La dimension de l'espace des solutions est  $m_2 - s$  où  $s$  est le nombre d'éléments strictement négatifs du vecteur  $v$ . •

**Preuve:**

Soit  $x_{1i}$  la  $i^{\text{ème}}$  ligne de  $x_1$ , alors on a :

1er cas

$$x_{1i} = \sum_{j=1}^{m_2} M_{ij} \cdot x_{2j} \text{ et } x_{2i} \geq 0$$

si  $M_{ij} < 0 \quad \forall 1 \leq j \leq m_2$

alors  $x_{1i} \leq 0$ , cette solution est absurde puisque  $x \in \mathbb{N}^m$ .

2ième cas

pour générer les  $m_2$  solutions on choisit  $x_2$  tel que  $\text{rang}(x_2) = m_2$

si  $\exists k$  tel que  $M_{ik} < 0$  et  $M_{ij} \leq 0$  pour  $1 \leq j \leq m_2$  et  $j \neq k$

alors pour satisfaire  $x \in \mathbb{N}^m$  on doit choisir  $x_{2k} = 0$  (d'après le 1er cas)

et donc  $\text{rang}(x_2) = m_2 - 1$ .

S'il existe  $s$  éléments tels que  $M_{ik} < 0$  et  $M_{ij} \leq 0$  pour  $1 \leq j \leq m_2$  et  $j \neq k$  alors  $\text{rang}(x_2) = m_2 - s$  car  $x_2$  aura  $s$  lignes nulles.

**Règle 2:** Si  $M$  a au moins une colonne  $j$  à coefficients tous strictement positifs alors le système [I.3.1] a toujours des solutions dans l'espace de dimension  $m_2$ . •

**Preuve:**

En effet si les coefficients de la colonne  $j$  sont tous positifs on peut toujours choisir  $x_{2j}$  assez grand pour compenser le poids des autres coefficients (pouvant être négatifs) d'une même ligne, et donc:

$$x_{1j} = x_{2j} \cdot M_{ij} + \sum_{k=1}^{j-1} x_{2k} \cdot M_{ik} + \sum_{k=j+1}^{m_2} x_{2k} \cdot M_{ik} \geq 0 \text{ pour } 1 \leq i \leq m_1$$

il suffit de prendre  $x_{2j} = \lceil \text{Min}(M_{kh}) \rceil$ ,  $1 \leq k \leq m_1$  et  $1 \leq h \leq m_2$ , si  $M$  contient au moins un élément négatif sinon  $x_2 = I$  ( $I$  étant la matrice identité  $m_2 \times m_2$ ).

Pour obtenir une famille génératrice minimale des solutions, il suffit de prendre pour  $x_2$  les vecteurs de la base canonique ( $e_i$  de dimension  $m_2$  où le terme d'indice  $j$  est fixé à  $x_{2j} = a = \lceil \text{Min}(M_{kh}) \rceil$  soit:

$$\begin{pmatrix} 1 \\ a \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ a \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ a \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ a \\ 0 \\ 1 \end{pmatrix} \leftarrow \text{indice } j$$

Comme on choisit la base canonique, toute autre solution, ne peut être que combinaisons linéaires des solutions ainsi générées. Une astuce de programmation consiste à prendre pour  $x_2$  la matrice identité où la ligne  $j$  est forcée à la valeur  $\lceil \text{Min}(M_{kh}) \rceil$ :

Une base génératrice des solutions est donnée par les vecteurs colonnes de la matrice  $S$  suivante:

$$S = \begin{bmatrix} RM \\ R \end{bmatrix} \quad \text{avec} \quad R = \begin{pmatrix} 1 & 0 & \dots & 0 \\ a & \dots & a & a \\ 0 & 0 & 1 & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$



**Règle 3:** Si les règles 1 et 2 ne sont pas vérifiées (M ne contient ni de ligne négative ni de colonne positive), une condition nécessaire mais non suffisante pour faire apparaître dans M une colonne à coefficients tous positifs est: chaque ligne de M doit contenir au moins un élément strictement positif. •

Pour cela, il suffit de transformer l'expression [II.4.3] comme suit:

$$\begin{aligned} x_1 &= M.D.D^{-1}.x_2 \\ \Leftrightarrow x_1 &= \bar{M}.\bar{x}_2 \end{aligned}$$

où  $\bar{M}$  est une matrice contenant une colonne positive et D est une matrice de la forme:

$$D = \begin{pmatrix} 1 & 0 & . & . & 0 & a_1 & 0 & . & . & 0 \\ 0 & 1 & . & . & 0 & a_2 & 0 & . & . & 0 \\ & & & & & \cdot & \cdot & & & \cdot \\ & & & & & \cdot & \cdot & & & 0 \\ 0 & 0 & . & . & 0 & a_{m_1} & 0 & . & 0 & 1 \end{pmatrix}$$

une telle matrice permet de substituer à une colonne de M une combinaison des autres colonnes, il suffit de choisir  $a_i > 0$  suffisamment grand pour générer une colonne positive.

Il est alors possible d'appliquer la règle 2 sachant que:

$$\bar{M} = M.D \text{ et } x_2 = D.\bar{x}_2.$$

**Règle 4:** Si la règle 1 n'est pas vérifiée (pas de ligne négative) et si on ne peut pas faire apparaître dans M une colonne à coefficients tous positifs, alors la méthode proposée ne permet pas de donner systématiquement toutes les solutions dans  $\mathbb{N}$ . •

Dans ce cas, on peut toujours transformer le système de la façon suivante:

$$x_1 = M x_2$$

$$\Leftrightarrow [M \quad -I] \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = 0$$

On obtient ainsi un nouveau système homogène beaucoup plus simple que le premier [II.2.4] car le premier bloc est réduit et le deuxième bloc est creux. La matrice  $[M \quad -I]$  se prête bien à la résolution par l'algorithme 1 notamment parce qu'elle doit contenir des colonnes contenant des éléments positifs et des éléments négatifs. C'est donc l'algorithme 1 qui sera retenu pour la résolution de ce système (les résultats de simulations confirment ce choix). La résolution d'un système d'équations homogènes par cette méthode est plus rapide que les deux autres méthodes si  $M$  vérifie la règle 2. Dans le cas contraire, on peut attendre des performances similaires.

**Remarque 4.1:**

Comme au cours des différentes simplifications (notamment [II.4.1]) nous effectuons des permutations de colonnes (termes multipliés par les  $x_i$ ), il est nécessaire de les mémoriser afin de réarranger les vecteurs solutions  $x$  en conséquence. •

Si les invariants obtenus ne sont pas minimaux (c'est le cas si le choix de  $x_2$  n'est pas  $I$ ,  $M$  ayant des coefficients négatifs), on peut simplement les obtenir par combinaison linéaire avec les autres éléments de la base génératrice.

**Algorithme 3:**

**M1.** Arranger  $C$  sous la forme [4.1]

et Mémoriser les permutations de colonnes

**M2.**  $M = |\text{Det}(C_1)| \cdot (C_1)^{-1} \cdot C_2$

**M3.** Si ( $M$  a une ligne  $< 0$ ) **Alors**

le système [II.2.4] n'a pas d'autre solution que 0

**Sinon**

trouver une colonne de  $M \geq 0$

générer les solutions

**Finsi**

**M4.** Réarranger  $x$  en fonction des permutations effectuées en **M1** (remarque 4.1)

**Exemple 4.1:**

Le RdP, dont la matrice d'incidence suit, modélise le protocole de communication de l'exemple 7.1 du chapitre I [GAR85]:

$$C = \begin{pmatrix} -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} \quad C \in \mathbb{Z}^{10 \times 9}$$

**M<sub>1</sub>** donne la décomposition:

$$C_1 = \begin{pmatrix} -1 & 0 & 1 & 1 & 1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & -1 \end{pmatrix}$$

**M<sub>2</sub>** calcule:

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \text{où } m_1 = 6 \text{ et } m_2 = 3, \text{ on a bien } m_1 + m_2 = m = 9$$

comme  $m_2 = 3$ , la base génératrice des solutions  $x$  sera constituée de 3 vecteurs (théorème 2.3.3).

$M_3$  génère les solutions qui réarrangées par  $M_4$  sont:

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{matrix} X_{11} \\ X_{12} \\ X_{13} \\ X_{14} \\ X_{15} \\ X_{16} \\ X_{21} \\ X_{22} \\ X_{23} \end{matrix} \left. \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \right\} \begin{matrix} X_1 \\ \\ \\ \\ X_2 \end{matrix}$$

Ces trois vecteurs forment bien une famille génératrice, d'ordre minimal, des solutions (T-semi-flots) du système [II.2.3]. Dans cet exemple le module  $M_4$  n'a pas de réarrangement à effectuer car il n'y a eu aucune permutation de colonne en  $M_1$ .

## 5. Discussion

Les quatre règles énoncées permettent d'envisager tous les cas de figure et d'être mathématiquement rigoureux. Notamment lorsque des solutions existent, connaissant la dimension de leur espace  $m_2 = m - \text{rang}(C)$ , et sachant trouver  $m_2$  solutions linéairement indépendantes, ces solutions constituent bien une base génératrice minimale de l'espace des solutions. De plus, dans la pratique cet algorithme testé sur des dizaines de matrices d'incidence a toujours permis d'obtenir une matrice  $M$  ayant une colonne positive. Nous avons réussi à isoler des cas pour lesquels cette colonne n'était pas apparente, mais en application de la règle 3, il a toujours été possible d'en faire apparaître une. Il semblerait donc que ce soit une propriété des RdP. Cependant, jusqu'à présent nous n'avons pu démontrer cet état de fait.

Dans l'algorithme 1, le nombre d'opérations est difficilement quantifiable. Il dépend du nombre d'éléments négatifs ( $|I|$ ) et positifs ( $|J|$ ) que compte la première colonne de la matrice d'incidence  $C$ . De plus, ce nombre évolue au cours du traitement, puisqu'on rajoute de nouvelles lignes (en  $M_{21}$  et  $M_{22}$ ). Cependant, en comparant les durées d'exécution des

deux types d'algorithmes, on obtient des temps sensiblement équivalents pour des systèmes de faibles dimensions.

Pour l'exemple 4.1:

Algorithme	Taille de C	Temps CPU
Algorithme 1	10x9	0.75
Algorithme 3	10x9	0.95

Tableau 5.1

Par contre dès que C est plus grande, l'algorithme 1 devient nettement plus lent, comme on peut le constater sur le tableau suivant:

Algorithme	Taille de C	Temps CPU
Algorithme 1	24x22	14.5
Algorithme 3	24x22	6.48
Algorithme 1	48x44	89.22
Algorithme 3	48x44	32.28

Tableau 5.2

### Remarques 5.1:

Ces temps d'exécution ne sont donnés qu'à titre comparatif (en secondes, sous Matlab et sur un Macintosh Centris 650).

Notons que l'algorithme 2 est plus rigoureux que le premier, il garantit un résultat d'ordre minimal et ne risque pas d'évolution exponentielle. Cependant, appliqué à des systèmes de grande dimension, il est moins rapide que l'algorithme 3. En effet, l'échelonnement de la matrice d'incidence C est obtenu en annulant les coefficients de la partie triangulaire supérieure de C (soit donc,  $\frac{n.m}{2}$  itérations pour une matrice à n lignes et m colonnes), traitement dont la durée dépend directement de la taille de C.

Le tableau suivant fourni un résumé des avantages et inconvénients des différentes méthodes, afin d'aider l'utilisateur dans son choix de la solution la plus adaptée à son problème:

<b>Algorithme 1</b>	<b>Algorithme 2</b>	<b>Algorithme 3</b>
donne uniquement les solutions dans $\mathbb{N}$	donne uniquement les solutions dans $\mathbb{Z}$	donne les solutions et la dimension de leur espace dans $\mathbb{Q}$ , $\mathbb{Z}$ et $\mathbb{N}$
rapide pour les systèmes de petite dimension	rapide pour les systèmes de petite dimension	rapide pour les systèmes de grande dimension
risque d'évolution exponentielle, ou de ne pas aboutir si pas de colonne à éléments $<0$ et $>0$ .	robuste	risque d'être lent si $M$ n'a pas de colonne à éléments tous positifs.

Tableau 5.3 Récapitulatif des propriétés

## 6. Cas des RdP de grande dimension [BOU94]

Lors de ce paragraphe, les résultats obtenus précédemment pour des RdP classiques sont étendus au cas des systèmes de grande dimension. Pour résoudre ce type de système il est possible d'appliquer les algorithmes précédents directement sur la matrice  $C$  et donc de résoudre le système global ( $n$  équations et  $m$  inconnues ou inversement), ce qui s'avère souvent très fastidieux (temps et robustesse des algorithmes) ou bien, d'exploiter une particularité des RdP, leur matrices d'incidence sont assez creuses. Notons, qu'il est toujours possible de transformer la matrice d'incidence du système à résoudre en une matrice composée de blocs diagonaux. Pour cela, deux cas de figure sont envisageables:

- le système de grande taille est constitué de sous-systèmes connus et apparents, interconnectés, auquel cas il suffit de numéroter correctement les places et les transitions du RdP pour avoir une matrice d'incidence de la forme suivante:

$$C = \begin{pmatrix} C'_1 & 0 & \dots & 0 \\ 0 & \cdot & 0 & \cdot \\ \cdot & 0 & \cdot & 0 \\ 0 & \dots & 0 & C'_n \end{pmatrix} \quad [\text{II.6.1}]$$

Mais cette forme diagonale parfaite est rarement possible du fait de la présence de liens résiduels (ou variables de couplage). En effet, comme les sous-systèmes communiquent entre eux, ils sont forcément reliés par des arcs et ont des places et/ou des transitions communes,

- le système de grande dimension est représenté par une matrice d'incidence dont les blocs diagonaux ne sont pas apparents. Il faut alors faire appel aux techniques de la littérature pour mettre  $C$  sous la forme [II.6.1]. Le paragraphe suivant, est dédié à la décomposition en blocs diagonaux.

### 6.1. Décomposition en blocs diagonaux

La décomposition en blocs diagonaux offre deux intérêts majeurs:

- la résolution de sous-systèmes d'ordre plus faible et plus simples,
- la possibilité de résoudre les sous-systèmes en parallèle, le temps de résolution global étant celui du plus grand sous-système.

Les matrices d'incidence des RdP de grande taille sont naturellement elles aussi de grande dimension, toutefois elles sont très creuses et se prêtent donc bien à une mise sous forme de blocs diagonaux. Ce problème est souvent appelé dans la littérature: **Technologie de Groupe** (T.G.) et correspond en fait à la décomposition d'un système (modélisé par un RdP) en plusieurs sous-systèmes de taille plus petite et plus simples à résoudre. Le principe d'une T.G. est de regrouper dans des blocs diagonaux, le maximum d'éléments non nuls de la matrice et de rejeter à l'extérieur des blocs les éléments nuls. Les éléments non nuls qui ne sont pas dans un bloc diagonal, sont appelés variables de couplage ou liens résiduels et sont dus au fait que les sous-systèmes ne sont pas complètement indépendants les uns des autres.

De nombreux travaux ont été consacrés à la T.G., notamment en productique pour l'ordonnancement des tâches (ordre de passage des pièces sur les machines d'un atelier de fabrication), aussi nous renvoyons le lecteur intéressé par cet aspect, à la littérature [LEM91] [KUS88] [DRI87]. Notre objectif ici est de montrer les difficultés rencontrées lors d'une T.G. et de proposer un moyen d'y remédier.

La décomposition est réalisée en effectuant des permutations de lignes (resp. colonnes) visant à regrouper les lignes (resp. colonnes) ayant un indice de similitude maximal. L'indice de similitude peut être défini ainsi:

si  $c_{ij}$  est l'élément d'indice de ligne  $i$  et d'indice de colonne  $j$  de la matrice d'incidence,

$$\text{soient } d_{ij} = \sum_{k=1}^m w(c_{ik}, c_{jk})$$

$$\text{avec } w(c_{ik}, c_{jk}) = \begin{cases} 0 & \text{si } c_{ik} \neq c_{jk} \\ 1 & \text{sinon} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si la ligne } i \text{ appartient au groupe de la ligne } j \\ 0 & \text{sinon} \end{cases}$$

le programme linéaire à résoudre pour les lignes est le suivant:

$$\max \sum_{i=1}^n \sum_{j=1}^m d_{ij} x_{ij} \quad [\text{II.6.2}]$$

sous les contraintes:

1/  $\sum_{j=1}^m x_{ij} = p$  impose un résultat avec exactement  $p$  blocs, l'utilisateur étant libre du choix d'une méthode de calcul des invariants, il impose le nombre de blocs qui conviennent à son algorithme.

2/  $x_{ij} \leq x_{jj} \quad \forall j = 1, \dots, n$  Une ligne  $i$  ne peut appartenir à un bloc  $j$ , que si ce dernier existe ( $x_{jj} = 1$ ).



Ce programme linéaire permet d'obtenir les  $p$  groupes de lignes maximisant la fonction objectif [II.6.2]. Le même traitement appliqué aux colonnes de  $C$  permet de générer les groupes de colonnes et donc les  $p$  blocs diagonaux.

Cependant la T.G. se heurte toujours au problème suivant:

Le but d'une T.G. est de faire apparaître dans la matrice d'incidence le maximum de groupes. Cependant, plus on crée de groupes et plus on augmente le risque de générer des liens résiduels qui seront gênant pour l'étape suivante, à savoir la résolution des sous-systèmes.

Il faut donc trouver un compromis entre les deux alternatives suivantes:

- réaliser une décomposition avec peu de liens résiduels mais aussi peu de groupes, ce qui n'allège pas suffisamment la résolution du système.

- réaliser une décomposition avec plus de groupes mais beaucoup de liens résiduels.

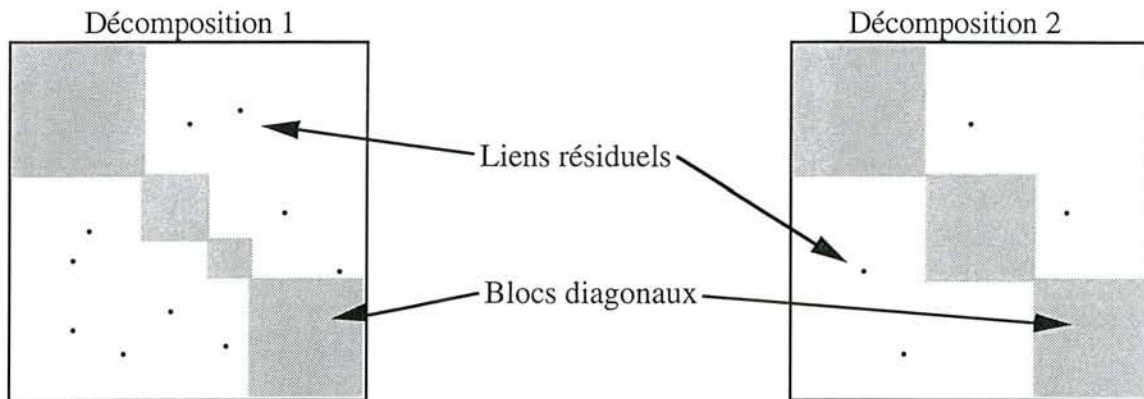


Figure 6.1 Décompositions en blocs diagonaux

Le choix de l'une ou l'autre des décompositions (1 ou 2) (par le choix du nombre de blocs  $p$  dans l'indice de similitude) est fait par l'utilisateur selon la méthode de calcul des invariants qu'il compte employer lors de l'étape suivante:

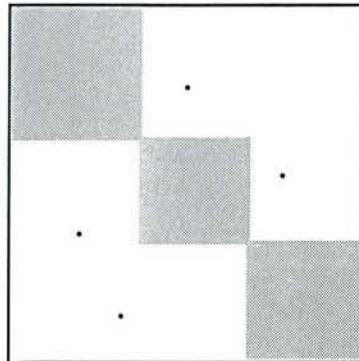
méthodes 1 et 2 → décomposition 1,  
 méthode 3 → décomposition 2.

La méthode 3 étant plus rapide que les deux autres pour des systèmes d'ordre assez élevé (tableau 5.2), la décomposition 2 s'impose et inversement la décomposition 1 est mieux adaptée aux systèmes d'ordre réduit (tableau 5.1).

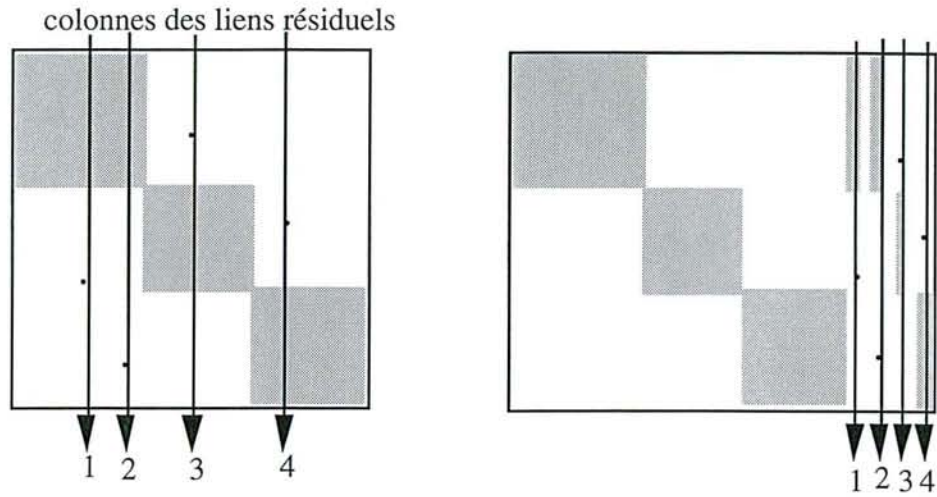
**Remarque 6.1:**

Il est possible de contourner le problème posé par les liens résiduels par une astuce simple. La matrice issue du programme linéaire est à son tour transformée par une procédure très rapide, qui va déplacer toutes les colonnes contenant des liens résiduels (les déplacements sont mémorisés comme dans la résolution des invariants par la troisième méthode § 4 remarque 4.1) après la dernière colonne de la matrice et générer ainsi des blocs diagonaux complètement indépendants et des blocs de couplage isolés séparément. Partant d'une matrice d'incidence en bloc diagonaux, le principe de cette transformation est explicité sur le schéma suivant:

soit une matrice  $C$ , composée de trois blocs diagonaux et quatre liens résiduels,



les colonnes contenant des liens résiduels sont déplacées après la dernière colonne,



On obtient une matrice avec 3 blocs diagonaux indépendants et 3 blocs de couplage:

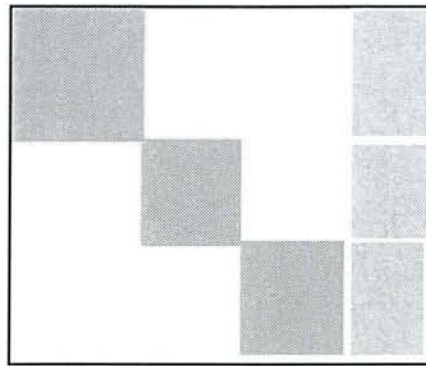


Figure 6.2 Isolement des blocs de couplage

La forme obtenue fig.6.2 est équivalente à la suivante, pour une décomposition en  $p$  blocs:

$$C = \begin{pmatrix} C_1 & 0 & \cdot & 0 & C_{c1} \\ 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot \\ 0 & \cdot & 0 & C_p & C_{cp} \end{pmatrix}$$

avec  $C_i \in \mathbb{Z}^{n_i \times m_i}$ ,  $C_{C_i} \in \mathbb{Z}^{n_i \times m_c}$ ,  $\sum_{i=1}^p m_i + m_c = m$  et  $\sum_{i=1}^p n_i = n$ .

Les blocs de couplage ont tous le même nombre de colonnes (= nombre de liens résiduels), de plus chaque bloc de couplage  $C_{ci}$  a le même nombre de lignes que le bloc diagonal  $C_i$ . La résolution d'un tel système fait l'objet du paragraphe suivant.

## 6.2. Résolution

Soit le système global mis sous forme de blocs diagonaux:

$$\begin{pmatrix} C_1 & 0 & \cdot & 0 & C_{c1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & C_p & C_{cp} \end{pmatrix} \begin{pmatrix} x_1 \\ \cdot \\ x_p \\ x_c \end{pmatrix} = 0 \quad [\text{II.6.3}]$$

La structure [II.6.3] est équivalente à la forme décentralisée suivante:

$$\begin{pmatrix} C_j & C_{cj} \end{pmatrix} \begin{pmatrix} x_j \\ x_c \end{pmatrix} = 0 \quad \text{pour } j = 1, \dots, p \quad [\text{II.6.4}]$$

Le calcul des invariants de chaque sous-système de la forme [II.6.4] est effectué indépendamment des autres puis une procédure appropriée (P1) permettra de retenir les solutions qui répondent à toutes les contraintes du système global.

Procédure  $P_1$ :

Le support minimal du sous-système  $j$  est:

$$X_{Dj} = \begin{pmatrix} x_{j1} & \dots & x_{jDj} \\ \cdot & & \cdot \\ x_{c1} & & x_{cDj} \end{pmatrix} \quad \text{pour } j = 1, \dots, p$$

où  $D_j$  est le rang du support de l'espace des solutions du sous-système  $j$ . En résolvant tous les sous-systèmes, on obtient  $X_D$  la concaténation des  $X_{Dj}$ :

$$X_D = \begin{pmatrix} x_{11} & \dots & x_{ID_1} & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \dots & x_{p1} & \dots & x_{pD_p} \\ x_{cl}^1 & \dots & x_{cD_1}^1 & \dots & x_{cl}^p & \dots & x_{cD_p}^p \end{pmatrix}$$

La solution globale est obtenue en retenant les vecteurs colonnes  $X_j \in X_D$  vérifiant:

$$C.X_j = 0$$

### 6.3. Exemple numérique

Dans l'exemple suivant, une matrice  $C$  découpée en quatre blocs diagonaux ainsi que quatre blocs de couplage est choisie pour modéliser un RdP de grande dimension. Après résolution de chaque sous-système par l'algorithme développé au §4, le support minimal  $X_D$  est issu de la procédure P1:

$$C_1 = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & -1 & 1 \end{pmatrix} \text{ et } C_{c1} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} \text{ et } C_{c2} = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \quad \text{et} \quad C_{c3} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix}$$

$$C_4 = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \end{pmatrix} \quad \text{et} \quad C_{c4} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$



Pour le même système, la solution est calculée en un temps plus de 4 fois moindre malgré l'utilisation d'une machine monoprocesseur. Sur une machine massivement parallèle, le système étant complètement découpé, chaque sous-système pourrait être résolu par un processeur (voir plusieurs processeurs) sans avoir besoin d'échanger des résultats partiels avec ses voisins. Auquel cas le temps de calcul serait pratiquement divisé par le nombre de sous-systèmes créés (la durée de la procédure  $P_1$  étant négligeable devant la durée des phases de résolution des sous-systèmes).

Algorithme	Taille de C	Temps CPU
Algorithme 1	48x44	89.22
Algorithme 3	48x44	32.28
Algorithme décentralisé	48x44	17.5

Tableau 6.1

La recherche du nombre d'opérations élémentaires (additions et multiplications) nécessaires au calcul des invariants nous paraît être une bonne évaluation de la rapidité de l'algorithme choisi. Aussi dans le tableau 6.2 nous présentons les chiffres relatifs à chaque méthode, sachant que, dans un souci de simplification, nous supposons que tous les sous-systèmes ont la même dimension, soit:

$$n_i = n = 10, m_i = m = 15, \text{rang}(C_i) = n_i = q = 10 \text{ et } m-q = r = 5 \text{ pour } i = 1, \dots, p.$$

Notons  $M_1$  la méthode globale consistant à appliquer l'algorithme 3 directement au système à résoudre et  $M_2$  la méthode décentralisée consistant à appliquer le même algorithme 3 à chaque sous-système après décomposition.

Méthode	Multiplications	Additions
$M_1$	$(p.q)^2.(m+1).p+p.q$	$p.q.(p.q-1).[(m+1).p-1]$
$M_2$	$p.[q^3+q^2.r+q-1+q.m.r.(q-1)]$	$p.q.(q-1).[q.r.(m-1).(p-1)+r+q-1]$

Tableau 6.2

La figure 6.3 permet de comparer les deux courbes représentant le nombre d'opérations élémentaires (additions+multiplications) nécessaires au calcul des invariants par les deux



approches ( $M_1$ , méthode globale, représentée en trait coupé et  $M_2$ , méthode décentralisée, représentée en trait plein).

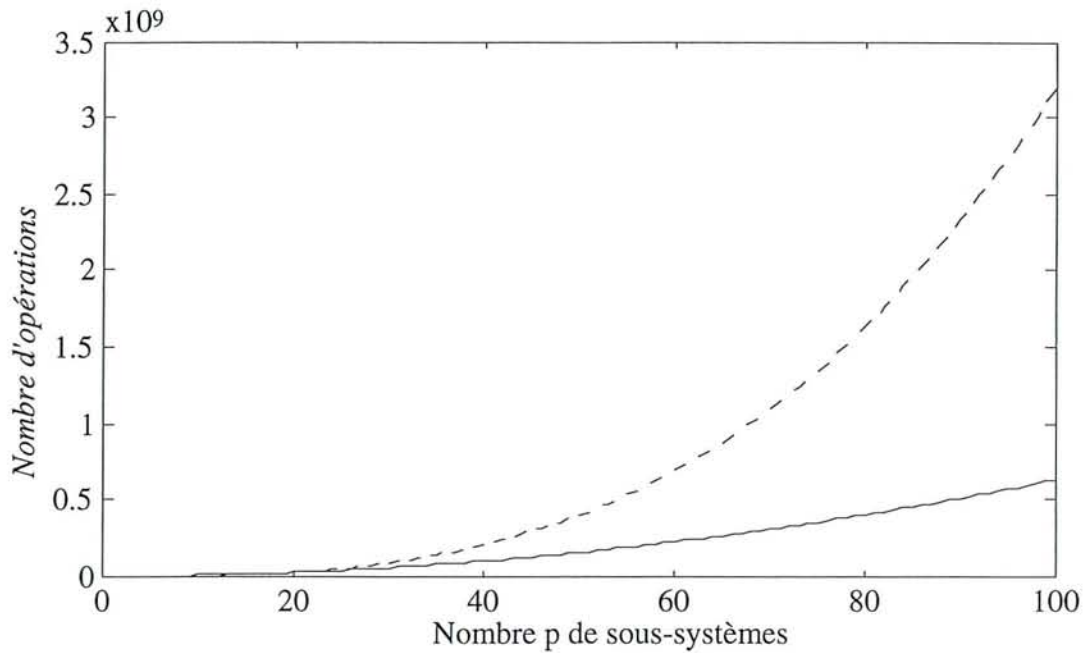


Figure 6.3 Confrontation du nombre d'opérations

## 7. Vérification des propriétés des systèmes

Cette analyse est initiée par un rappel de quelques notions intéressantes sur les invariants et les informations que l'on peut en tirer [MUR88] [MOS89] [POR93], puis sur l'exemple 7.1 du chapitre précédent dont les invariants ont été calculés nous verrons les implications de ceux-ci.

### Lemme 7.1:

Soit  $R$  un RdP muni d'un P-semi-flot  $s$  et soit  $\bar{P} = \{P_i \in P / s(P_i) > 0\}$  alors  $\bar{P}^* = \bar{P}$

Intuitivement ce lemme signifie que les ensembles de places ayant un marquage constant sont obtenus grâce aux ensembles d'arcs reliant une place  $t$  à une place  $t'$ . On peut aussi dire que les places  $\bar{P}$  composant le circuit ne sont alimentées par aucune transition externe.

**Définition 7.1:**

Un RdP est dit couvert par des P-semi-flots si et seulement si, pour chaque place  $P_i$ , il existe un P-semi-flot  $s$  de  $R$  avec  $s(P_i) \neq 0$ .

**Définition 7.2:**

Un RdP  $R$  est dit borné si et seulement si  $M_0$  est fini et s'il existe  $n \in \mathbb{N}$  tel que pour tout  $M_i \in A(R, M_0)$  et pour toute place  $P_i \in P$ , on ait  $M(P_i) \leq n$ .

**Théorème 7.1:**

Soit  $R$ , un RdP et soit  $M_0$  fini. Si  $R$  est couvert par des P-semi-flots alors  $R$  est borné. •

Preuve:

soit  $P_j \in P$ , soit  $s$  un P-semi-flot positif avec  $s(P_j) > 0$  et soit  $M \in A(R, M_0)$ ,

puisque 
$$M(P_j) \cdot s(P_j) \leq \sum_{P_i \in P} M(P_i) \cdot s(P_i) = M \cdot s = M_0 \cdot s$$

alors 
$$M(P_j) \leq \frac{M_0 \cdot s}{s(P_j)}$$

L'exemple choisi pour le calcul des invariants est le RdP 7.1 du chapitre I. Les résultats obtenus sont les suivants:

**T-semi-flot:**

$$S = \{[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]\}$$

Ce problème a une solution unique: " $T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10} T_{11} T_{12} T_{13} T_{14} T_{15}$ ", ce qui montre que le système est capable de se réinitialiser et que de plus il ne le fait qu'après exécution de toutes les tâches prévues, propriété due au fait qu'on est en présence d'un graphe d'événements.

**P-semi-flots:**

$$F = \left\{ \begin{array}{l} [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1] \\ [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1] \\ [1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1] \\ [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1] \\ [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1] \end{array} \right\}$$

On obtient le nombre maximum de P-semi-flots soit  $n\text{-rang}(C)=5$  (dimension de l'espace des solutions dans  $\mathbb{Z}$ ). Si on note  $F_i$  ( $1 \leq i \leq 5$ ) les éléments de  $F$  et en application de [II.2.1] et du théorème 7.1, la solution  $F_1$  permet d'écrire les relations suivantes:

$$M(P_1) + M(P_4) + M(P_7) + M(P_{13}) + M(P_{15}) + M(P_{17}) + M(P_{19}) = M_o(P_1) + M_o(P_4) + M_o(P_7) + M_o(P_{13}) + M_o(P_{15}) + M_o(P_{17}) + M_o(P_{19}) = 1,$$

alors  $M(P_i) \leq 1$  pour  $i$  tel que  $F_{1i} \neq 0$  ( $F_{1i}$  est la  $i$ ème composante du vecteur  $F_1$ )

Les places  $P_i$  mises en jeu dans cette égalité ne seront jamais marquées simultanément et le système passera donc par les états correspondants séquentiellement. De plus ces places sont bornées par un. De la même manière on obtient les relations suivantes:

$$\begin{aligned} M(P_1)+M(P_2)+M(P_5)+M(P_8)+M(P_{10})+M(P_{11})+M(P_{15})+M(P_{17})+M(P_{19}) &= 1, \\ M(P_1)+M(P_3)+M(P_6)+M(P_9)+M(P_{10})+M(P_{11})+M(P_{15})+M(P_{17})+M(P_{19}) &= 1, \\ M(P_1)+M(P_4)+M(P_7)+M(P_{14})+M(P_{16})+M(P_{18})+M(P_{19}) &= 1, \\ M(P_1)+M(P_2)+M(P_5)+M(P_8)+M(P_{10})+M(P_{12})+M(P_{16})+M(P_{18})+M(P_{19}) &= 1. \end{aligned}$$

Chaque relation correspond à un circuit (def. 5.1.1) facilement visible sur le graphe, il en existe d'autres, mais il correspondent à des solutions combinaisons linéaires de celle exhibées. On peut vérifier que certaines étapes peuvent se dérouler en parallèle (exemple:  $P_2, P_3$  ou  $P_4$ ) alors que d'autres ne pourront se faire qu'en séquences bien définies ( $P_2$  suivie de  $P_{10}$ : pas de mélange avant la préparation du lait). Toutes les places appartiennent à un circuit, ce RdP est couvert, donc borné, de plus en application du théorème 4.1 (chapitre I) il est vivant pour le marquage  $[M(P_1)=1$  et  $M(P_i)=0$  pour  $2 \leq i \leq 19]$  et sauf.

Sur cet exemple simple, il a donc été possible de vérifier certaines propriétés du système et ceci par simple calcul des invariants. Il montre donc la puissance d'une telle technique et l'intérêt qu'elle peut présenter.

### 8. Conclusion

La méthode que nous proposons pour le calcul des invariants présente le double avantage d'être simple à programmer, notamment sur un logiciel de type Matlab, et fiable quant aux résultats fournis. L'ordre de la solution dans  $\mathbb{N}$  peut être annoncé rapidement et l'espace de recherche des solutions ( $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ ) est laissé au choix de l'utilisateur. Le défaut majeur des deux premiers algorithmes présentés est d'être assez lourds pour la résolution des systèmes de grande dimension, ce qui n'est pas le cas de la solution proposée (les temps globaux sont souvent deux fois moins importants). Cet algorithme ne présente aucun risque de bouclage infini du simple fait qu'il n'est pas itératif. De plus, il se prête bien au calcul des verrous et des trappes comme cela a été défini dans [MEM78] ou [ALA85]. Enfin, les résultats obtenus et vérifiés sur de nombreux exemples sont d'ordre minimal sans aucune simplification intermédiaire.

## 9. Bibliographie

[**ALA81**] H. ALAIWAN, G. MEMMI "Algorithmes de recherche des solutions entières positives d'un système linéaire d'équations homogènes" Revue Technique Thomson CSF Vol. 14, N° 1, pp. 125-135, mars 1982.

[**ALA85**] H. ALAIWAN, J.M. TOUDIC "Recherche de semi-flots, de verrous et de trappes dans les réseaux de Petri" TSI Vol. 4, N° 1, pp. 104-112, 1985.

[**BAR84**] S. BARNETT "Matrices in control theory" Robert E. Krieger Publishing Company, Malabar, Florida, 1984.

[**BER78**] G. BERTHELOT "Vérification de réseaux de Petri" Thèse de Doctorat de 3ième cycle de l'université Pierre et Marie CURIE, 12 janv. 1978.

[**BER79**] B. BERTHOMIEU "Analyse structurelle des RdP méthodes et outils" Thèse de doctorat d'ingénieur de l'Université Paul Sabatier de Toulouse.

[**BOU93**] A. BOURJIJ, M. BOUTAYEB, M. DAROUACH, G. KRZAKALA "Du calcul des invariants dans les RdP" soumis à la revue TSI (Dec. 93).

[**BOU94**] M. BOUTAYEB, A. BOURJIJ, M. DAROUACH "A decentralized algorithm to determine invariants in Petri Nets" soumis à la revue IEEE Trans. on Circuits and Systems (Juin. 94).

[**BRA83**] G. W. BRAMS "Réseaux de Petri: théorie et pratique" T1 et T2 Masson 1983.

[**DAV89**] R. DAVID, H. ALLA "Du Grafctet aux Réseaux de Petri" HERMES 1989.

- [**DR187**] N. DRIDI, J. M. PROTH "Ordonnement des tâches: une méthode basée sur la technologie de groupes" Second International Conference on Production Systems, I.N.R.I.A., Tome 1, pp. 67-74, avril 1987.
- [**DUF89**] I. S. DUFF, A. M. ERISMAN, J. K. REID "Direct methods for sparse matrices" Clarendon Press Oxford 1989
- [**GAR85**] K. GARG "An approach to performance specification of communication protocols using timed Petri Nets" IEEE Trans. on Software Engineering Vol. SE11, N° 10, pp. 1216-1224, oct 1985.
- [**HOR90**] R. HORN, C. R. JOHNSON "Matrix Analysis" Cambridge Université Press 1990.
- [**KUS88**] A. KUSIAK, S. C. WING "Decomposition of manufacturing systems" IEEE journal of Robotics and Automation, Vol. 4, N° 5, octobre 1988.
- [**LAN76**] S. LANG "Structures algébriques" InterEditions Paris 1976.
- [**LAN85**] P. LANCASTER, M. TISMENETSKY "The theory of matrices" Computer science and applied mathematics, Academic press inc. 1985
- [**LAN89**] J. LANDIN "An introduction to algebraic structures" Dover Publications, INC. New York, Paris 1989.
- [**LEM91**] H. LEMONIAS "Une application aux problèmes d'odonnement, la technologie de groupe" Ecole d'été d'automatique de Grenoble, 9-13 sept. 1991.
- [**MAR81**] J. MARTINEZ, M. SILVA "A simple and fast algorithm to obtain all invariants of a generalized Petri Net" 2nd European workshop on application and theory of Petri Net, Bod-honneff, 1981.

- [**MAR91**] D.C. MARINESCU, M. BEAVEN, R. STANSIFER "A parallel algorithm for computing invariants of Petri Net models" Proceedings of the fourth international workshop on PNs and performance models, Melbourne, IEEE Computer Soc., pp. 136-43, 1991.
- [**MEM78**] G. MEMMI "Fuites et semi-flots dans les réseaux de Petri" Thèse de Docteur-ingénieur de l'Université Pierre et Marie CURIE, 18 déc. 1978.
- [**MIN83**] M. MINOUX "Programmation mathématique" Dunod, Bordas, 1983.
- [**MOS89**] A. MOSLEMIE "A logical basis automated interpretation of S-invariants of predicate/transition nets" CH2766-4/89/0000-0248, 1989 IEEE.
- [**MUR88**] T. MURATA, B. SHENKER, S.M. SHATZ "Detection of Ada deadlocks using Petri Net invariants" IEEE Transactions on Software Engineering Vol. 15, N° 3, pp. 314-326, march 1989.
- [**MUR89**] T. MURATA "Petri Nets: properties, analysis and applications" Proceedings of IEEE Vol. 77, N° 4, pp. 571-580, april 1989.
- [**NOD92**] P. NAUDIN, C.QUITTE "Algorithme Algébrique" Masson 1992.
- [**POR93**] L. PORTINALE "Exploiting T-invariant analysis in diagnostic reasoning on a Petri net model" 16th International conference on Application and Theory of Petri nets 21-25 jun. 1993, Chicago. Lecture Notes in Computer Science, Vol. 6, 1991, Springer-Verlag.
- [**SIL88**] M. SILVA, J.M. COLOM "On the computation of structural synchronic invariants in P/T nets" Lecture Notes in Computer Science Vol. 340, pp. 386-417, Springer-Verlag, 1988.

[**TAN91**] T. TANIDA, T. WATANABE, K. ONAGA "A polynomial-time algorithm for finding a semi-generator of Petri Net invariants" CH 3006-4/91/0000-2838, IEEE 1991.

[**TOU82**] J. M. TOUDIC "Algorithmes d'algèbre linéaire pour l'analyse structurelle des réseaux de Petri" Revue Technique Thomson-CSF, Vol. 4, N° 1, Mar. 1982.



## CHAPITRE III

### LA SURVEILLANCE

## 1. Introduction

Avec l'essor de l'informatique et l'évolution exponentielle de l'échelle d'intégration des composants, l'automatisation dispose d'autant d'atouts supplémentaires. En effet, la commande des procédés industriels, initialement réalisée par des cartes analogiques, est de plus en plus réalisée par des cartes numériques (type automates programmables) voire des calculateurs. Ceci est d'autant plus vrai que le coût des moyens de calcul (notamment les ordinateurs personnels) connaît une évolution inversement proportionnelle aux développements techniques. La commande numérique touche un éventail de processus toujours plus large, depuis le simple appareil d'analyse jusqu'aux procédés de très haute technologie tels qu'une centrale nucléaire. Cependant, cette automatisation accrue et sophistiquée met en œuvre un grand nombre de paramètres souvent impossible à contrôler en temps réel par un opérateur, d'où la nécessité de systèmes de surveillance capables de suivre l'évolution des processus et de faciliter l'intervention de l'être humain. Initialement, les systèmes de surveillances se cantonnaient à un rôle de supervision c'est-à-dire qu'ils permettaient d'afficher et d'analyser les données générées par des systèmes automatisés. Ces logiciels de supervision trouvent une large audience dans l'industrie, aussi seront-ils intégrés dans le système de surveillance proposé dans ce chapitre.

Dans le paragraphe suivant, nous présentons globalement les systèmes de surveillance, puis dans le paragraphe 3, quelques techniques de détection de défauts basées sur l'utilisation des RdP, notamment par le calcul des invariants. Enfin nous concluons sur une discussion du choix et de l'adéquation des méthodes proposées aux différents types de systèmes.

## 2. Les systèmes de surveillance

Les logiciels de supervision sont largement utilisés dans l'industrie aussi existe-t-il des versions pour micro-ordinateur (P.C.) à des prix très abordables. Ils permettent à un opérateur de suivre (en temps réel) l'évolution de procédés industriels automatisés. Généralement ce type de logiciel dispose d'un module graphique pour la représentation du système commandé. Des variables affichées sur un écran donnent l'état des capteurs, des actionneurs et signalent d'éventuelles alarmes. Le schéma de principe d'un tel système est donné par la figure 2.1.

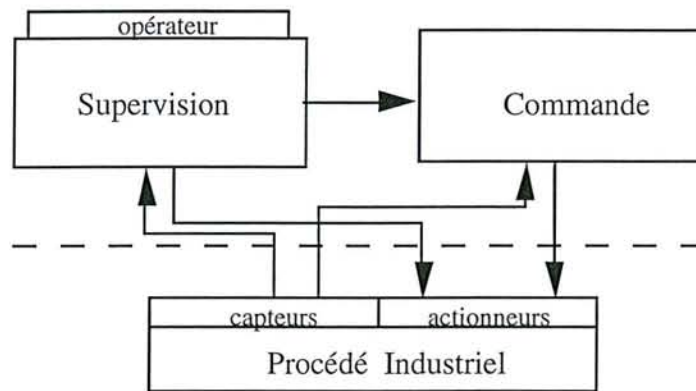


Figure 2.1 *Supervision*

L'opérateur garde souvent un rôle prépondérant car son expérience et sa connaissance du procédé sont jugés indispensables d'autant plus que les modèles ne sont pas toujours très fiables [GRE85]. Cependant de tels logiciels ne constituent pas à eux seuls un système de surveillance autonome, mais plutôt un outil d'aide à la prise de décision. Un tel système reste faillible et sujet aux erreurs humaines: inattention, mauvaise interprétation,....

Les systèmes de surveillance actuels sont beaucoup plus performants car ils permettent en plus de détecter les défauts, de les localiser et de prendre les décisions adéquates, qu'elles soient totales (reconfiguration de la commande, arrêt d'urgence,...) ou bien partielles (une simple aide à la prise de décision par l'opérateur). Le système de surveillance préconisé dans ce chapitre propose, non seulement, d'utiliser l'existant, c'est-à-dire un logiciel de supervision qui constitue un excellent interface avec l'opérateur et un bon outil de visualisation du procédé, mais aussi de compléter le champ de surveillance pour intégrer la détection de défauts, la localisation de ces derniers et enfin la prise de décision. L'architecture d'un tel système est représentée par le schéma suivant:

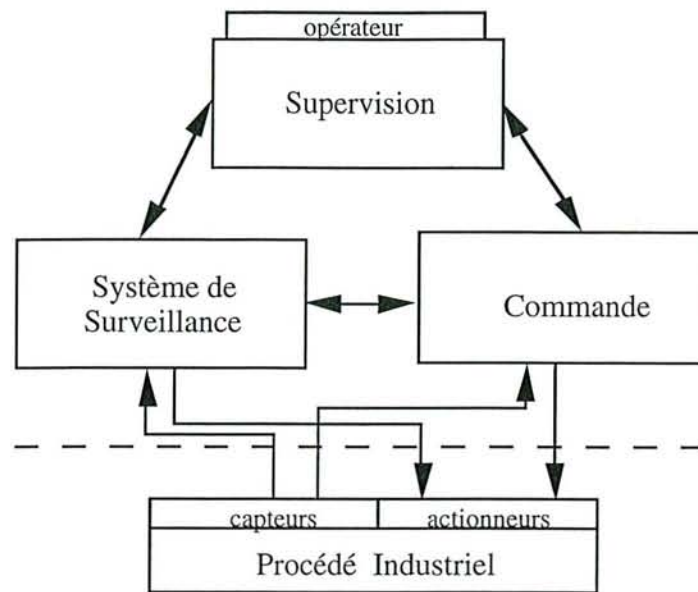


Figure 2.2 *Système de surveillance*

D'autre part vu la complexité des processus industriels, l'importance des sommes investies et les objectifs de l'heure actuelle à savoir (les flux tendus):

- zéro défaut,
- zéro délai,
- zéro stock,

il est devenu incertain de laisser la surveillance de telles usines à la vigilance des seuls opérateurs. Les systèmes de surveillance peuvent bénéficier du double avantage de la rapidité des processeurs (souvent en parallèle) permettant un traitement en temps réel et d'une grande capacité de stockage des mémoires vives et mémoires de masse (plusieurs giga-octets), garantissant un accès rapide à une base de connaissances la plus riche possible. Ce sujet fait l'objet de nombreux travaux [LAP85], [AUB91], [SAH92], [CAS94],... et la plupart de ces auteurs s'accordent à reconnaître qu'un système de surveillance, pour être efficace, doit être conçu séparément tout en interagissant avec le système de commande. Le système présenté dans ce travail s'appuie sur ce concept tout en développant quelques particularités. La surveillance est réalisée par plusieurs modules spécialisés qui communiquent entre eux. Ces derniers sont organisés en un système hiérarchique où la base est formée par le système d'acquisition des données et le sommet, la prise de décision, qu'elle soit automatique (action directe sur le procédé lui-même ou sur le système de commande) ou semi-automatique (aide à la prise de décision).

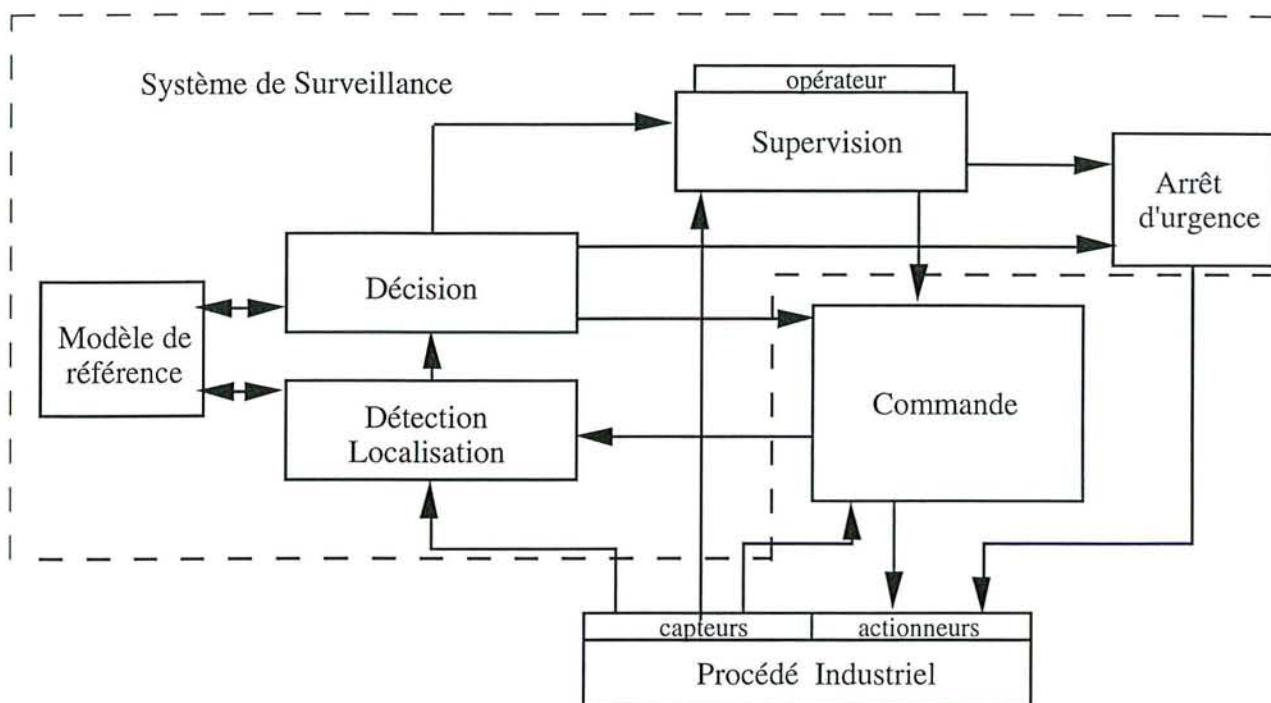


Figure 2.3 Structure d'un système de surveillance

Le module détection/localisation de la figure 2.3 reconstruit l'état réel du système grâce aux acquisitions. Cet état comparé à un modèle de référence permet de générer des résidus et donc de détecter les états incohérents que pourrait prendre le système commandé. Disposant de l'ensemble des informations, le défaut ayant été détecté et localisé, le rôle du module de décision sera de choisir la stratégie à adopter:

- informer l'opérateur si ce type d'incident n'est pas prévu,
- afficher une simple alarme,
- agir directement sur le processus, déclencher l'arrêt d'urgence,
- reconfigurer la commande.

Dans tous les cas, s'il est nouveau, l'incident sera répertorié et rajouté à la base de connaissances.

Les modules du système de surveillance peuvent être écrits soit dans un langage de programmation classique (C, Pascal, ...), soit dans le langage fourni par le logiciel de supervision. Cette dernière possibilité simplifie les communications inter-modules mais reste souvent trop lourde (langages souvent graphiques mais inadaptés aux calculs complexes et aux communications avec des modules extérieurs car tous les drivers ne peuvent être prévus).

Les Ateliers Flexibles (Flexible Manufacturing Systems) sont l'un des domaines où les systèmes de surveillance sont les plus utilisés [JUD94] [ROS93] [ROU92] [ZHO92] [ZHO94] [WIE88]. En effet dans ces unités de production de nombreux processus concurrents se partagent un ensemble de ressources d'où les risques de blocage. De tels problèmes sont bien décrits et analysés par les RdP notamment les RdP à objets [BEN91] [BOU91] qui de plus sont un outil de commande [BRA88]. C'est donc un exemple privilégié d'utilisation d'un système de commande et de surveillance basé sur les RdP [AGA92] [BON93].

### **3. La détection**

Les techniques classiques de détection de défauts, depuis le simple test statistique jusqu'à la logique floue, ont été abondamment traitées par de nombreux auteurs [PAT90], [RAG90], [BRU90]. Mais les approches basées sur les RdP demeurent encore assez récentes et peu exploitées, aussi essaierons-nous d'approfondir ce sujet. Lors du chapitre II, l'intérêt des invariants ayant été présenté, nous exploiterons ces résultats dans l'optique de la détection des défauts.

#### **3.1. Détection par modèle de référence**

##### **3.1.1. Concept**

L'originalité d'une telle méthode repose sur l'utilisation d'un modèle de référence décrit par les RdP [COM91] [CON91] [CON92], l'utilisation la plus courante des RdP étant l'analyse des systèmes [ZAR90] [YEN92] [TER93] [SRI93] ou la modélisation et la commande des systèmes à événements discrets [USH89] [WIL90] [VAL93]. La première étape préconisée pour la conception d'un système de surveillance est l'étude du procédé pour dégager ses fonctionnalités et ses propriétés, ce qui permettra d'établir un modèle grâce aux RdP. Le rôle de ce modèle sera de refléter l'état normal du procédé pour constituer une référence en temps réel. Ce modèle doit donc être le plus fidèle possible, aussi est-il toujours plus sûr de décomposer le système en sous-systèmes plus simples à étudier. Pour chaque sous-système, on établit un inventaire le plus exhaustif possible des

états accessibles par le procédé, toutefois pour la clarté de la représentation, seuls les états importants et/ou sensibles seront retenus et constitueront les places du RdP. De même, un recensement des événements, dont l'occurrence implique un changement d'état, fournit les transitions du RdP. La relation de précédence entre les états et les événements constitue les arcs du RdP. Cette approche, bien que simple, s'avère efficace pour la conception des modèles, d'autant plus qu'il n'existe pas de recette miracle pour la génération automatique de RdP.

Le modèle obtenu est décrit par une matrice d'incidence, l'état du système est, quant à lui, donné par un simple vecteur colonne (nombre de marques dans chaque place).

Le système de surveillance est chargé de garantir qu'à tout moment le modèle de référence est dans le même état que le procédé commandé. Pour cela, il doit communiquer avec le système de contrôle et appliquer les mêmes commandes au modèle. L'évolution du modèle est réalisée par le franchissement de séquences de transitions. Dans certaines méthodes de détection de défauts, l'état du système est comparé à des états inacceptables connus ou déjà rencontrés et mémorisés dans une base de connaissances. Dans la méthode proposée, on effectue l'opération inverse, l'état instantané du système est comparé à un état supposé normal, à savoir, l'état du modèle. Le modèle étant insensible aux bruits et perturbations, sûr de tout risque de panne (qu'elles soient de capteurs, d'actionneurs ou encore d'autres éléments du procédé), les risques d'erreurs ne peuvent être dus qu'aux séquences de franchissement réalisées.

### 3.1.2. Application

Pour réaliser la détection, le système de surveillance construit le vecteur colonne  $M$  représentant l'état du processus réel. On suppose que l'on peut disposer de la totalité de l'état du système et donc qu'il est complètement mesuré. Cet état est alors comparé périodiquement (période  $= \tau$ ) à celui du modèle dans les mêmes conditions. Le choix de la période d'échantillonnage  $\tau$  est effectué en fonction de la vitesse d'évolution du procédé et de la rapidité des moyens de calcul dont on dispose. L'état initial du processus commandé,  $M_0$ , ainsi que celui du modèle,  $M_{0r}$ , sont supposés connus et bien sûr égaux. Ainsi, l'algorithme de détection basé sur un modèle de référence peut être présenté comme suit:

**Algorithme 1:**

Initialisations; à  $t = 0$   $M_0 = M_{0r}$

**Tant que (vrai) faire**

Acquisition (données);  
Reconstruction ( $M(t), M_r(t)$ );

$$\text{résidu} = \sum_{i=1}^{|P|} |M(P_i) - M_r(P_i)|$$

**si** (résidu  $> \varepsilon$ ) **alors**  
Localisation (défauts);  
Décision;

**Finsi**

$t = t + \tau$ ;

**Fintq**

$M(P_i)$  est le marquage de la place  $P_i$ , une coordonnée du vecteur d'état du procédé,  
 $M_r(P_i)$  est le marquage de la place  $P_i$ , une coordonnée du vecteur d'état du modèle de référence.

Le module d'acquisition fournit périodiquement les données prélevées sur le processus. Ces données sont utilisées par le module de reconstruction pour créer le vecteur  $M$ , état du système à l'instant  $t$ , le marquage de chaque place correspond à l'état d'un élément du processus (vanne ouverte, niveau, température, ...). L'état de référence à chaque instant est calculé par la relation fondamentale des RdP:

$$M_r(t) = M_r(t-1) + C \cdot \sigma$$

où  $M_r(t-1)$  est l'état de référence à l'instant précédent,  $C$  est la matrice d'incidence et  $\sigma$  une séquence de franchissement correspondant à la commande appliquée au processus à l'instant  $t-1$ . La séquence  $\sigma$  n'est correcte et franchissable que si elle vérifie la règle:

$$M_r(t-1) - C^- \cdot \sigma \geq 0 \quad \text{où } C = C^+ - C^- \quad [\text{III.3.1}]$$

Alors, un état est incohérent s'il vérifie l'inégalité suivante:



**si**  $|M(P_i) - M_r(P_i)| > \varepsilon_i$  **alors**  $P_i$  est un état incohérent **Finsi.**

Le choix des seuils  $\varepsilon$  et  $\varepsilon_i$  dépend de la précision des capteurs et du nombre de marques véhiculées par le RdP, ils sont souvent pris nuls (cas des réseaux saufs). Les résidus sont calculés dans cet algorithme avec une norme 1 [HOR90], mais toute autre norme peut être utilisée en fonction de la sensibilité recherchée. En effet les risques sont soit de détecter de fausses pannes ( $\varepsilon$  trop petit) soit de ne pas détecter certains défauts ( $\varepsilon$  trop grand). Cependant, après une phase d'apprentissage pour déterminer les bonnes valeurs pour les " $\varepsilon_i$ ", le calcul des résidus suffit à la détection des défauts, de plus la localisation est rapide.

### 3.1.3. Localisation

Soit  $M_r$  l'état estimé du système par modèle de référence,  $M_{r/0}$  son état initial et  $M_0$  l'état initial réel. Il n'est pas très contraignant et il est réaliste de supposer que l'on connaît l'état initial du système (ce qui est essentiel pour un RdP), dans ce cas:

$$M_{r/0} = M_0$$

et 
$$M_{k+1} = M_k + C \cdot \sigma_k$$

L'algorithme précédent permet de vérifier qu'à chaque période d'échantillonnage:

**si**  $|M_{k+1} - M_{r/k+1}| \leq \varepsilon$  **alors** l'état du système est normal,  
**sinon** on est en présence d'un défaut **Finsi.**

Soit  $M_{k+1}^i$  le marquage défaillant de la place  $P_i$  à l'instant  $k+1$  (noté précédemment  $M_{k+1}(P_i)$ ). Le problème est de déterminer l'origine du défaut. Un défaut peut être dû à une panne de capteur ou une défaillance du système. Pour systématiser la localisation du défaut nous proposons un test simple. On pose comme hypothèse qu'un défaut du système a toujours une incidence sur d'autres composantes du système (exemple: une électrovanne défaillante qui ne s'ouvre pas va provoquer une surchauffe de la bouilloire et une diminution anormale du niveau d'eau). On peut alors décentraliser le système en plusieurs sous-systèmes (comme cela a été présenté lors du chapitre précédent) dont les états sont

corrélés. Le raisonnement inverse consiste à dire qu'un capteur fournit une fausse information mais n'agit pas sur les autres éléments du système si la commande n'utilise pas le feed-back (commande en boucle ouverte). Il suffit donc d'observer si dans les instants suivant le moment de détection du défaut, d'autres erreurs apparaissent sur le système et si elles ont une corrélation possible avec le premier défaut constaté:

**si**  $M_{k+2} \neq M_{r/k+2}$  **alors**  
**si**  $\exists j \neq i$  tel que  $M_{k+2}^j \neq M_{r/k+2}^j$ ,  $(P_j \in R_k)$  et  $(P_i \in R_k)$  **alors** la  
défaillance est due au système (sachant que toute place  $P_i$  appartient à au moins  
un sous-système  $R_k$ ),  
**sinon** la défaillance est due au capteur.  
**finsi**  
**finsi**

Remarquons que le nouveau défaut apparu à l'instant  $k+2$  peut être un nouveau défaut et non une conséquence du précédent, mais on suppose que cette probabilité est assez faible pour être négligée. Cet algorithme est applicable aux systèmes à évolution rapide dont une étude préalable a permis de déterminer les sous-systèmes.

### Définition 3.1:

On dit que deux places  $P_i$  et  $P_j$  sont corrélées, si l'apparition d'un défaut sur  $P_i$  va engendrer une erreur sur  $P_j$ . Le lien entre les deux places peut ne pas être représenté par le RdP.

Cependant les conséquences d'un défaut peuvent apparaître après plusieurs pas (franchissement de plusieurs séquences) normaux, nous avons donc besoin de savoir:

comment sont corrélées les places entre elles (définition 3.1),

et quelle est la durée d'observation du système avant de pouvoir décider si le défaut est dû au capteur ou au système.

Dans cette optique nous proposons un algorithme d'isolation des défauts.

Remarque: Il n'existe pas forcément un lien direct (arc pré, transition et arc post) entre eux places corrélées. Aussi, seul une bonne connaissance du système permet de répertorier pour chaque place, toutes celles qui lui sont corrélées.

Nous proposons de consigner ces informations dans une matrice, R, des corrélations inter-places. R sera donc une matrice carrée de dimension  $n \times n$  ( $n = |P|$ ), binaire, telle que:

$$\begin{cases} R(i,j) = 1 & \text{si } P_i \text{ est corrélée à } P_j \text{ et } (i \neq j) \\ R(i,j) = 0 & \text{sinon} \end{cases}$$

On construit une matrice, D, des défauts constatés à chaque instant. D contient n lignes et une colonne par défaut. Chaque colonne contenant une seule composante non nulle ( $D(i,j)=1$  indique un défaut sur la place  $P_i$  et  $\forall k \neq i \ D(k,j)=0$ ).

**Exemple:** soit un système modélisé par un RdP à 5 places, si on constate deux défauts sur

$$P_1 \text{ et } P_3 \text{ tels que } \begin{cases} |M(P_1) - M_r(P_1)| > \varepsilon \\ |M(P_3) - M_r(P_3)| > \varepsilon \end{cases}$$

$$\text{alors } D = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

La matrice, D', ci-dessous:

$$D' = R.D,$$

est telle que chacune de ces colonnes fournit l'ensemble des places pouvant être affectées par les défauts constatés. Cependant, on peut éliminer dans ce test toutes les places qui ne font l'objet d'aucune évolution à cet instant. En effet, pour les systèmes de grande dimension, en limitant l'espace d'investigation, le gain de temps peut être relativement important. Cette indication peut être obtenue de la façon suivante:

$$\begin{aligned} \text{soit } M_{k+1} &= M_k + C.\sigma \\ \Leftrightarrow \Delta M &= C.\sigma \end{aligned}$$

On décompose le vecteur  $\sigma$  en une somme de vecteurs binaires (afin de ne perdre aucune information car deux transitions franchies simultanément peuvent avoir un effet

compensatoire sur une ou plusieurs places, l'une retirant une marque et l'autre en rajoutant une, voir par exemple figure 3.1):

$$\sigma = \sum_{i=1}^k a_i \cdot \sigma_i \quad \text{où } a_i \text{ est un entier positif et } \sigma_i \text{ un vecteur binaire de } \mathbb{N}^m$$

$$S = \left( a_1 \cdot \begin{pmatrix} \sigma_1 \end{pmatrix} \quad \dots \quad a_k \cdot \begin{pmatrix} \sigma_k \end{pmatrix} \right)$$

sur le RdP suivant, le franchissement de la séquence "T<sub>1</sub>T<sub>2</sub>" affecte les places P<sub>1</sub>, P<sub>2</sub> et P<sub>3</sub> pourtant si on n'observe que le résultat final, seuls les marquages des places P<sub>2</sub> et P<sub>3</sub> ont évolué. D'où l'intérêt de la décomposition des séquences de franchissement en vecteurs binaires.

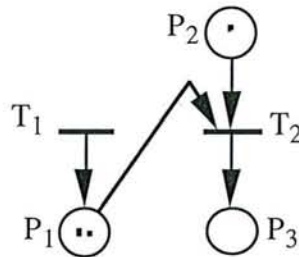


Figure 3.1 Intérêt de la décomposition d'une séquence.

soit  $X = C.S$ ,

alors le vecteur,  $\tilde{X}$ , obtenu par la transformation suivante:

$$\begin{cases} \tilde{X}(i) = 1 & \text{si } \exists X(i,j) \neq 0 \quad \forall j \\ \tilde{X}(i) = 0 & \text{sinon} \end{cases}$$

fournit l'ensemble des places ayant subi un changement lors du franchissement de la séquence  $\sigma$  (P<sub>i</sub> a été modifiée si  $\tilde{X}(i) = 1$ ).

Enfin l'ensemble des places recherchées (places susceptibles d'être affectées, à cet instant, par un défaut apparu précédemment) est donné par la matrice:

$\tilde{D} = D' \odot \tilde{X}$ .      l'opérateur  $\odot$  est le produit de Hadamard.

Chaque colonne  $j$  de  $\tilde{D}$  donne l'ensemble des places susceptibles d'être affectées par un défaut constaté:  $\{P_i / \tilde{D}(i,j) = 1\}$ .

Une telle approche est d'autant plus simple si la commande est réalisée à l'aide des RdP.

#### 3.1.4. Conclusion

L'information est remontée depuis le processus où se situe l'origine du défaut jusqu'à un module de décision, cerveau du système de surveillance. Ce système hiérarchisé simplifie les communications, ce qui permet une prise de décision rapide avec un risque d'erreur minimal. Cette méthode de détection a le double avantage d'être rapide et très simple à mettre en œuvre. Le modèle de référence étant une simple matrice, il est facile à mémoriser et le calcul de son évolution est immédiat. De plus, la localisation des défauts est réalisée simultanément. Cependant, cette méthode peut s'avérer complètement inutile si le système de surveillance n'arrive pas à synchroniser correctement l'état du modèle et celui du processus réel, d'où l'importance du choix et du respect de la période d'échantillonnage.

### 3.2. Détection par les invariants

#### 3.2.1. Concept

Des auteurs tels que J. PROCK [PRO91], T. RACHELL et D. G. MEYER [RAC92] ont introduit cette méthode originale de détection de défauts par les invariants dans le cas de RdP particuliers (Graphe d'événements et RdP vivant et sauf à choix libre). Cette approche est reprise et développée ici, dans le cas des systèmes modélisés par des RdP généralisés qu'ils soient interprétés ou pas. Comme pour la méthode précédente, on suppose que le système automatisé est suffisamment instrumenté pour permettre de reconstruire totalement l'état  $M(t)$  du système à chaque instant  $t$ . Le calcul des invariants de marquage, objet du

chapitre précédent, permet de disposer de relations qui restent vraies quelle que soit l'évolution du système.

### 3.2.2. Application

Soient  $F_i$  les vecteurs semi-flots vérifiant l'équation:

$$F_i^T \cdot C = 0 \quad C \text{ étant la matrice d'incidence du RdP.}$$

A chaque instant, le marquage atteint, appelé  $M(t)$ , devra donc satisfaire les relations suivantes:

$$F_i^T M(t) = F_i^T M(0), \quad 1 \leq i \leq |S| \quad [\text{III.3.2}]$$

où  $M(t)$  est le vecteur des marquages du RdP à l'instant  $t$ ,  $M(0)$  le marquage initial et  $|S|$  le nombre de semi-flots.

#### **Algorithme 2:**

```

Initialisations;

Tant que (vrai) faire
    Acquisition (données);
    Reconstruction ( $M(t)$ );

    Pour  $i = 1$  à  $|S|$  faire
        si  $F_i^T M(t) \neq F_i^T M(0)$  alors
            Localisation (défauts);
            Décision;
        Finsi
    Finpour
     $t = t + \tau$ ;
Fintq
  
```

Connaissant le marquage initial, on peut donc périodiquement acquérir l'état du système  $M(t)$  et évaluer les relations [III.3.2]. Dans le cas où l'une d'elles n'est pas vérifiée, on est en présence d'un défaut. De plus la relation ne faisant intervenir qu'un nombre limité de

places du RdP, la localisation est encore une fois immédiate car souvent les places communes à un invariant représentent les différents états d'un même module et donc un circuit.

### **Définition 3.2:**

Un **circuit** dans un RdP est un chemin commençant et finissant à un même nœud (place ou transition) et où tous les autres nœuds du chemin n'apparaissent qu'une seule fois.

T. RACHELL et D. G. MEYER ont démontré dans un récent article [RAC92] qu'un ensemble de places =  $\{P_{i_1}, P_{i_2}, \dots, P_{i_n}\}$  d'un graphe marqué vivant et sauf forment un circuit si et seulement si le vecteur binaire  $v$  associé est un invariant. Sachant qu'un circuit peut constituer un invariant et que les différents états d'un même composant (bouton marche/arrêt par exemple) forment un circuit, notre approche est donc inverse. Dans la modélisation par les RdP, on s'attachera à inventorier les composants du système et leurs états respectifs mais surtout il faudra conserver les circuits et ainsi faire apparaître un maximum de relations d'invariance. Il est donc indéniable qu'une telle méthode est liée à la structure du réseau et que, pour être efficace, chaque place doit être impliquée dans au moins un invariant (voir définition 5.1.2 du chap. I).

### **3.2.3. Conclusion**

Le calcul des invariants peut être assez pénalisant en temps CPU mais il est bien évidemment effectué en dehors de cet algorithme puisqu'il sert à valider le modèle. Aussi, cette méthode est-elle encore plus rapide et simple à implémenter que la précédente. Toutefois, elle a ses limites car elle n'est efficace que si on dispose de suffisamment d'invariants (cas des réseaux fortement connexes) pour tester tous les états ou au moins les plus importants. D'autre part certains défauts risquent de ne pas être détectés, car un marquage peut vérifier une relation invariance et être incohérent. Enfin, dans le cas où le RdP n'a pas d'invariants (cas où  $M$  a une ligne strictement négative II.4.3), la méthode est inutilisable.

### 3.3. Remarques

1/ Certains auteurs, [JAL92] [HUR88], réalisent la détection de défauts par exploration d'arbres de fautes. Contrairement aux approches précédentes, le RdP ne modélise plus le système lui même mais les arbres des défauts et accidents susceptibles d'intervenir. L'équation fondamentale des RdP et les règles de franchissements sont alors utilisées pour assister un opérateur à diagnostiquer un défaut ou un accident et à identifier les alarmes. Les places du RdP symbolisent les symptômes de chaque défaut. La présence d'une marque dans une place signifie l'occurrence d'un symptôme particulier. Le marquage initial du RdP est constitué des symptômes d'accidents ou alarmes. Après franchissement des transitions validées, le marquage final obtenu fournit l'information attendue, c'est-à-dire l'occurrence d'un défaut (respectivement pas de défaut) si la place sortie du RdP contient une marque (respectivement pas de marque). En cas de fonctionnement normal, la place sortie du RdP (arbre du défaut) ne contient pas de marque.

Une telle méthode peut s'avérer rapide et efficace mais elle suppose une bonne connaissance du système, un recensement exhaustif des défauts et accidents ainsi que leurs causes et symptômes et la présence d'un opérateur averti.

2/ Un système discret est décrit par les équations:

$$\begin{cases} x_{k+1} = A \cdot x_k + B \cdot u_k \\ y_k = H \cdot x_k \end{cases} \quad [\text{III.4.1}]$$

une condition nécessaire et suffisante d'observabilité est:

$$\text{rang} \begin{bmatrix} H \\ HA \\ \dots \\ HA^{n-1} \end{bmatrix} = n$$

De la même manière si on applique ce théorème sur les systèmes décrits par les RdP on obtient le système suivant:



$$\begin{cases} M_{k+1} = I.M_k + C.\sigma_k \\ y_k = H.M_k \end{cases} \quad \text{[III.4.2]}$$

où les matrices A et B du système [III.4.1] sont I et C dans le système [III.4.2]. Auquel cas, la condition nécessaire et suffisante devient:

$$\text{rang} \begin{bmatrix} H \\ HI \\ HI^2 \\ \dots \\ HI^{n-1} \end{bmatrix} = \text{rang}[H] = n$$

mais une telle condition n'a pas de sens puisque la matrice A du système [III.4.1] est I dans le système [III.4.2], la condition d'observabilité du système ne porte que sur H, ce qui implique que pour être observable, l'état d'un tel système doit être entièrement mesuré. Cependant, on peut considérer que l'état d'un système est constitué de M (marquage) et de  $\sigma$  (séquence de franchissement). Auquel cas, on obtient un système singulier sous-déterminé (dans la littérature, on parle aussi de système implicite ou généralisé):

$$\begin{cases} (I \quad -C) \cdot \begin{pmatrix} M_k \\ \sigma_k \end{pmatrix} = (I \quad 0) \cdot \begin{pmatrix} M_{k-1} \\ \sigma_{k-1} \end{pmatrix} \\ y_k = \begin{pmatrix} H_1 & H_2 \end{pmatrix} \begin{pmatrix} M_k \\ \sigma_k \end{pmatrix} \end{cases} \quad \text{[III.4.3]}$$

le vecteur  $y_k$  étant une mesure partielle de l'état  $M_k$  et  $\sigma_k$ . A partir de ces mesures, on peut alors reconstituer M et  $\sigma$ , c'est-à-dire construire une estimation  $\hat{M}$  et  $\hat{\sigma}$  de M et  $\sigma$ , par des algorithmes basés sur des approches telles que :

- le maximum de vraisemblance [NIK92],
- les moindres carrés [DAR93],
- la variance minimale [BAS92], [DAR94].

Le franchissement d'une séquence de transitions,  $\sigma$ , étant instantané, on considère, dans le système [III.4.3], qu'il a lieu lorsque le marquage du réseau, M, évolue. Cette différence entre [III.4.2] (où  $\sigma$  est considéré comme une entrée exogène tout comme u dans [III.4.1])

et [III.4.3] se justifie pour deux raisons. D'une part, le franchissement des transitions peut être considéré, à juste titre, comme endogène à la dynamique du RdP. D'autre part, on obtient alors une condition d'observabilité moins restrictive que dans [III.4.2]. En effet, pour estimer l'état du réseau (c'est-à-dire pour obtenir  $\hat{M}$  et  $\hat{\sigma}$ ) à partir des mesures  $y$ , il faut que le triplet de matrices  $[(I \ -C), (I \ 0), (H_1 \ H_2)]$  soit détectable et que la matrice  $\begin{pmatrix} I & -C \\ H_1 & H_2 \end{pmatrix}$  soit de rang plein colonnes [NIK92], [BAS92], [DAR93], [DAR94].

Les approches proposées dans ces quatre articles donnent des estimateurs de l'état équivalents. Ainsi, lorsque l'état du réseau est partiellement mesuré, on peut appliquer ces estimateurs aux RdP pour le diagnostic par modèle de référence et par les invariants ( $\hat{M}$  et  $\hat{\sigma}$  remplacent  $M$  et  $\sigma$ ), les méthodes de détection de défauts basées sur un observateur de l'état étant étudiées en détail dans [PAT90] et [RAG90].

#### 4. Discussion

Le choix entre ces différentes méthodes est conditionné par la structure du système étudié, notamment certains d'entre eux n'ont pas de P-semi-flot, auquel cas l'algorithme 2 est inapplicable. De tels systèmes sont peu fréquents, c'est le cas des systèmes complètement commandables [MUR77]. En effet dans l'automatique des systèmes continus, un système est complètement commandable si et seulement si:

$$\text{rang}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n \quad B \text{ et } A \text{ sont les matrices du système [III.4.1],}$$

cette condition nécessaire et suffisante transposée au système [III.4.2] nous permet de tirer la condition de commandabilité suivante:

$$\text{rang}[C \ CI \ CI^2 \ \dots \ CI^{n-1}] = \text{rang}[C] = n, \quad C \text{ est la matrice du système [III.4.2] et } I \text{ est la matrice identité,}$$

d'où le théorème suivant:

#### **Théorème 4.1:** [MUR77]

Pour qu'un RdP soit complètement commandable il faut que  $\text{rang}(C) = n$  •

Cette condition de commandabilité signifie qu'il existe une séquence de franchissement,  $\sigma$ , telle que le réseau puisse atteindre n'importe quel marquage,  $M$ , à partir d'un marquage initial quelconque  $M_0$ . Dans la pratique un système dont la matrice d'incidence  $C$  vérifie le théorème 4.1 ( $C$  de rang plein lignes) est un réseau qui n'a pas de P-semi-flot puisque le système homogène:

$$C^T \cdot x = 0$$

n'a pas de solution non nulle (voir chapitre précédent).

L'algorithme 2 est donc inutilisable pour les systèmes complètement commandables alors que l'algorithme 1 reste valable.

## 5. Conclusion

Lors de ce chapitre, nous nous sommes efforcés de donner une idée des fonctionnalités d'un système de surveillance. Certains modules comme le système de décision qui fait appel à des domaines tels que l'intelligence artificielle et les bases de données ont été volontairement négligés car ces domaines ne sont pas de la compétence de l'équipe de Longwy du C.R.A.N.. Quant aux choix des orientations prises, à savoir l'utilisation des réseaux de Petri, elles sont justifiées par la rapidité et la simplicité des algorithmes obtenus. Bien évidemment un tel système a ses faiblesses et ses limites mais les techniques proposées peuvent être utilisées avantageusement comme complément à un système déjà existant.

## 6. Bibliographie

[AUB91] J. F. AUBRY, C. ZANNE "Intégration de la sûreté de fonctionnement dans la conception des systèmes de commande des processus électromécaniques" A.P.I.I., Vol. 25, N° 4, pp. 297-324, 1991.

[AGA92] R. AGARWAL "A Petri Net based approach for verifying integrity of production systems" Int. J. Man-Machine Studies, Vol. 36, pp. 447-68, 1992.

- [**BAS92**] A. BASSONG ONANA, M. DAROUACH "Optimal filtering for singular systems" *Control Theory and Advanced Technology*, Vol. 8, pp. 731-742, 1992.
- [**BEA91**] S. F. BEAUMONT, S. GENTIL "Modèle qualitatif de comportement pour un système d'aide à la supervision" *R.A.I.R.O. APII*, N° 25, pp. 325-48, 1991.
- [**BEN91**] S. BEN AHMED, M. MOALLA, M. COURVOISIER, and R. VALETTE "Flexible manufacturing production system modelling using object Petri nets and their analysis" *IMACS 13th World Congress and IMACS conference on Modelling and Control of Technological Systems juillet 1991-mai 1991*, Dublin-Lille.
- [**BON93**] A. De BONNEVAL "Mécanismes de reprise dans les systèmes de commande à évènements discrets" *Thèse de Doctorat de l'université Paul SABATIER*, 27 sept. 1993.
- [**BOU91**] J.P. BOUREY, J.C. GENTINA "Conception structurée et modulaire d'architectures de contrôle réparti en production manufacturière" *R.A.I.R.O. APII* 1991, N° 25, pp. 349-76.
- [**BRA88**] K. P. BRAND, J. KOPAINSKY "Principles and engineering of process control with Petri Net" *IEEE Transactions on Automatic Control*, Vol. 33, N° 2, pp. 136-49, fev. 1988.
- [**BRU90**] J. BRUNET, D JAUME, M. LABARRERE, A. "Détection et diagnostic de pannes" *Editions Hermes Paris* 1990.
- [**CAS94**] J. CASSAR, R. LITWAK, V. COCQUEMPOT, M. STAROSWIECKI "Approche structurelle de la conception de systèmes de surveillance pour des procédés industriels complexes" *Diagnostic et sûreté de fonctionnement*. Vol. 4, N° 2, pp. 179-202, 1994.
- [**COM91**] M. COMBACAU "Commande et surveillance des systèmes à évènements discrets complexes: application aux ateliers flexibles" *Thèse de doctorat de l'université PAUL SABATIER*, 16 déc. 1991.

[CON91] L. CONSOLE, L. PORTINALE "Model-based diagnosis of system malfunction with Petri nets" IMACS 13th World Congress and IMACS conference on Modelling and Control of Technological Systems juillet 1991-mai 1991, Dublin-Lille.

[CON92] L. CONSOLE, L. PORTINALE "Model-based diagnosis of system malfunction with Petri Nets" Robotics and Flexible Manufacturing systems, IMACS 13th WorldCongress Jul. And May 1991, pp. 417-26, 1992.

[DAR93] M. DAROUACH, M. ZASADZINSKI, D. MEHDI "State estimation of stochastic singular linear systems" International Journal of Systems Science, Vol. 24, pp. 345-354, 1993.

[DAR94] M. DAROUACH, M. ZASADZINSKI, A. BASSONG ONANA "Connection between the three-block generalized Riccati equation and the standard Riccati equation" IEEE Transactions on Automatic Control, Vol. 39, pp. 1755-1758, 1994.

[GRE85] J. S. GREENSTEIN, S. LAM "An experimental study of dialogue-based communication for dynamic human-computer task allocation" International Journal of Man-Machine studies (1985), N° 23, pp. 605-21.

[HOR90] R. HORN, C. R. JOHNSON "Matrix Analysis" Cambridge Université Press 1990.

[HUR88] G. S. HURA, J. W. ATWOOD " The use of Petri nets to analyse coherent fault trees" IEEE Transactions on Reliability, Vol. 37, N° 5, Dec. 1988.

[JAL92] N. A. JALEL, H. NICHOLSON "The use of Petri Nets for fault diagnosis" Conference Canadienne et Exposition, Automatisation Industrielle, Vol. 2, pp. 32.1-4, 1-3 juin 1992.

[JUD94] R. P. JUDD, T. N. FAIZ "Optimal deadlock avoidance for a class of manufacturing systems using Petri nets" ACC-94, WM12-1:50, Baltimore, Maryland jun. 1994.

- 
- [LAP85] J.C. LAPRIE "Sûreté de fonctionnement des systèmes informatiques et tolérance aux fautes" Techniques de l'ingénieur, Traité Mesures et Contrôle, pp.R 7595 1-12.
- [MIL88] P. MILLOT "Supervision des procédés automatisés et ergonomie" Editions Hermes Paris 1988.
- [MUR77] T. MURATA "State equation, controllability and maximal matchings of Petri nets" IEEE Transactions on Automatic Control, pp. 412-16, jun. 1977.
- [NIK92] R. NIKOUKHAH, A.S. WILLSKY, B.C. LEVY "Kalman filtering and the Riccati equations for descriptors systems" IEEE Transactions on Automatic Control, Vol. 37, pp. 1325-1342, 1992.
- [PAT90] R. J. PATTON, P. M. FRANCK, R. N. CLARCK "Fault diagnosis in dynamic systems, theory and applications" Prentice-Hall Englewood Clmiffs, 1990.
- [PER91] D. PERDU, A. H. LEVIS "A Petri Net model for evaluation of expert systems in organizations" Automatica, Vol. 27, N° 2, pp. 2225-37, 1991.
- [PRO91] J. PROCK "A new technique for fault detection using Petri Nets" Automatica, Vol. 27, N° 2, pp. 239-45, 1991.
- [RAC92] T. RACHELL, D. G. MEYER "Fault detection and initial state verification by linear programming for a class of Petri nets" ACC/FM7, pp. 3191-95, 1992.
- [RAG90] J. RAGOT, M. DAROUACH, D. MAQUIN, G. BLOCH "Validation de données et diagnostic" Editions Hermes Paris 1990.
- [ROS93] E. ROSZKOWSKA, J. JENTINK "Minimal restrictive deadlock avoidance in FMS" ECC-93 Groningen.

- [**ROU92**] G. ROUX, B. DESCOTES-GENON, P. LADET "Operations checking in flexible manufacturing systems". Robotics and Flexible Manufacturing systems, IMACS 13th World Congress Jul. And May 1991, pp. 305-14, 1992
- [**SAH92**] A. SAHRAOUI "Vers une approche globale de surveillance des systèmes à évènements discrets" APII, Vol. 26, N° 2, pp 91-106, 1992.
- [**SRI93**] V. S. SRINIVASAN, M. JAFARI "Fault detection/monitoring using Petri nets" IEEE Transactions on Systems, Man and Cybernetics , Vol. 23, N° 4, pp. 1155-62, jun. 1992.
- [**TER93**] E. TERUEL, J. M. COLOM, M. SILVA "Linear analysis of deadlock-freeness of Petri net models" ECC-93 Groningen.
- [**USH89**] T. USHIO "On controllability of controlled Petri Nets" Control theory and Advanced Technologie, Vol. 5, N° 3, pp. 265-275, sept. 1989.
- [**VAL93**] C. VALENTIN, P. LADET "Flow modelling in a class of hybrid systems" IEEE SMC-93 Le Touquet.
- [**WIE88**] H WIELE, G. LIERMANN, H. RASCHEL "Monitoring, diagnostic and operator leading systems for flexible automated manufacturing" Computers in Industry 10 (1988) pp. 61-88.
- [**WIL90**] R. G. WILLSON, B. H. KROGH "Petri Net tools for the specification and analysis of discrete controllers" IEEE Transactions on software engineering, Vol. 16, N° 1, pp. 39-50, jan. 1990.
- [**YEN92**] H. YEN "Applications of conflict-free Petri Nets to parallel programs and asynchronous circuits" IPCCC'92, CH129-4/92/000-0766, IEEE 1992.
- [**ZAR90**] M.R. ZARGHAM. K.J. DANHOF "Toward definition of fault analysis for Petri Nets models" Information Processing Letters N° 34 (1990), pp. 299-305.

[ZH092] M. ZHOU, F. DICESARE, A. A. DESROCHERS "A hybrid methodology for synthesis of Petri Net models for manufacturing systems" IEEE Transactions on Automatic Control, Vol. 8, N° 3, pp. 350-61, jun. 1992.

[ZH094] Y. ZHONGHU, X. WENDONG, X. XINHE "Deadlock avoidance in manufacturing systems of parallel mutual exclusion" ACC-94, WM12-2:50, Baltimore, Maryland jun. 1994.



THESE de Abdellatif BOURJIJ

La page 104

est manquante dans l'original.

## CHAPITRE IV

### APPLICATIONS INDUSTRIELLES

## **Présentation**

Ce chapitre est consacré à la présentation d'applications développées dans le cadre de contrats de l'Equipe d'Automatique et de Recherche Appliquée de Longwy (E.A.R.A.L.) du Centre de Recherche en Automatique de Nancy (C.R.A.N.) avec des entreprises ou des laboratoires de recherche étrangers. A partir de deux applications à caractères différents, nous voulons montrer l'intérêt que peut présenter l'utilisation des Réseaux de Petri dans la modélisation et le diagnostic des processus industriels.

Le premier cas étudié:

### **un réseau de générateurs de vapeur simulé**

est dû à un projet de coopération Franco-Québécoise entre l'E.A.R.A.L. et le Centre de Production Automatisée de Jonquière. Le thème de ce projet est: Technologie de Production Assistée par Ordinateur (T.P.A.O.) pour la commande de systèmes à base de multiprocesseurs orientés vers des applications industrielles. Dans le cadre de cette coopération, un premier modèle de simulateur a été étudié par M. Zasadzinski au Québec, modèle repris localement et développé grâce aux conseils avisés de M. Feidt, pour donner l'outil présenté dans ce mémoire.

Le deuxième cas:

### **un quantomètre**

fait l'objet d'un contrat entre l'E.A.R.A.L., les sociétés LENOIR AUTOMATISMES implantée à Longwy et FONDERIES ET ACIERIES DU MAROC implantée à Casablanca. Cet appareil d'analyse des aciers, bien qu'utilisant un principe toujours d'actualité, est construit avec une technologie vieille de 25 ans. Le but du contrat est de rénover l'appareil hors service, de l'instrumenter et de concevoir et développer un système de commande à base de micro-ordinateur.

## Ière Partie Le Simulateur

### 1. Introduction

Le simulateur est un logiciel ayant nécessité de nombreux développements informatiques inutiles à la compréhension de ce chapitre, aussi, nous ne détaillerons que les aspects indispensables. Les RdP ont été utilisés dans la conception mais aussi la modélisation du simulateur, afin d'en garantir le bon fonctionnement. Ce simulateur a pour finalité, la conception et la validation d'algorithmes de commande. Aussi est-il doté d'une architecture modulaire, intégrant un nombre variable (laissé au choix de l'utilisateur) de bouilloires alimentant un réseau. Le choix d'un tel processus chimique se justifie par l'importance de son implantation dans le milieu industriel. La vapeur d'eau surchauffée est utilisée dans l'industrie comme source d'énergie [COR84] (elle est par exemple utilisée dans les usines de cellulose pour sécher le papier). La vaporisation de l'eau est un procédé assez simple pour produire une énergie facilement transportable mais il demeure assez complexe à modéliser et à réguler. De plus, les différents modules constituant le simulateur ont des échelles de temps très disparates. Les bouilloires, simulées par un modèle mathématique, ont une évolution continue dans le temps. Par contre, les modules de communication (lecteurs et rédacteurs présentés § 3.2) et le module de commande sont discrets. Les générateurs de vapeur peuvent être décrits par les RdPC alors que les modules de communication et de commande sont décrits par les RdP discrets. L'ensemble peut être représenté dans un même graphe par les RdPH [JAC93]. Une telle représentation a permis la conception et l'analyse de l'architecture du simulateur [BOU93] et fournit un modèle de référence pour le diagnostic.

### 2. Architecture du simulateur

#### 2.1. Modèle mathématique d'une bouilloire

Dans ce paragraphe, un modèle mathématique du fonctionnement d'une bouilloire est proposé. La figure suivante donne une idée du fonctionnement d'une bouilloire avec ses variables d'entrées et de sorties. Schématiquement, pour produire de la vapeur, il faut alimenter une bouilloire avec un débit d'eau ( $\dot{m}_1$ ) et lui fournir un débit de chaleur ( $\dot{q}$ ). A

l'ébullition, on obtient un débit de vapeur ( $\dot{m}_v$ ) et il reste un niveau d'eau (level). L'objectif de la commande est de garantir un débit massique de vapeur désiré, constant tout en stabilisant le niveau d'eau dans la cuve.

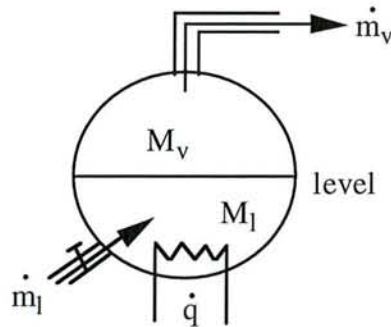


Figure I.2.1 Une bouilloire

Dans ce qui suit nous présentons les expressions qui caractérisent le fonctionnement d'une bouilloire. La masse totale  $M$  de matière dans une bouilloire est constituée de la masse d'eau  $M_l$  et de la masse de vapeur  $M_v$ , soit:

$$M = M_l + M_v \quad [\text{IV.2.1}]$$

La variation de masse  $\dot{M}$  est donnée par la différence entre le débit massique d'eau ( $\dot{m}_l$ ) à la température du réseau (on parle ici du réseau d'alimentation en eau) et le débit massique de vapeur ( $\dot{m}_v$ ):

$$\dot{M} = \dot{m}_l - \dot{m}_v \quad [\text{IV.2.2}]$$

A partir des deux équations [IV.2.1] et [IV.2.2], on obtient le bilan de matière suivant:

$$\dot{M}_l + \dot{M}_v = \dot{m}_l - \dot{m}_v \quad [\text{IV.2.3}]$$

L'expression du bilan énergétique est:

$$\dot{m}_l h_{lr} - \dot{m}_v h_v + \dot{q} - \dot{q}_B = 0 \quad [\text{IV.2.4}]$$

où  $h_{lr}$ ,  $h_v$  et  $q_B$  représentent les enthalpies spécifiques de l'eau du réseau d'alimentation et de la vapeur, ainsi que l'énergie interne de la bouilloire. L'enthalpie spécifique de l'eau du réseau  $h_{lr}$  peut être approximée par le polynôme suivant [ROH87]:

$$h_{lr} = A T_r^2 + B T_r \quad \text{avec} \quad A = 3.16 * 10^{-4} \quad \text{et} \quad B = 4.19$$

où  $T_r$  est la température dans le réseau d'alimentation en eau. L'enthalpie spécifique de l'eau dans la bouilloire,  $h_l$ , prend l'expression suivante en fonction de la température  $T_{sat}$  ( $T_{sat}$  = température de la vapeur saturée dans la bouilloire), pour une plage de variation entre 0.01°C à 200°C, (incertitude relative maximale à 0.8%):

$$h_v = h_l + L_v \quad \text{où} \quad L_v \quad \text{est la chaleur latente}$$

$$L_v = A T_{sat}^5 + B T_{sat}^4 + C T_{sat}^3 + D T_{sat}^2 + E T_{sat} + F$$

avec  $A = -0.69851352 * 10^{-9}$ ,  $B = 0.47175172 * 10^{-6}$ ,  $C = -0.12963934 * 10^{-3}$ ,  
 $D = 0.12413792 * 10^{-1}$ ,  $E = 0.27913724 * 10^1$  et  $F = 0.25037 * 10^4$

L'enthalpie de l'ensemble liquide-vapeur dans la bouilloire,  $q_B$ , est:

$$q_B = M_l h_l + M_v h_v \Rightarrow \dot{q}_B = \dot{M}_l h_l + \dot{M}_v h_v \quad \text{[IV.2.5]}$$

$\rho_l$  et  $\rho_v$  sont respectivement les masses volumiques de l'eau et de la vapeur à  $T_{sat}$  et sont données par les relations suivantes:

$$\rho_l = A T_{sat}^4 + B T_{sat}^3 + C T_{sat}^2 + D T_{sat} + E$$

avec  $A = -0.4879268 * 10^{-7}$ ,  $B = 0.27819864 * 10^{-4}$ ,  $C = -0.7525276 * 10^{-2}$ ,  
 $D = 0.111546$  et  $E = 999.085$ .

$$\log_{10}(\rho_v) = A T_{sat}^5 + B T_{sat}^4 + C T_{sat}^3 + D T_{sat}^2 + E T_{sat} + F$$

avec  $A = -0.8254406 * 10^{-12}$ ,  $B = 0.8362999 * 10^{-9}$ ,  $C = -0.39891116 * 10^{-6}$ ,  
 $D = 0.122079 * 10^{-3}$ ,  $E = 0.2987838 * 10^{-1}$  et  $F = -2.3142974$ .

Le volume total de la bouilloire est:

$$V_{\text{tot}} = \frac{M_l}{\rho_l} + \frac{M_v}{\rho_v}$$

La masse de vapeur  $M_v$  peut être exprimée en fonction du volume total  $V_{\text{tot}}$  et de la masse d'eau:

$$M_v = \rho_v V_{\text{tot}} - \frac{\rho_v}{\rho_l} M_l$$

En dérivant l'expression ci-dessus,  $V_{\text{tot}}$  étant constant, on obtient la relation [IV.2.6]:

$$\dot{M}_v = -\frac{\rho_v}{\rho_l} \dot{M}_l \quad \text{[IV.2.6]}$$

L'équation [IV.2.5] remplacée dans [IV.2.6] donne:

$$\dot{m}_l h_{lr} - \dot{m}_v h_v + \dot{M}_l h_l + \dot{M}_v h_v + \dot{q} = 0$$

La variable  $\dot{m}_v$  remplacée par sa valeur dans [IV.2.3] et  $\dot{M}_v$  par sa valeur dans [IV.2.6], permettent de déduire une équation où la seule inconnue est  $\dot{M}_l$ :

$$\dot{m}_l h_{lr} - \dot{m}_l h_v + \dot{M}_l h_v - \dot{M}_l h_l + \dot{q} = 0 \quad \text{donc} \quad \dot{M}_l = \frac{(h_{lr} - h_v) \dot{m}_l + \dot{q}}{h_l - h_v}$$

Finalement, à partir des équations [IV.2.6] et [IV.2.3], le débit massique de vapeur est exprimé en fonction du débit massique d'eau et de la variation de la masse d'eau:

$$\dot{m}_v = \dot{m}_l + \left( \frac{\rho_v}{\rho_l} - 1 \right) \dot{M}_l$$

Grâce à ce modèle mathématique, on est en mesure de calculer les sorties ( $\dot{m}_v$  et level) d'une bouilloire connaissant les entrées ( $\dot{m}_1$  et  $\dot{q}$ ). On peut donc énoncer un algorithme réalisant ce traitement itératif pour simuler ce processus continu. Dans l'algorithme proposé ci-dessous, la variable "scan" (en unité de temps) est introduite pour ajuster le simulateur aux réalités physiques d'un générateur de vapeur. En effet, le temps de calcul de la variation de la masse d'eau,  $\dot{M}_1$ , va dépendre de la rapidité du processeur utilisé, aussi pour ramener ce temps à la valeur d'évolution du processus simulé, nous avons introduit une variable sur laquelle peut intervenir l'utilisateur.

**Algorithme A:**

Initialisations;

**Tant que** (vrai) **faire**lire ( $\dot{m}_1$ ,  $\dot{q}$ );

$$\dot{M}_1 = \frac{(h_{lr} - h_v)\dot{m}_1 + \dot{q}}{h_1 - h_v};$$

$$M_1 = M_1 + \text{scan} \cdot \dot{M}_1;$$

$$\dot{m}_v = \dot{m}_1 + \left(\frac{\rho_v}{\rho_1} - 1\right)\dot{M}_1;$$

$$V_1 = \frac{M_1}{\rho_1};$$

$$\text{level} = \frac{100 * V_1}{V_{\text{tot}}};$$

écrire (level,  $\dot{m}_v$ );**Fintq****2.2. Le réseau de génération de vapeur**

Pour des raisons de sûreté et de puissance, plusieurs bouilloires sont utilisées en parallèle pour alimenter le réseau. Chaque bouilloire est simulée par un programme



développé en langage C et dont le modèle mathématique a été présenté précédemment. Les différentes bouilloires sont indépendantes, elles prennent leurs entrées et leurs consignes dans une zone de mémoire partagée et y déposent, après calcul, leurs sorties. Un programme en langage C joue le rôle du réseau, il lit dans la mémoire partagée les sorties des différentes bouilloires et calcule le débit global du réseau qui alimente l'usine. Un tel système permet à l'utilisateur de choisir la configuration qu'il désire commander et donc d'en ajuster la difficulté.

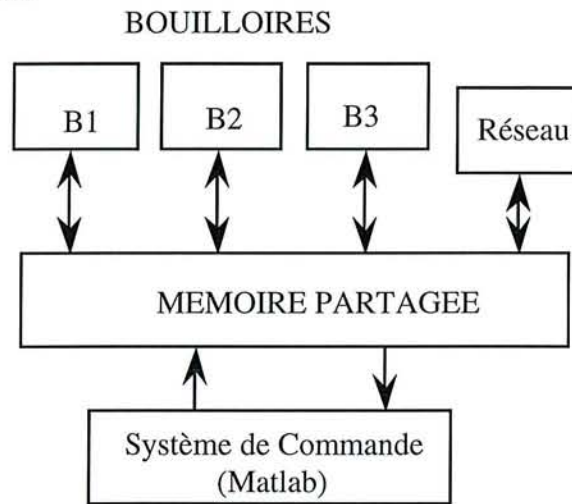


Figure I.2.2 Le simulateur

Les bouilloires et le système de commande sont implantés sur une station SUN alors que le logiciel de supervision l'est sur un DPX2000, l'ensemble est relié par un réseau local (ETHERNET).

### 2.3. Le système de commande

Ce simulateur a été réalisé pour fournir à l'équipe "commande" du laboratoire un outil de test et d'évaluation pour les algorithmes de commande. Cependant, une contrainte essentielle des utilisateurs est qu'ils souhaitent utiliser Matlab pour leurs développements. Il a donc fallu intégrer ce logiciel dans l'environnement du simulateur en créant des routines de communication entre ces deux éléments. Le système d'exploitation de la station SUN étant un dérivé d'UNIX (SOLARIS), une spécificité de ce système a été exploitée: les échanges par tubes (pipes), pour permettre une communication rapide. Le programme de

commande doit acquérir périodiquement les consignes de l'utilisateur et les sorties des bouilloires, en fonction desquelles, il calcule les entrées des bouilloires.

Une bouilloire est un système multivariable comprenant:

deux entrées: le débit massique d'eau et l'apport de chaleur ( $\dot{m}_l$  et  $\dot{q}$ )

et deux sorties: le débit massique de vapeur et le niveau d'eau ( $\dot{m}_v$  et level).

L'équation liant les entrées aux sorties est la suivante:

$$\begin{bmatrix} \text{Level} \\ \dot{m}_v \end{bmatrix} = \begin{bmatrix} F_1(s) & F_2(s) \\ F_3(s) & F_4(s) \end{bmatrix} \begin{bmatrix} \dot{m}_l \\ \dot{q} \end{bmatrix}$$

avec:

$$F_1(s) = \frac{c_1 b_1}{s}, \quad F_2(s) = \frac{c_1 b_2}{s}, \quad F_3(s) = b_3 \quad \text{et} \quad F_4(s) = b_4. \quad b_1 = \left( \frac{h_v - h_{lr}}{h_v - h_l} \right), \quad b_2 = \left( \frac{-1}{h_v - h_l} \right),$$

$$b_3 = \left[ 1 + \left( \frac{h_v - h_{lr}}{h_v - h_l} \right) \left( \frac{\rho_v}{\rho_l} - 1 \right) \right], \quad b_4 = \left[ \left( 1 - \frac{\rho_v}{\rho_l} \right) \left( \frac{1}{h_v - h_l} \right) \right] \quad \text{et} \quad c_1 = \left( \frac{100}{V_{\text{tot}} \cdot \rho_l} \right).$$

## 2.4. La supervision

Le simulateur est implanté sur une station de travail SUN (sous UNIX) équipée notamment d'un compilateur C pour la création des bouilloires et des modules de communication et d'un logiciel de calculs mathématiques, Matlab, pour la conception d'algorithmes de commande. Le logiciel de supervision (DEXTERITY) est installé sur un DPX2000 (BULL) et permet de visualiser en temps réel l'état du système avec en particulier le niveau et le débit vapeur des bouilloires. Les données de l'ensemble des modules du simulateur, étant stockées dans une zone de mémoire partagée sur la station SUN, elles sont aisément copiables, par des routines dédiées, sur une mémoire tampon du DPX2000 via un réseau local (Ethernet).

Le rôle actuel de la supervision se limite à une visualisation de l'état du simulateur:

- affichage des bouilloires et de leurs entrées et sorties,
- affichage de l'état du réseau,
- affichage des alarmes.

Cependant, une extension des attributions de la supervision est prévue avec, en particulier, la possibilité de choisir graphiquement la structure du réseau et de pouvoir agir directement sur les actionneurs (% d'ouverture des électrovannes, brûleurs, ...).

### 3. Sûreté de fonctionnement

#### 3.1. Modélisation

Le simulateur étant modulaire, l'utilisateur peut choisir le nombre de bouilloires qu'il désire contrôler. Un système avec une seule bouilloire peut être représenté ainsi:

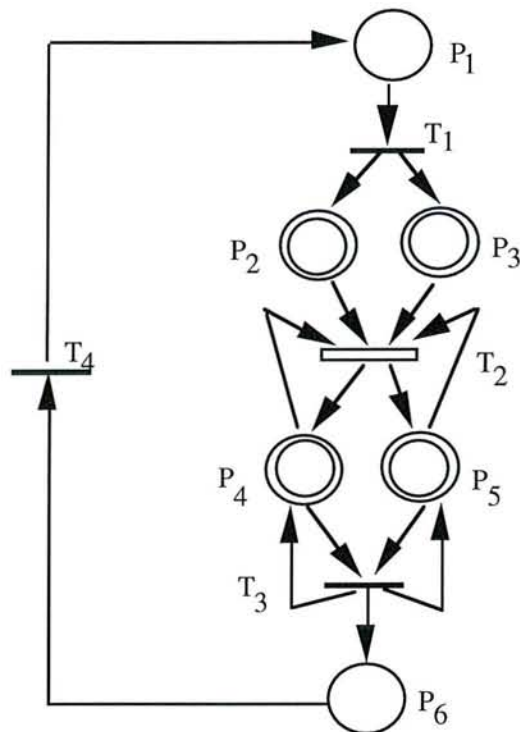


Figure I.3.1 RdP Hybride

Transitions:

- $T_1$ : Tâche lecteur,
- $T_2$ : Tâche production de vapeur (bouilloire),
- $T_3$ : Tâche rédacteur,
- $T_4$ : Tâche de commande.

Une temporisation  $d_i$  ( $i=1,3,4$ ) est associée aux D-transitions et une vitesse de franchissement est associée à la C-transition  $T_2$ .

Places:

- $P_1$ : lecture,
- $P_2$ : donnée d'entrée  $\dot{m}_1$ ,

- $P_3$ : donnée d'entrée  $\dot{q}$ ,  
 $P_4$ : donnée de sortie  $\dot{m}_v$ ,  
 $P_5$ : donnée de sortie level,  
 $P_6$ : écriture.

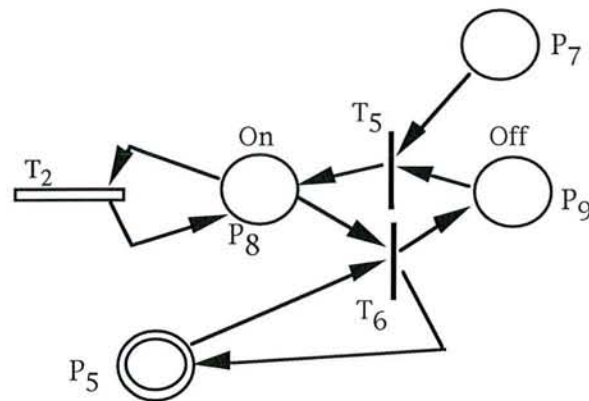
Physiquement, les mémoires tampon  $P_1$  et  $P_2$  sont des mémoires partagées. Le système de commande a besoin des données  $\dot{m}_v$  et level pour calculer  $\dot{m}_l$  et  $\dot{q}$ , tandis que la bouilloire ( $T_2$ ) a besoin des entrées  $\dot{m}_l$  et  $\dot{q}$  pour produire de la vapeur ( $\dot{m}_v$ ) et déterminer le niveau d'eau dans la bouilloire (level). Le transfert des données est réalisé par les tâches  $T_1$  et  $T_3$ . Initialement, la place  $P_1$  contient un jeton. Le poids des arcs, liant les transitions  $T_1$  et  $T_2$  aux C-places, est paramétré, il est réel et déterminé par le système de commande à chaque période d'échantillonnage. La C-transition  $T_2$  est validée après le premier tir de  $T_1$  et continuellement tirée: elle simule la production de vapeur. Le système étant étudié dans son régime permanent, le marquage initial est le suivant:

$$M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \dot{m}_v \\ \text{level} \\ 0 \end{pmatrix}$$

La matrice d'incidence correspondant au réseau du RdPH de la figure I.3.1 est donnée par :

$$C_i = \begin{pmatrix} -1 & 0 & 0 & 1 \\ \dot{m}_{li} & -\Delta\dot{m}_{li} & 0 & 0 \\ \dot{q}_i & -\Delta\dot{q}_i & 0 & 0 \\ 0 & \Delta\dot{m}_{vi} & 0 & 0 \\ 0 & \Delta\text{Level}_i & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \quad [\text{IV.3.1}]$$

Ce réseau peut être enrichi d'un mécanisme hybride, d'arrêt de la production en cas d'urgence, représenté ainsi:



**Figure I.3.2** Arrêt d'urgence

La transition  $T_2$  et la place  $P_5$  étant les mêmes que celles de la figure 2.3. La D-transition  $T_6$  est validée si  $30 \geq \text{level} \geq 70\%$ , son franchissement provoque le retrait d'une marque de la place  $P_8$  et donc l'arrêt de la production.

La figure I.3.3 représente un modèle de référence global constitué de trois bouilloires, un réseau chargé de recueillir la production des trois générateurs et un système discret de gestion des accès en mémoire.

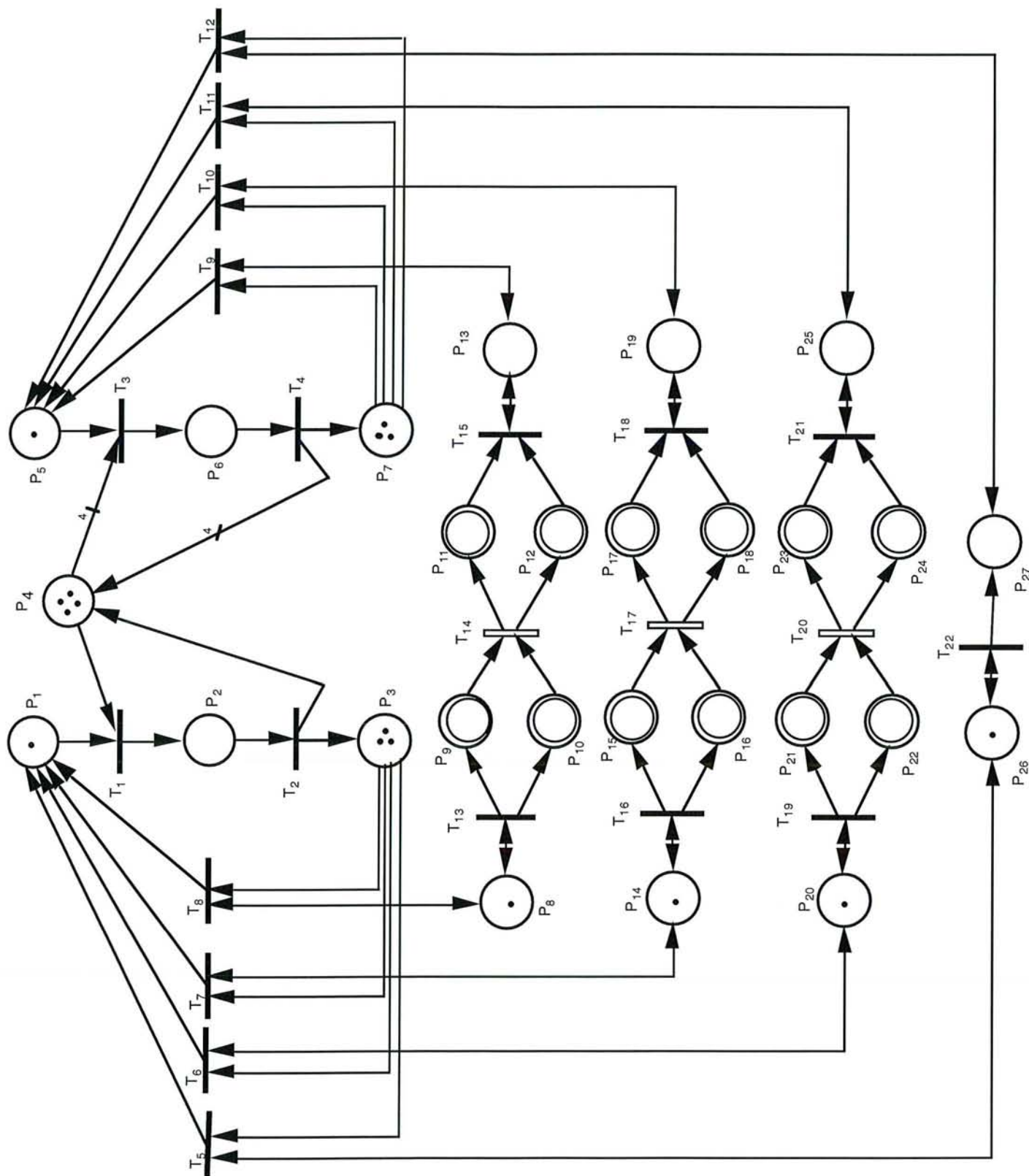


Figure I.3.3 *Modèle global*

### 3.2. Surveillance

Le modèle hybride (Fig. I.3.3) ainsi constitué présente plusieurs avantages [BOU94].

Dans la partie discrète, on reconnaît le problème classique du lecteur/écrivain. Les échanges sont réalisés à l'aide de quatre tâches écrivains et quatre tâches lecteurs. Mais, il ne peut y avoir de communication saine sans gestion des conflits. Pour éviter l'accès à des résultats erronés, on ne doit autoriser qu'une seule écriture à la fois et aucune lecture pendant l'écriture. Regardons donc comment le calcul des invariants permet de s'assurer que ces propriétés sont bien vérifiées

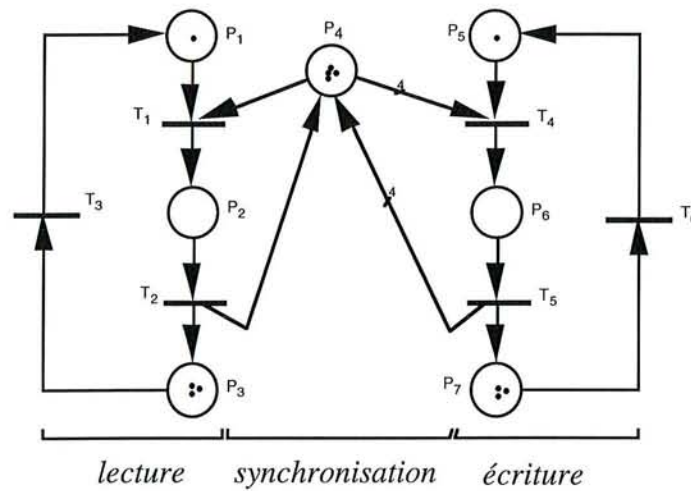


Figure I.3.4. Partie discrète

Les places représentent les états accessibles par le processus lecteurs/rédacteurs et les transitions les événements permettant le passage d'un état à l'autre. Ainsi la présence d'une marque dans:

la place  $P_1$  indique qu'une tâche est prête à lire dans le tampon,

la place  $P_2$  indique qu'une tâche est en cours de lecture dans le tampon,

la place  $P_3$  indique qu'une tâche est en attente pour lire dans le tampon,

la place  $P_5$  indique qu'une tâche est prête à écrire dans le tampon,

la place  $P_6$  indique qu'une tâche est en cours d'écriture dans le tampon,

la place  $P_7$  indique qu'une tâche est en attente pour écrire dans le tampon,

la place  $P_4$  sert à synchroniser les accès dans la mémoire partagée de manière à empêcher les états interdits décrits précédemment.

Le calcul des invariants par les méthodes exposées lors du chapitre II permettent d'exhiber les solutions suivantes:

**T-semi-flots:**

$$S = \left\{ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \right\}$$

Ces deux solutions forment une base génératrice minimale de l'ensemble des solutions à ce problème.  $S_1$  (resp.  $S_2$ ) montre que le franchissement de la séquence " $T_1T_2T_3$ " (resp. " $T_4T_5T_6$ ") permet le retour au marquage initial et donc que le système est capable de se réinitialiser en accomplissant les deux tâches prévues (lecture ou écriture).

**P-semi-flots:**

$$F = \left\{ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \right\}$$

Les éléments de l'ensemble F forment une famille génératrice minimale des P-semi-flots du RdP de la figure I.3.4., et toute solution est une combinaison linéaire de ces éléments. Remarquons que nous obtenons trois solutions qui correspondent chacune à un circuit qui est un bloc fonctionnel du processus:

lecture	$M(P_1) + M(P_2) + M(P_3) = K_1,$	[II.7.1]
écriture	$M(P_5) + M(P_6) + M(P_7) = K_2,$	
synchronisation	$M(P_2) + M(P_4) + 4.M(P_6) = K_3$	

où  $K_1, K_2$  sont des constantes (respectivement le nombre de tâches écrivains et le nombre de tâches lecteurs, ici  $K_1 = K_2 = K_3 = 4$ ). Ceci nous permet de vérifier que le modèle colle bien au processus réel et que la synchronisation est correctement réalisée. Lors d'un fonctionnement normal du processus, il sera possible à tout moment de vérifier que les états atteints (marquages) sont cohérents et donc vérifient les égalités précédentes. De plus comme chaque invariant correspond à une fonction on peut localiser les défauts .



Enfin en application du théorème 7.1 (chap. II), on peut montrer que ce réseau est borné, en particulier les places P2 et P6 et retrouver quelques propriétés essentielles que doit vérifier le modèle:

$M(P_6) \leq 1$  et donc à un instant donné, il n'y aura qu'une seule écriture possible, si  $M(P_6) = 1$  alors  $M(P_4) = 0$  ainsi aucun autre accès au tampon mémoire n'est autorisé,

$M(P_2) \leq 4$  et donc le nombre de lectures simultanées est limité à 4,

si  $M(P_2) \geq 1$  alors  $M(P_4) < 4$  ainsi toute écriture est interdite.

Cette gestion des accès en mémoire partagée est bien plus sûre que celle dont nous disposons dans une version antérieure, basée sur les sémaphores (qui ne constituent qu'un distributeur de tickets). Le réseau étant vivant, on évite les risques de blocage et il suffit de donner la même priorité aux tâches pour éviter les famines. Le modèle proposé étant temporisé, le calcul des temps d'indisponibilité des marques (places de synchronisation lecture/écriture) est alors effectué pour simuler les périodes d'échantillonnage choisies par l'utilisateur et ce grâce aux relations [I.6.3] et [I.6.4].

Un tel système nous permet donc de résoudre un premier problème purement informatique et d'apporter une sûreté d'échange des données. Ceci est d'autant plus important que les processus lecteurs constituent les actionneurs du simulateur et les processus écrivains les capteurs. En effet, les écrivains permettent de mémoriser les valeurs des sorties  $m_{vi}$  et  $level_i$  (débit de vapeur et niveau) de la bouilloire  $i$  (ils apportent des informations sur le processus), alors que les lecteurs permettent de fournir à chaque bouilloire  $i$  ses entrées  $m_{li}$  et  $q_i$  (apport d'eau et de chaleur).

La partie continue du RdP (Fig. I.3.3) permet de calculer en temps réel, les entrées et les sorties des bouilloires et de disposer d'un modèle de référence comme il a été défini lors du chapitre III. Pour le calcul de ces entrées/sorties, nous utilisons une matrice d'incidence paramétrique,  $C_i$  [IV.3.1], dont les valeurs sont données par le modèle mathématique de l'algorithme A. L'étude du réseau de génération de vapeur est réalisée dans sa phase stationnaire, le modèle utilisé supposant un fonctionnement à température et pression constantes.

En application de la relation fondamentale de calcul de l'évolution du marquage des RdPH [I.6.5], le modèle RdP permet de disposer en temps réel d'une référence pour le diagnostic. Notons que les vitesses de franchissement des transitions continues jouent le même rôle que la variable "scan" pour le simulateur.

On peut alors injecter des défauts sur le processus (simulateur) pour en étudier le comportement et appliquer l'algorithme 1 du chapitre III de détection de défauts. Un défaut de capteur est facilement simulable, il suffit de faire lire ou écrire une information erronée par l'un des processus lecteur ou écrivain. Les défauts du système peuvent quant à eux être directement injectés sur les bouilloires (le cas le plus fréquent sur ce type d'installation étant les fuites).

## IIème Partie Le quantomètre

### 1. Introduction

Le quantomètre est un appareil de mesure qui permet de déterminer la composition atomique d'échantillons de matière. Il fournit les concentrations en éléments chimiques déterminés et pour des métaux et alliages programmés. Le modèle étudié, "série 34000 de ARL", est un quantomètre simultané, c'est-à-dire qu'il est capable de mesurer simultanément l'intensité de plusieurs lignes spectrales dans la lumière émise par l'échantillon. Il est couramment utilisé par les fondeurs pour vérifier la composition des alliages qu'il fabriquent.

### 2. Principe de fonctionnement

Lorsqu'on fournit à un atome une certaine énergie, certains de ses électrons changent d'orbite. En regagnant leur orbite d'origine, ces électrons restituent une énergie bien précise sous forme de lumière dont la longueur d'onde est propre à chaque atome.

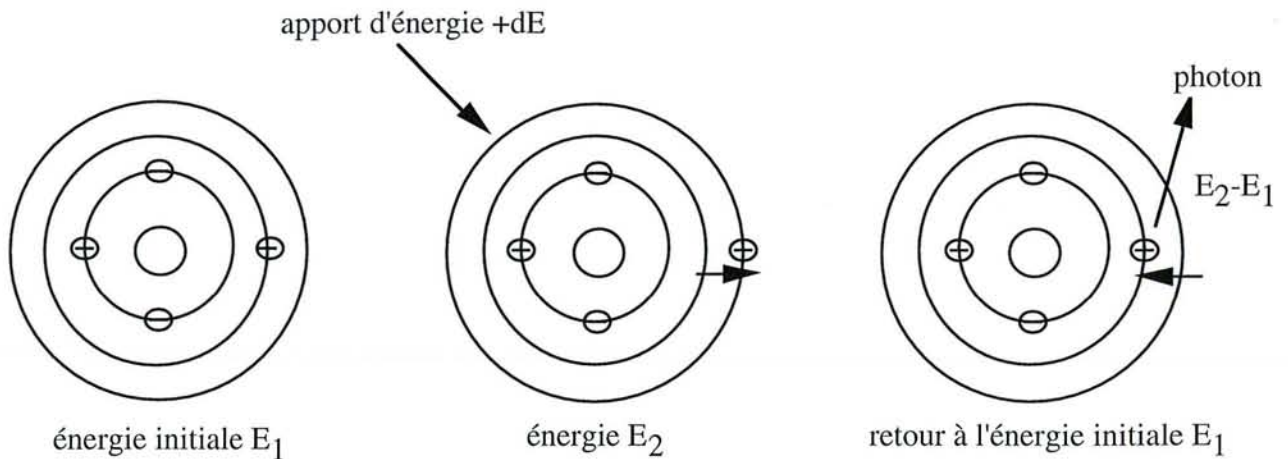


Figure II.2.1 Représentation schématique de l'excitation

Ce phénomène étant atomique, il n'est quasiment pas affecté par la forme chimique sous laquelle se présentent les atomes. Un appareil basé sur ce principe peut donc déterminer les

quantités de chaque atome dans l'échantillon, mais pas la forme sous laquelle il se trouve. Pour exploiter ce principe, le quantomètre provoque une étincelle (arc électrique) à la surface de l'échantillon pour lui apporter de l'énergie. Les rayons lumineux dus à l'énergie restituée par les électrons rejoignant leur orbite sont réfléchis sur des photomultiplicateurs associés à chaque type d'atome dont on veut connaître la teneur dans l'échantillon. Des courbes réalisées à l'aide d'étalons permettent de réaliser la conversion des tensions issues des photomultiplicateurs en concentrations des éléments recherchés dans l'échantillon.

### 3. Structure de l'instrument

Il est composé de trois éléments de base:

- la source et le statif (excitation)
- le spectromètre (optique)
- l'unité de commande (ordinateur et cartes d'acquisitions)

#### 3.1. La source et le statif

Cette unité fournit l'énergie nécessaire pour la vaporisation et l'excitation des atomes de l'échantillon. Elle est aussi appelée unisource car il n'y a que des décharges unidirectionnelles entre la cathode (échantillon) et l'anode (électrode). Pour permettre différentes configurations de préintégration et d'intégration, l'unisource est programmable depuis l'ordinateur (différents niveaux de tensions sont possibles: 300V, 500V,...).

Lors d'une analyse, on crée deux types d'étincelles. Pendant 15 secondes on provoque une étincelle de haute énergie destinée à préparer (homogénéiser) la surface de l'échantillon. Puis on crée une deuxième étincelle de moindre énergie durant 5 secondes, dont la lumière est analysée. Un obturateur (actionneur) permet d'éviter l'éblouissement des phototubes pendant la première étincelle.

Pour éviter toute interaction entre l'atmosphère et la surface de l'échantillon, l'enceinte contenant l'échantillon baigne dans une atmosphère d'argon (appelée statif d'argon). Un mince flux d'argon parcourt le statif en l'absence de toute analyse. Pendant l'étincelage, ce flux est augmenté automatiquement. La porte d'accès au statif est munie d'un interrupteur permettant d'éviter l'étincelage dans le cas où celle-ci serait ouverte.

### 3.2. Le spectromètre

C'est l'enceinte mécanique qui renferme le système optique. Pour garantir les performances analytique de l'appareil, la géométrie de ce système doit être stable. Aussi, pour éviter les déformations dues à la dilatation, le caisson est thermostaté. D'autre part comme l'air absorbe les rayons ultraviolets, le caisson est placé sous vide (pression de l'ordre de  $50 \mu\text{m Hg}$ ).

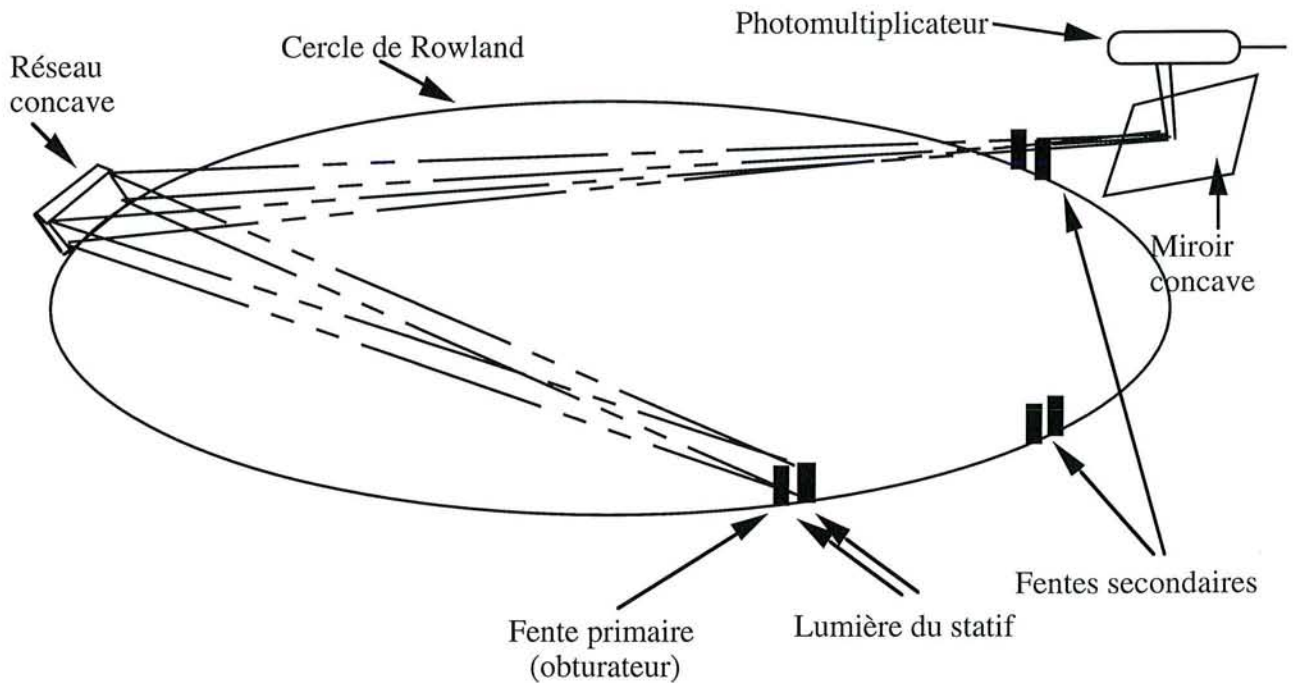


Figure II.3.1 Le spectromètre

La lumière issue du statif est diffractée par un miroir cylindrique (le réseau) vers les fentes secondaires qui ne laissent passer qu'une étroite bande de longueur d'onde. Les rayons qui traversent cette dernière fente sont focalisés par un miroir sur le photomultiplicateur. Enfin, le photomultiplicateur converti l'énergie lumineuse en courant qui sera stocké dans un intégrateur et servira au calcul des concentrations en éléments recherchés. Le spectromètre dispose d'un photomultiplicateur propre à chaque élément qu'il est capable d'analyser (le 34000 de FAM dispose de 17 photomultiplicateurs).

### 3.3. L'unité de commande

L'unité de commande est constituée d'un micro-ordinateur compatible PC (processeur i486 DX2, 4 Mo de RAM, disque dur 240 Mo), équipé de deux cartes d'acquisitions (16 entrées analogiques +/- 32mV à 10V, 2 sorties analogiques 0-5V ou 10V, 16 entrées et 16 sorties numériques TTL). Elle assure principalement trois tâches distinctes:

- la commande de l'appareil,
- le calcul des concentrations,
- la maintenance (logicielle et matérielle).

Le logiciel de contrôle, développé en langage C, remplace toutes les cartes de commande et diagnostic prévues initialement par le constructeur. Dès la mise sous tension de l'appareil, un diagnostic est réalisé puis affiché à l'écran. Les paramètres testés sont les suivants:

- la température de la cabine,
- le vide,
- les basses tensions,
- les hautes tensions.

Ce diagnostic est préalable aux analyses et conditionne la suite des opérations, c'est-à-dire qu'en cas de défaillance importante constatée sur l'un de ces paramètres, les analyses sont interdites avant le retour à un état normal.

## 4. La commande et le calcul des concentrations

Lors d'une analyse, le micro-ordinateur doit coordonner une série d'ordres aux actionneurs. Cette séquence dure quelques secondes mais constitue la charpente de toute l'analyse, c'est donc vers elle que seront dirigés tous les efforts de sécurisation. Une analyse peut être décomposée en trois phases:

- L'électrovanne d'argon est ouverte (flushing time, 5 secondes) puis l'échantillon est bombardé une première fois à haute tension (pré-intégration, 500Volts) pendant quinze secondes pour en homogénéiser la surface. Avant cette attaque, l'obturateur (de la fente primaire) a été préalablement fermé pour empêcher la lumière restituée par l'échantillon de pénétrer dans l'enceinte de l'optique.

- L'obturateur est ouvert et une deuxième étincelle, tout de suite après la première mais à moindre énergie (intégration, 300Volts) et d'une durée de cinq secondes, est utilisée pour l'analyse proprement dite. Les rayons lumineux sont diffractés dans l'enceinte du spectromètre vers les différents phototubes qui eux chargeront leur intégrateur respectif.

- L'électrovanne d'argon est fermée puis le PC fait l'acquisition des tensions contenues dans les différents intégrateurs. Enfin grâce aux tables adéquates (propre à chaque type d'échantillon), le calcul des concentrations est réalisé puis affiché.

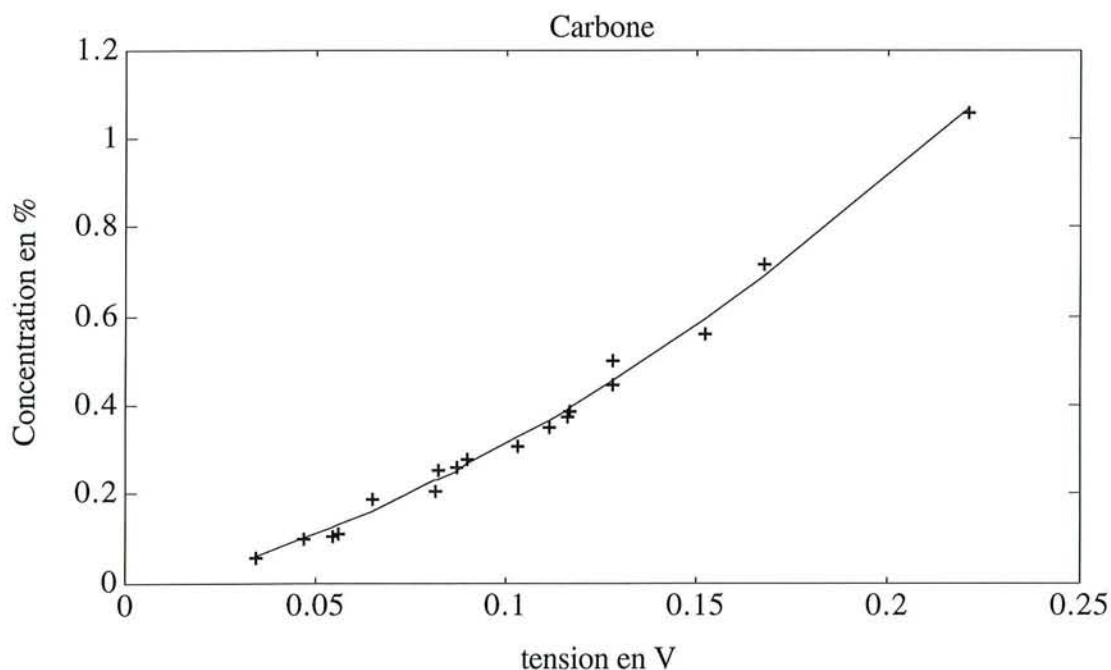
Les étapes d'une analyse sont les suivantes:

<b>Etape</b>	<b>Opération</b>
1	start à 1, permet la réalisation d'une mesure
2	ouverture de l'électrovanne d'argon
3	obturation de la fenêtre (fente primaire)
4	sélection du mode de l'unisource (A,B,C)
5	autorisation de l'intégration
6	début de l'étincelage (15 sec, haute tension)
7	arrêt de l'étincelage
8	ouverture de la fenêtre (fente primaire)
8	sélection de tous les intégrateurs
10	mise à la terre des intégrateurs
11	restauration du transfert normal
12	sélection du mode de l'unisource (A,B,C)
13	autorisation de l'intégration
14	début de l'étincelage (5 sec, basse tension)
15	arrêt de l'étincelage
17	fermeture de l'électrovanne d'argon
16	sélection des intégrateurs
18	obturation de la fenêtre
19	lecture des intégrateurs

**Tableau 4.1** *Séquence d'analyse*

A la fin de l'analyse, un programme sur le PC est chargé de convertir les tensions issues des intégrateurs en concentrations. Pour cela, on utilise des polynômes (approximant les courbes  $\% = f(V)$ ) obtenus par interpolation lors d'une phase d'étalonnage.

La phase d'étalonnage est réalisée une seule fois à partir d'un ensemble d'étalons (concentrations connues). L'analyse des étalons permet de disposer d'un ensemble de points (représentés par des "+" sur la figure II.4.1) de coordonnées (x, y) où x est la tension obtenue après analyse et y la concentration fournie par le fabricant. Le but de l'interpolation est de calculer l'équation de la courbe (en trait plein sur la figure II.4.1) qui approche le mieux tous ces points.



**Figure II.4.1** Courbe de conversion

Ceci revient, pour chaque composant chimique, à calculer les coefficients  $(a_n, a_{n-1}, \dots, a_0)$  d'un polynôme d'ordre n (en pratique un ordre 2 ou 3 suffit) vérifiant les expressions suivantes:



$$\begin{cases} a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_1 x_1 + a_0 = y_1 \\ a_n x_2^n + a_{n-1} x_2^{n-1} + \dots + a_1 x_2 + a_0 = y_2 \\ \dots \\ a_n x_m^n + a_{n-1} x_m^{n-1} + \dots + a_1 x_m + a_0 = y_m \end{cases} \quad [\text{IV.4.1}]$$

où chaque  $x_i$  est la tension mesurée sur le quantomètre pour l'étalon  $i$ , correspondant à une concentrations  $y_i$  d'un atome particulier. Pour avoir une solution acceptable à ce problème on doit avoir  $m > n$  (le nombre d'étalons,  $m/3$ , chaque étalon étant analysé trois fois, doit être au moins égal à l'ordre du polynôme recherché plus un).

Le système [IV.4.1] peut encore s'écrire sous la forme équivalente suivante:

$$X.a = y \quad [\text{IV.4.2}]$$

$$\text{où } X = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{pmatrix}, \quad a = \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-1} \\ a_n \end{pmatrix} \quad \text{et } y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_{n-1} \\ y_m \end{pmatrix}$$

Les étalons utilisés étant tous différents, la matrice  $X^T$  est une matrice de Vandermonde et  $X^T X$  est de rang plein. La meilleure approximation du vecteur  $a$  est obtenue en résolvant [IV.4.2] au sens des moindres carrés:  $a = (X^T X)^{-1} X^T y$ . Pour implémenter en langage C cette solution d'une manière numériquement robuste, nous avons choisis d'utiliser la factorisation UD. A partir de l'équation [IV.4.2] nous avons:

$X^T X.a = X^T y$  où  $(X^T X)$  est une matrice carrée, symétrique et de rang plein donc inversible. La factorisation UD de l'inverse de la matrice  $(X^T X)$  est:

$$(X^T X)^{-1} = U^T D U$$

où  $U^T$  est une matrice triangulaire supérieure ayant des 1 sur sa diagonale et  $D$  une matrice diagonale.

On obtient alors le résultat suivant:

$$[a = U^T.D.U.X^T.y.$$

La courbe de la figure II.4.1 est un cas réel obtenu par une interpolation d'ordre 2 à partir de 54 mesures (18 étalons). Les coefficients du polynôme obtenu sont les suivants:  $a_0 = -0.0285$ ,  $a_1 = 2.1385$  et  $a_2 = 12.7292$ . La fonction polynomiale ainsi calculée permet de convertir les tensions mesurées par l'appareil en concentration de carbone pour tout échantillon de ce groupe.

Le programme de calcul des concentrations dispose alors en mémoire d'une courbe (un polynôme) pour chaque élément (Carbone, Fer, Manganèse, ..) et pour chaque groupe de métaux analysés. Cependant, lors d'une analyse, la moindre défaillance (électrovanne d'argon non ouverte, obturateur bloqué,..) génère des résultats erronés. Aussi ces défauts doivent être détectés, localisés et étudiés en vue de prendre une décision, voire pour enrichir une base de connaissances.

5. Sûreté de fonctionnement

5.1. Modélisation

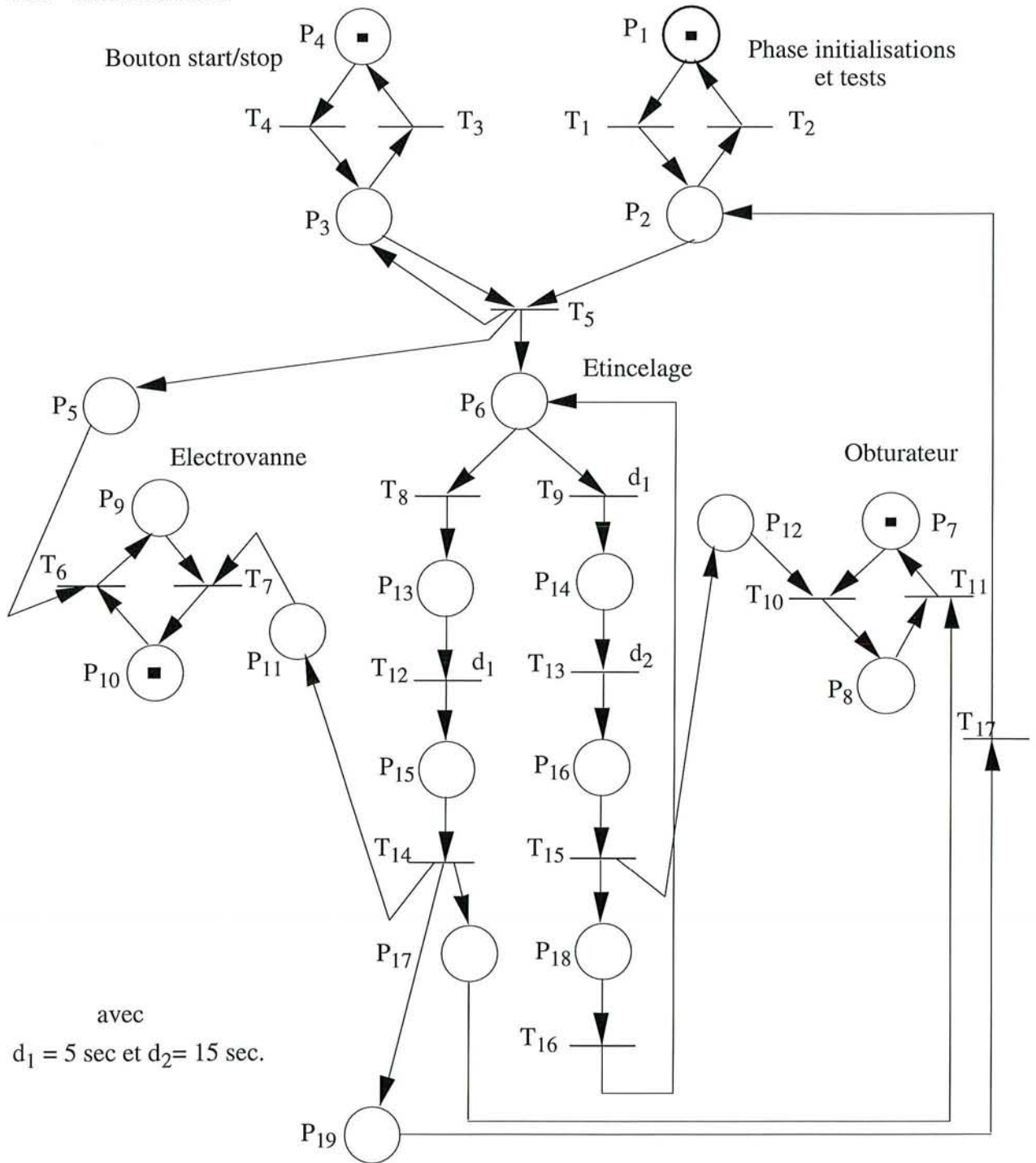


Figure II.5.1 Modèle RDP du quantomètre

Le modèle de la figure II.5.1 représente le fonctionnement de l'appareil de mesure avec principalement la séquence d'analyse présentée au tableau 4.1. Ce modèle reconstruit les différents états pris par les éléments vitaux de l'instrument (électrovanne, obturateur, ...). Les actions associées aux différentes places du réseaux sont les suivantes:

<b>Symbole</b>	<b>Signification</b>
P <sub>1</sub>	arrêt
P <sub>2</sub>	fin des tests
P <sub>3</sub>	bouton start/stop on
P <sub>4</sub>	bouton start/stop off
P <sub>5</sub>	validation de l'ouverture de l'électrovanne
P <sub>6</sub>	prêt à étinceler
P <sub>7</sub>	obturateur ouvert
P <sub>8</sub>	obturateur fermé
P <sub>9</sub>	électrovanne d'argon ouverte
P <sub>10</sub>	électrovanne d'argon fermée
P <sub>11</sub>	validation de la fermeture de l'électrovanne
P <sub>12</sub>	validation de la fermeture de l'obturateur
P <sub>13</sub>	étincelle basse tension
P <sub>14</sub>	étincelle haute tension
P <sub>15</sub>	arrêt étincelle B.T.
P <sub>16</sub>	arrêt étincelle H.T.
P <sub>17</sub>	validation de l'ouverture de l'obturateur
P <sub>18</sub>	RAZ des intégrateurs
P <sub>19</sub>	fin de l'analyse lecture de intégrateurs

**Tableau 5.1** Signification des places

Le passage d'un état à un autre a lieu sur occurrence d'un événement ou à la fin d'une temporisation, ceci se traduit sur un RdP par le franchissement d'une transition. A titre d'exemple, la fin des tests (transition T<sub>1</sub>) est indiqué par la présence d'un jeton dans la place P<sub>2</sub>. A cet instant deux transitions sont validées T<sub>2</sub> et T<sub>5</sub> mais une seule peut être franchie, il y a un conflit effectif. Si les tests sont bons, on choisit de franchir T<sub>5</sub> et donc d'autoriser les analyses sinon T<sub>2</sub> est franchie et la séquence de tests devra être réexécutée.

Cependant une séquence d'analyse ne peut démarrer que si le bouton marche/arrêt est dans la position "ON", c'est-à-dire si la place  $P_3$  contient une marque. Pour effectuer une analyse l'opérateur doit en faire la demande en sélectionnant le programme correspondant sur le PC puis en confirmant son choix par la mise en position "M" du bouton Marche/Arrêt.

## 5.2. Surveillance

Soit  $C$ , la matrice d'incidence du RdP modélisant le fonctionnement d'un quantomètre et dont le graphique est donné par la figure II.5.1:

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

En application de l'algorithme 1 du chapitre III, cette matrice est utilisée pour valider une séquence de franchissement [III.3.1]. A chaque étape, grâce à la relation fondamentale des RdP, le nouveau marquage est calculé.

Pour utiliser l'algorithme 2 du chapitre III, il est nécessaire de calculer les invariants. L'une des méthodes exposées lors du chapitre II permet d'obtenir les résultats qui suivent.

### T-semi-flots

Ils expriment la capacité du système à se réinitialiser, cette propriété est indispensable pour un appareil de ce type. En effet, après chaque analyse, le quantomètre doit pouvoir revenir à son état initial et donc être en mesure d'effectuer une autre analyse.

Soient  $S_i$  ( $i = 1, 2, 3$ ) les trois T-semi-flots obtenus:

$$\begin{aligned} S_1 &= [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\ S_2 &= [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\ S_3 &= [1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \end{aligned}$$

$S_1$  correspond à la séquence de franchissement " $T_1T_2$ ". Cette séquence équivaut au choix d'empêcher l'analyse suite à un défaut constaté sur l'un des éléments de l'appareil. A la mise sous tension, le programme de contrôle du quantomètre effectue toujours un test des composants sensibles (basses tensions, hautes tensions, température, vide,...). Le diagnostic de l'appareil est affiché et si l'une des valeurs est aberrante, l'opérateur est sollicité pour pallier à ce défaut. Le marquage initial pour un fonctionnement normal de l'appareil est:

$$M_0^T = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$S_2$ , bien qu'étant un résultat numériquement juste, ne correspond pas à une action logique car cela reviendrait à mettre le bouton M/A en position Marche puis aussitôt après en position Arrêt. Cependant, on a bien un retour à l'état initial.

Enfin  $S_3$  est une séquence complète d'analyse à l'issue de laquelle l'appareil revient à ses conditions initiales.

### P-semi-flots

Le système  $C^T \cdot x = 0$  a quatre solutions entières positives:

$$\begin{aligned} F_1 &= [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\ F_2 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\ F_3 &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \\ F_4 &= [1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1] \end{aligned}$$

Chaque P-semi-flot correspond à un cycle (Def. 3.2.1 Chap.III). Les vecteurs  $F_i$  ( $1 \leq i \leq 4$ ) permettent de déduire les relations d'invariance et les interprétations suivantes:

$F_1 \Rightarrow$  l'ensemble de places  $\{P_3, P_4\}$  donne les états du bouton M/A (Marche/Arrêt). Une marque en  $P_3$  (1,0) indique que le bouton est en position arrêt sinon une marque en  $P_4$  (0,1) indique que le bouton est en position marche, tout autre marquage est faux, d'où la relation:

$$M(P_3) + M(P_4) = M_0(P_3) + M_0(P_4) = 1$$

$F_2 \Rightarrow$  l'ensemble de places  $\{P_9, P_{10}\}$  donne les états de l'électrovanne (Ouverte/Fermée). Une marque en  $P_9$  (1,0) indique qu'elle est ouverte sinon une marque en  $P_{10}$  (0,1) indique qu'elle est fermée, tout autre marquage est faux, d'où la relation:

$$M(P_9) + M(P_{10}) = M_0(P_9) + M_0(P_{10}) = 1$$

$F_3 \Rightarrow$  l'ensemble de places  $\{P_7, P_8\}$  donne les états de l'obturateur (Ouvert/Fermé). Une marque en  $P_7$  (1,0) indique qu'il est fermé sinon une marque en  $P_8$  (0,1) indique qu'il est ouvert, tout autre marquage est faux, d'où la relation:

$$M(P_7) + M(P_8) = M_0(P_7) + M_0(P_8) = 1$$

$F_4 \Rightarrow$  l'ensemble de places  $\{P_1, P_2, P_6, P_{13}, P_{14}, P_{15}, P_{16}, P_{18}, P_{19}\}$  représente la phase d'étincelage. Cette phase est constituée de deux étapes. Pendant la première étape, l'étincelle dure 15 secondes et se fait sans éblouir les phototubes (obturateur fermé), alors que lors de la seconde étincelle (seulement 5 secondes) l'obturateur est ouvert, d'où la relation:

$$M(P_1) + M(P_2) + M(P_6) + M(P_{13}) + M(P_{14}) + M(P_{15}) + M(P_{16}) + M(P_{18}) + M(P_{19}) = 1.$$

Pendant la durée de ces deux étapes l'électrovanne d'argon est ouverte donc:

$$M(P_9) = 1. \quad \text{[IV.5.1]}$$

D'après le théorème 7.1 du chapitre II, toutes les places intervenant dans une expression d'invariance sont bornées. On a donc montré que:

$$M(P_i) \leq 1 \quad \text{pour } i \in \{1,2,3,4,6,7,8,9,10,13,14,15,16,17,18,19\},$$

tous les états décisifs sont bornés (les places  $P_5$ ,  $P_{11}$  et  $P_{12}$  sont des états intermédiaires, non bornés). En effet on n'a pas de circuit à ce niveau du RdP car on transmet un ordre à un actionneur (électrovanne, obturateur) mais ce processus n'est pas équipé de capteurs permettant de vérifier que l'ordre a bien été exécuté.

Le calcul des invariants permet donc de valider le modèle, de s'assurer qu'il répond correctement au cahier des charges et que les états accessibles par les éléments sensibles du quantomètre sont admissibles. D'autre part, à chaque itération de l'algorithme 2 du chapitre III, les quatre relations précédentes doivent rester vraies. Si l'une des relations n'est pas vérifiée à un instant  $t$  alors on est en présence d'un défaut, de plus la localisation est immédiate puisque les places mises en jeu font partie d'un cycle qui, lui-même, correspond à une fonction précise (obturateur, électrovanne,...).

L'intérêt d'un système de surveillance dans ce cas n'est pas de réaliser en temps réel une reconfiguration de la commande. En effet, comme le système passe par une suite d'états finis, les séquences de commande sont figées. Par contre le modèle RdP est très utile pour détecter les éventuels états incohérents par lesquels pourrait passer le système et diagnostiquer les causes de ces défaillances.

L'historique des défauts dont on dispose pour ce système est encore assez récent, néanmoins les points suivants doivent être sûrs car ils sont vitaux pour le système:

1- L'ouverture de l'argon doit avoir lieu cinq secondes avant le début de l'étincelage **sinon** l'attaque sera insuffisante d'où une mauvaise homogénéisation de la surface de l'échantillon. De plus, l'étincelle n'étant pas effectuée sous une atmosphère neutre, l'unisource risque d'être endommagée.

2- La fermeture de l'obturateur doit avoir lieu avant le début de la première étincelle **sinon** les phototubes seront éblouis.

3- Le déclenchement de l'étincelle doit se produire **sinon** il n'y aura pas d'attaque et donc pas de signal de sortie.



4- L'ouverture de l'obturateur doit avoir lieu avant le début de la deuxième étincelle **sinon** les phototubes ne recevront pas les rayons lumineux.

5- La fermeture de l'argon doit avoir lieu à la fin de la deuxième étincelle **sinon** il y aura un risque de pénurie d'argon pour les prochaines analyses.

Dans un système de ce type, la vérification des seules relations d'invariance ne permet pas de détecter les défauts précités. Les invariants permettent d'isoler un circuit et donc un composant mais ceci est insuffisant pour ce système car c'est la corrélation entre deux éléments qui est importante (argon-étincelle, obturateur-étincelle, ...). On a donc besoin de connaître, à un instant précis, la position de chaque élément. Cette information n'est pas fournie par les invariants. Mais on peut avantageusement compléter nos tests par des relations telles que la relation [IV.5.1] et vérifier que, pendant la phase d'étincelage, l'électrovanne d'argon est ouverte.

Le diagnostic par modèle de référence, tel qu'il a été présenté lors du chapitre III, répond parfaitement aux besoins de cette application car les défauts précédents (1 à 5) sont détectables et localisables, mais cela suppose un état complètement mesuré.

La détection des défauts est quasi immédiate, mais la rapidité de cette étape n'est pas primordiale. En effet, la seule décision à prendre en temps réel est l'arrêt d'urgence par exemple pour éviter un étincelage sans argon. La décision attendue est de valider ou d'invalider une analyse si le système est passé par un état aberrant non grave et dans ce cas de fournir une indication à l'opérateur sur l'élément incriminé.

## **Conclusion**

Les applications présentées dans ce chapitre continuent de faire l'objet de travaux de l'équipe d'automatique et de recherche appliquée de Longwy (E.A.R.A.L./C.R.A.N.). La première application est toujours en cours de développement d'autant plus que le logiciel de supervision n'est qu'un prototype en cours de développement chez "BULL". La deuxième application impliquant directement des industriels, le quantomètre remis en état a repris son service normal aux Aciéries et Fonderies du Maroc à Casablanca alors que nous continuons les développements théoriques. En exposant deux applications aussi différentes l'une de l'autre, nous avons voulu montrer le champ d'utilisation des RdP et leur adéquation à la modélisation et au diagnostic des processus industriels.

### **Bibliographie**

**[BOU93]** A. BOURJIJ, M. ZASADZINSKI, M. DAROUACH, G. KRZAKALA, M. MUSSET "On the use of hybrid Petri Nets for control process safety: application to a steam-boilers network simulator" IEEE SMC le Touquet, oct. 1993.

**[BOU94]** A. BOURJIJ, M. ZASADZINSKI, M. DAROUACH "Diagnostic par les réseaux de Petri hybrides: application à un réseau de générateurs de vapeur simulé" Symposium ADPM'94 / AFCET, Automation of Mixed Processes: Dynamical Hybrid Systems, Bruxelles, 23-24 nov. 1994

**[COR84]** R. CORI, C. MAFFEZONI "Practical-Optimal Control of Drum Boiler Power Plant" Automatica, vol.20, N° 2, pp 163-173, 1984.

**[JAC93]** L. JACOT, P. LADET "A modeling framework for hybrid systems" IEEE SMC le Touquet, oct. 1993.

**[ROH87]** M. W. ROHSENOW, J.P. HARTENETT, E. N. GANIC "Handbook of Heat Transfert Fundamentals" Mc Graw-Hill Book Company 1987.

## **CONCLUSION GENERALE**

L'objectif visé par ce travail est d'apporter notre contribution à un sujet d'actualité et d'une importance stratégique: "La sûreté de fonctionnement des processus industriels". Le sujet étant vaste, le choix de l'orientation a été délicat.

Les résultats exposés dans ce mémoire peuvent être décomposés en deux classes:

- méthode, par la proposition d'algorithmes de résolution de problèmes tels que le calcul des invariants ou la détection de défauts par les Réseaux de Petri,
- mise en œuvre, par l'application de ces méthodes sur des processus industriels.

Dans un premier chapitre, un rappel des Réseaux de Petri est proposé avec leur apport graphique, leur représentation matricielle, leurs propriétés ainsi que quelques extensions telles que les RdP hybrides. En conclusion du chapitre, deux exemples sont donnés pour montrer la facilité d'utilisation de cet outil mais aussi tout l'intérêt que présente une telle approche.

Lors du second chapitre, nous réalisons un tour d'horizon des méthodes de calcul des invariants et nous proposons une nouvelle approche pour la résolution de ce problème. La confrontation des performances des différents algorithmes montre l'intérêt de la méthode. Dans le cas de systèmes de grande dimension, une méthodologie basée sur la technologie de groupes est développée.

Dans le troisième chapitre, nous étudions l'emploi des RdP pour le diagnostic des processus. Récemment plusieurs équipes, notamment celles spécialisées en productique, utilisent des modèles de référence basés RdP pour la conduite des ateliers flexibles. Un algorithme de détection de défauts par modèle de référence est explicité. Bien que ce diagnostic soit global, son application peut s'avérer difficile pour certains systèmes:

- trop complexes,
- dont le système de commande est inaccessible
- ou encore dont l'état n'est pas complètement mesuré.

C'est pourquoi nous proposons une solution intermédiaire exploitant les relations de redondance données par les P-semi-flots. Cette technique est moins sûre que la première, mais elle est facile à mettre en œuvre et peut s'avérer assez efficace comme solution complémentaire.

Nous avons choisi d'appliquer nos développements théoriques sur des applications industrielles, qui font l'objet de contrats de recherche de l'E.A.R.A.L./C.R.A.N.. Cette orientation s'est avérée intéressante, bien que pas toujours très évidente.

Les algorithmes proposés ont été testés sur des exemples de la littérature ou encore sur des cas réels et des comparaisons ont été effectuées avec les algorithmes classiques pour montrer l'apport au niveau rapidité de calcul et simplicité de mise en œuvre. Cependant, d'autres approches de détection de défauts notamment par analyse d'un arbre de fautes pourraient être améliorées et implantées sur le réseau de générateurs de vapeur simulé. Ce dernier étant d'ailleurs un peu négligé, le système d'exploitation UNIX constituant un frein aux utilisateurs. Une solution serait de porter l'application sous MS/DOS sur un "PC" (un nœud du réseau Ethernet) ou plus simplement de réaliser une interface utilisateur sur "PC".

La deuxième application (le quantomètre) constituait une bonne base de tests, les impératifs de l'industrie font qu'elle n'est plus disponible aux chercheurs. Cependant, on dispose d'un modèle et de toutes les caractéristiques de l'appareil. Les simulations restent donc possibles.

Le problème posé par les méthodes de diagnostic basées RdP, mesure de tout l'état du système, ce qui nécessite une grande quantité de capteurs, permet de proposer une nouvelle voie de recherche, le développement d'observateurs pour les RdP.

Les RdP, bien qu'étant un outil puissant, présentent une déficience du fait de leur incapacité à représenter la dynamique de certains processus. Aussi, il serait intéressant d'étudier l'apport de l'algèbre Max+.

Enfin, l'orientation choisie, à savoir les RdP est un créneau récent au sein de l'équipe de Longwy, mais une brèche est ouverte et de jeunes chercheurs continueront certainement dans cette voie.

## **LISTE DES FIGURES**

**Chapitre I**

2.1	Quatre saisons	9
2.2	Assemblage	9
2.3	RdP généralisé	11
3.1	Matrice d'incidence	13
4.1	Quelques classes de RdP	18
5.1	RdP d'une double file FIFO	20
5.2	Graphe des marquages	20
5.3	Graphe de couverture	21
6.1	Fonctionnement d'un RdPT	23
6.2	Introduction aux RdPC	27
6.3	Trajectoire des marquages	27
6.4	Nœuds d'un RdPH	30
6.5	RdP Hybride	33
7.1	RdP d'un automatisme de fabrication	34
7.2	RdPT d'une phase de connexion-déconnexion	36

**Chapitre II**

6.1	Décompositions en blocs diagonaux	65
6.2	Isolement des blocs de couplage	67
6.3	Confrontation du nombre d'opérations	73

**Chapitre III**

2.1	Supervision	82
2.2	Système de surveillance	83
2.3	Structure d'un système de surveillance	84
3.1	Intérêt de la décomposition d'une séquence	91

## Chapitre IV

I.2.1	Une bouilloire	107
I.2.2	Le simulateur	111
I.3.1	RdP Hybride	113
I.3.2	Arrêt d'urgence	115
I.3.3	Modèle global	116
I.3.4	Partie discrète	117
II.2.1	Représentation schématique de l'excitation	121
II.3.1	Le spectromètre	123
II.4.1	Courbe de conversion	126
II.5.1	Modèle RdP du quantomètre	129



Nom : BOURJIJ

Prénom : Abdellatif

DOCTORAT de l'UNIVERSITE HENRI POINCARÉ, NANCY-I

en AUTOMATIQUE

VU, APPROUVÉ ET PERMIS D'IMPRIMER

Nancy, le 19 DEC. 1994 - u° 458

Le Président de l'Université



---

## "CONTRIBUTION À LA SURETE DE FONCTIONNEMENT DES PROCESSUS INDUSTRIELS PAR LES RESEAUX DE PETRI"

**Résumé:** Dans ce mémoire, nous nous sommes intéressés à la sûreté de fonctionnement des processus industriels. L'approche préconisée consiste à modéliser le système par les Réseaux de Petri. L'algèbre linéaire, notamment le calcul des invariants, est utilisée pour vérifier qu'un modèle a les propriétés requises (vivant, borné ..) et répond au cahier des charges. Aussi, nous proposons un algorithme efficient de calcul des invariants, particulièrement adapté aux systèmes de grande dimension. Puis deux méthodes de diagnostic basées, pour la première sur un modèle de référence (RdP), pour la seconde sur l'utilisation des invariants, sont présentées. Enfin, ces développements théoriques sont appliqués à deux processus industriels: un réseau de générateurs de vapeur simulé et un quantomètre à émission optique.

**Mots clefs:** Réseaux de Petri, diagnostic, P-semi-flots, T-semi-flots, sûreté de fonctionnement.

---

## "CONTRIBUTION TO THE INDUSTRIAL PROCESSES SAFETY WORKING BY THE USE OF PETRI NETS"

**Abstract:** The aim of this work is the safety working of industrial processes. The proposed approach consists in modelling the process through Petri Nets. The model validation, i. e. verifying if the model has the required specifications, is achieved by using linear algebra tools, especially by invariants computation. In fact, we suggest an efficient algorithm, which is particularly adapted to large scale systems, for the invariants computations. Then two diagnosis methods are presented: the first one is based on reference model and the second on the use of invariants. Finally, these theoretical results are applied to industrial processes: a steam boilers net simulator and a mass spectrometer.

**Keywords:** Petri Nets, diagnostic, S-invariants, T-invariants, safety working.